On an Intermediate Task for Classifying URL Citations on Scholarly Papers

Kazuhiro Wada[1], **Masaya Tsunokake**[2], **Shigeki Matsubara**[1,3]
[1]Graduate School of Nagoya University, Nagoya, Japan
[2]Research and Development Group, Hitachi, Ltd., Kokubunji, Tokyo, Japan
[3] Information Technology Center, Nagoya University, Nagoya, Japan
wada.kazuhiro.s8@s.mail.nagoya-u.ac.jp
masaya.tsunokake@gmail.com
matubara@nagoya-u.jp

Abstract

Citations using URL (URL citations) that appear in scholarly papers can be used as an information source for the research resource search engines. In particular, the information about the types of cited resources and reasons for their citation is crucial to describe the resources and their relations in the search services. To obtain this information, previous studies proposed some methods for classifying URL citations. However, their methods trained the model using a simple fine-tuning strategy and exhibited insufficient performance. We propose a classification method using a novel intermediate task. Our method trains the model on our intermediate task of identifying whether sample pairs belong to the same class before being fine-tuned on the target task. In the experiment, our method outperformed previous methods using the simple fine-tuning strategy with higher macro F-scores for different model sizes and architectures. Our analysis results indicate that the model learns the class boundaries of the target task by training our intermediate task. Our intermediate task also demonstrated higher performance and computational efficiency than an alternative intermediate task using triplet loss. Finally, we applied our method to other text classification tasks and confirmed the effectiveness when a simple fine-tuning strategy does not stably work.

Keywords: Digital Libraries, Scholarly Document Processing, Text Categorization, Citation Analysis, Intermediate Task Training

1. Introduction

In academic activities, preparing research resources (e.g., data and programs) to be utilized is essential. In scholarly papers, these resources are often cited by URLs (Zhao et al., 2018; Park and Park, 2019), which are called **URL citations** (Tsunokake and Matsubara, 2022). Unlike conventional citations for papers, URL citations refer to various resources and appear in various locations, such as references, footnotes, and body texts. If research resource services utilize URL citations, researchers could search and discover the research resources more easily. Furthermore, researchers could discover related studies that cite the same resource. To develop such services, information about the URL citations is required. In particular, the information about the types of cited resources and reasons for their citation is vital to describe the resources and their relations in the search services.

To extract information on the URL citations, previous studies defined classification schemes and proposed the methods that categorize the URL citations (Zhao et al., 2019; Tsunokake and Matsubara, 2022). The scheme defined by Tsunokake and Matsubara (2022) has 3 labels (*role*, *type*, and *function*). Figure 1 shows an example of each label, where *role* and *type* represent the types of URL citations and *function* represents the purpose of the citation, respectively. Their methods classify the URL citations by fine-tuned BERT (Zhao et al., 2019; Tsunokake and Matsubara, 2022).

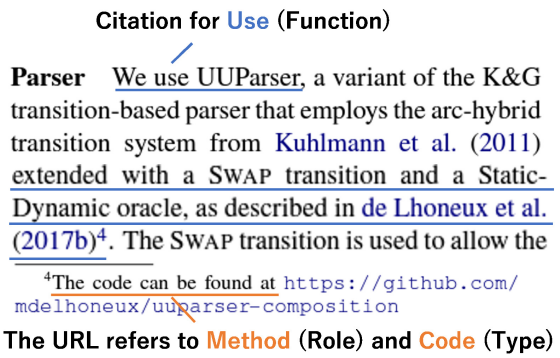

Figure 1: Example of the labels of URL citations (cited from de Lhoneux et al. (2019))

Previous studies employ a simple fine-tuning strategy. However, their method exhibited insufficient performance. One promising approach for improving performance when simple fine-tuning does not work is intermediate task training, which trains on the supplemental task (**intermediate task**) before fine-tuning the target task (Phang et al., 2018). This approach may boost the performance in URL citation classification. When using this approach, it is necessary to find an appropriate intermediate

LREC-COLING 2024, pages 12359–12369

task (Chang and Lu, 2021; Poth et al., 2021). Previous studies have found the appropriate intermediate task by selecting tasks based on their relatedness to the target task (Poth et al., 2021). Unlike major NLP tasks, it is difficult to find classification tasks related to URL citation classification, which is minor and complex. One solution for overcoming this challenge is to train another related task on the same dataset used for the target task.

In this paper, we propose a URL citation classification method using a novel intermediate task of identifying whether sample pairs belong to the same class. We assume that the model can learn the class boundaries of the target task through training on our intermediate task. This allows the model to distinguish classes more easily in the fine-tuning stage.

We evaluated our method on URL citations obtained from international conference papers.[1] In the experiment, our method achieved higher macro and weighted average F-scores on the $\text{BERT}_{\text{base}}$ and $\text{RoBERTa}_{\text{base}}$ by 0.014 and 0.013 for *role*, 0.038 and 0.026 for *type*, and 0.093 and 0.028 for *function* compared to that of previous methods using simple fine-tuning strategy. We also verified that our method performed effectively in minority classes.

Furthermore, we visualized the feature space using the t-SNE (van der Maaten and Hinton, 2008) and computed silhouette coefficients to validate our hypothesis. The results showed that the model learned the class boundaries of the target task by training on our intermediate task. In further experiments, our intermediate task demonstrated higher performance and computational effectiveness than metric learning using Triplet loss, an alternative intermediate task. We also applied our method to other text classification tasks such as AG-news (Zhang et al., 2015) and CoLA (Warstadt et al., 2018), and confirmed its effectiveness in the tasks in which a simple fine-tuning strategy does not stably work.

Our contributions are as follows:

- We proposed a method using a novel intermediate task, identifying whether sample pairs belong to the same class for classifying the URL citations.
- Experimental results show the effectiveness of our method with different model sizes and architectures. In particular, our method achieved higher macro and weighted average F-scores on the $\text{BERT}_{\text{base}}$ compared to that of previous methods that only fine-tuned the model to the target task.
- We visualized the feature space of $\text{BERT}_{\text{base}}$ and showed that the model learns the class boundaries of the target task by training on our intermediate task.
- We confirmed that our method was effective in the text classification tasks in which a simple fine-tuning strategy does not stably work.

[1] Our code is available at https://github.com/matsubara-labo/URL_Citation_Classification_Intermediate

2. Related Work

2.1. Classification of URL Citations

Previous studies have attempted to categorize URL citations (Zhao et al., 2019; Tsunokake and Matsubara, 2022). Zhao et al. (2019) first proposed the clarification schema for URL citations. This schema has 3 labels (*resource role*, *type*, and *function*). The *resource role* and *type* represent the kinds of resources, and the *resource function* denotes the purpose of citations. Based on Zhao et al. (2019)'s schema, Tsunokake and Matsubara (2022) proposed a new categorization. They split *Data* label of *type* into more fine-grained labels (*Dataset*, *DataSource*, and *Knowledge*). They added *Mixed* as a new label into *role* and *type* because multiple resources can be cited with a single URL. Tsunokake and Matsubara (2022) proposed the method to classify URL citations based on sentences surrounding the citations (**citation context**), section titles, and footnote texts used in the citation. Their method fine-tunes BERT on the target task in a multi-task learning manner.

2.2. Intermediate Task Training

The widespread supervised learning approach involves fine-tuning the pre-trained large language models for target tasks. Phang et al. (2018); Chang and Lu (2021) proposed intermediate task training to bridge the gap between the pre-trained and desired features for the target task.

However, Poth et al. (2021); Pruksachatkun et al. (2020) pointed out that using the intermediate task is not always effective. To address this issue, Poth et al. (2021) presented a method for selecting appropriate intermediate tasks based on the similarity of datasets and pre-trained models. Additionally, Pruksachatkun et al. (2020) analyzed the characteristics of appropriate intermediate tasks by comparing the performance of probing tasks, which measures specific linguistic abilities.

From these studies, we must collect the dataset and select an appropriate task to take advantage of the conventional intermediate task training. To overcome these limitations, we propose a novel intermediate task, which reuses the dataset for the target task and is closely related to the target task.

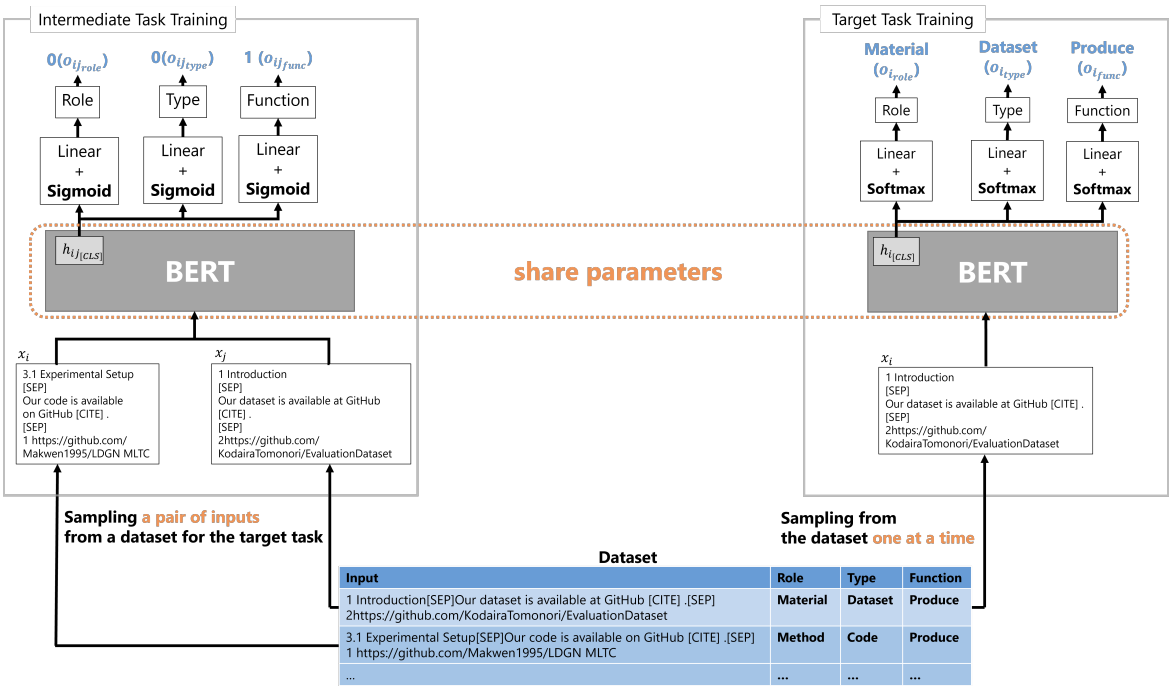

Figure 2: Overview of our method

3. Method

3.1. Problem Settings

We follow the schema of Tsunokake and Matsubara (2022). In this schema, one URL citation contains 3 labels (*role*, *type*, and *function*). Figure 1 shows the concepts of each label. While both *role* and *type* represent the type of resources referenced by the URL, *role* and *type* specify major and minor categories, respectively. *Function* represents the purpose of URL citation.

3.2. Overview of Our Method

We propose a method to classify URL citations via novel intermediate task training. Figure 2 shows the overview of our method. In our method, the model is trained on our intermediate task before training the target task—the classification of URL citations. The model learns the class boundaries of the target task by training on our intermediate task (See Section 3.3). The model consists of a BERT-like model as an encoder and a linear layer as an output layer. During training on both the intermediate and target tasks, the model shares the parameters of an encoder.

3.3. Intermediate Task

We expect that the model learns the class boundaries of each class via training on our intermediate task. Our intermediate task takes a pair of inputs $(\boldsymbol{x_i}, \boldsymbol{x_j})$ used in a target task and determines whether the pair belongs to the same class. The output from the encoder corresponding to the [CLS] token is fed to the linear layer of each label. The model uses the sigmoid as an activation function. This can be denoted as:

$$
\begin{aligned}
\boldsymbol{h_{ij}} &= E([\boldsymbol{x_i};\boldsymbol{x_j}]) \\
o_{ij_{role}} &= \sigma(\boldsymbol{W}^{inter}_{role}\boldsymbol{h_{ij}}_{[CLS]} + \boldsymbol{b}^{inter}_{role}) \\
o_{ij_{type}} &= \sigma(\boldsymbol{W}^{inter}_{type}\boldsymbol{h_{ij}}_{[CLS]} + \boldsymbol{b}^{inter}_{type}) \\
o_{ij_{func}} &= \sigma(\boldsymbol{W}^{inter}_{func}\boldsymbol{h_{ij}}_{[CLS]} + \boldsymbol{b}^{inter}_{func})
\end{aligned}
$$

where E denotes the BERT-like model, σ denotes a sigmoid function, $\boldsymbol{W}^{inter}_{role}, \boldsymbol{W}^{inter}_{type}, \boldsymbol{W}^{inter}_{func} \in \mathbb{R}^{d_h}$, which d_h is a dimension of a hidden layer of BERT-like model, and $\boldsymbol{b}^{inter}_{role}, \boldsymbol{b}^{inter}_{type}, \boldsymbol{b}^{inter}_{func} \in \mathbb{R}$. We assumed that training on this intermediate task enables a BERT-like model to learn the features that distinguish the classes in the target task. Following previous studies, we employ multi-task learning to leverage label relevance (Tsunokake and Matsubara, 2022; Zhao et al., 2019; Sener and Koltun, 2018). Therefore, the loss is calculated by the equation below:

$$
\begin{aligned}
L^{inter} &= L(o_{ij_{role}}, \delta_{y_{i_{role}} y_{j_{role}}}) \\
&+ L(o_{ij_{type}}, \delta_{y_{i_{type}} y_{j_{type}}}) \\
&+ L(o_{ij_{func}}, \delta_{y_{i_{func}} y_{j_{func}}})
\end{aligned}
$$

where L denotes a binary cross entropy, y_k denotes a true class of each label corresponding to x_k, and δ is a Kronecker's delta.

3.4. Target Task

The objective of the target task is to categorize URL citations with their type and purpose. This task has 3 labels (*role*, *type*, *function*), and the model classifies these labels simultaneously using multitask learning. The model uses citation contexts (c_i), section titles (t_i), and footnote or reference texts (f_i) as inputs, following Tsunokake and Matsubara (2022). Citation contexts contain a target sentence (s_i) to be classified and adjacent sentences (s_{i-1}, s_{i+1}). The output vector from the encoder corresponding to the [CLS] token is fed to the linear layer of each label. The model adopts the softmax as an activation function. The BERT-like models are initialized with the parameters of the encoder after training on our intermediate task. The model can be formulated as:

$$
\begin{aligned}
\boldsymbol{h_i} &= E(\boldsymbol{x_i}) \\
&\quad (c_i = [s_{i-1}; s_i; s_{i+1}], \boldsymbol{x_i} = [c_i; t_i; f_i]) \\
\boldsymbol{o}_{i_{role}} &= softmax(\boldsymbol{W}^{target}_{role}\boldsymbol{h}_{\boldsymbol{i}_{[CLS]}} + \boldsymbol{b}^{target}_{role}) \\
\boldsymbol{o}_{i_{type}} &= softmax(\boldsymbol{W}^{target}_{type}\boldsymbol{h}_{\boldsymbol{i}_{[CLS]}} + \boldsymbol{b}^{target}_{type}) \\
\boldsymbol{o}_{i_{func}} &= softmax(\boldsymbol{W}^{target}_{func}\boldsymbol{h}_{\boldsymbol{i}_{[CLS]}} + \boldsymbol{b}^{target}_{func})
\end{aligned}
$$

where E is the same as that of the intermediate task, $\boldsymbol{W}^{target}_{role}, \boldsymbol{W}^{target}_{type}, \boldsymbol{W}^{target}_{func} \in \mathbb{R}^{d_{class} \times d_h}$, and $\boldsymbol{b}^{target}_{role}, \boldsymbol{b}^{target}_{type}, \boldsymbol{b}^{target}_{func} \in \mathbb{R}^{d_{class}}$, which d_{class} is the number of class of each label. The loss is denoted as:

$$
\begin{aligned}
L^{target} &= L(\boldsymbol{o}_{i_{role}}, y_{i_{role}}) + L(\boldsymbol{o}_{i_{type}}, y_{i_{type}}) \\
&+ L(\boldsymbol{o}_{i_{func}}, y_{i_{func}})
\end{aligned}
$$

where L denotes a cross-entropy, and y_k denotes a true class of each label corresponding to x_k.

4. Experiment

To validate the efficacy of our method, we conducted experiments with different model sizes and architectures.

4.1. Experimental Settings

Dataset We use a dataset containing URL citations annotated with *role*, *type*, and *function*, following Tsunokake and Matsubara (2022) (See section 3.1). This data was created from papers in ACL anthology[2]. Table 1 shows the class distribution of each label. Each class of *role* is evenly distributed. However, *type* and *function* have a skewed distribution, which is expected to make the training challenging. We split the dataset into train (2,391 samples), validation (299 samples), and test (299 samples) sets.

[2]The papers are distributed under the Creative Commons 3.0 BY-NC-SA or 4.0 BY

Model We implemented our method with PyTorch[3], HuggingFace[4], and NLTK[5] libraries and used a pre-trained $\text{BERT}_{\text{base}}$ (Devlin et al., 2019), $\text{BERT}_{\text{large}}$ (Devlin et al., 2019) and $\text{RoBERTa}_{\text{base}}$ (Liu et al., 2019) as an encoder. A citation context, section title, and footnote text are concatenated with [SEP] and are fed to the model.

Parameters The model was trained using early stopping with patience at 2 in the intermediate task and 5 in the target task. We used Adam (Kingma and Ba, 2014) optimizer, and its leaning rate is 1e–5 (for a base model) and 4e–6 (for a large model). We determined these parameters based on the best macro F-score of the validation dataset using a random search. The batch size[6] for the intermediate task is 4 (for a base model) and 2 (for a large model), and that of the target task is 8 (for a base model) and 2 (for a large model). The pairs for the intermediate task were randomly generated from the dataset for the target task, and the number of them was 300,000. An RTX3080 GPU and i7-11700K (3.6GHz) CPU were used for the training for about 3 days.

Baseline We compared our method with Zhao et al. (2019)'s method, which utilizes the same model but only takes citation context as input, and Tsunokake and Matsubara (2022)'s method (denoted as "Tsunokake (2022)"), which employs the same model and inputs of our method. However, these methods do not use the intermediate task.

Evaluation We used the macro averaged and weighted averaged F-score as a metric.

4.2. Result

We trained each model with 3 seeds. Table 2 shows the average score of each method. Except for *role* with $\text{BERT}_{\text{large}}$, our method outperforms other methods in macro averaged and weighted averaged F-scores regardless of different model sizes and architectures. This result demonstrates the effectiveness of our method.

Table 3 shows the macro averaged F-score of each class. Compared to Tsunokake and Matsubara (2022), the F-scores of our method were improved by 0.444 in *Compare* and 0.200 in *Extend*. These improvements[7] are larger than that of other labels. Hence, these results indicate our

[3]PyTorch.org https://pytorch.org/

[4]Hugging Face https://huggingface.co/

[5]nltk.org https://www.nltk.org/

[6]We selected maximum sizes that can be loaded on our GPU as the batch sizes.

[7]The improvement in Compare is 3.7% and that in Extend is 3.3%.

Role		Type				Function	
label	size	label	size	label	size	label	size
Method	1,102	Tool	629	Paper	279	Use	1,231
Material	870	Code	473	DataSource	235	Introduce	886
Supplement	835	Dataset	353	Document	217	Produce	653
Mixed	182	Website	305	Mixed	182	Compare	111
		Knowledge	282	Media	34	Extend	100
						Other	8

Table 1: Class distribution of each label

	Role		Type		Function	
	macro	weighted	macro	weighted	macro	weighted
Zhao et al. (2019)+BERT$_{base}$	0.691	0.696	0.459	0.525	0.387	0.705
Tsunokake (2022)+BERT$_{base}$	0.753	0.743	0.518	0.596	0.386	0.735
Ours+BERT$_{base}$	**0.767**	**0.756**	**†0.556**	**†0.622**	**†0.479**	**†0.763**
Zhao et al. (2019)+BERT$_{large}$	0.663	0.669	0.447	0.503	0.376	0.700
Tsunokake (2022)+BERT$_{large}$	**0.782**	**0.765**	0.522	0.599	0.388	0.716
Ours+BERT$_{large}$	0.764	0.756	**†0.606**	**†0.652**	**†0.532**	**†0.771**
Zhao et al. (2019)+RoBERTa$_{base}$	0.719	0.725	0.488	0.552	0.384	0.716
Tsunokake (2022)+RoBERTa$_{base}$	0.752	0.746	0.530	0.606	0.397	0.734
Ours+RoBERTa$_{base}$	**0.776**	**0.763**	**†0.576**	**†0.650**	**0.455**	**0.742**
Ours+BERT$_{base}$	**0.767**	**0.756**	**0.556**	**0.622**	0.479	**0.763**
Triplet+BERT$_{base}$	0.765	0.750	0.497	0.564	**0.527**	**0.763**

Table 2: Averaged scores (macro and weighted F-score) of each method. The bolded text indicates the highest score for each encoder model. The items marked with a dagger (†) indicate that a significant difference was observed against Tsunokake and Matsubara (2022) with a significance level of 0.05.

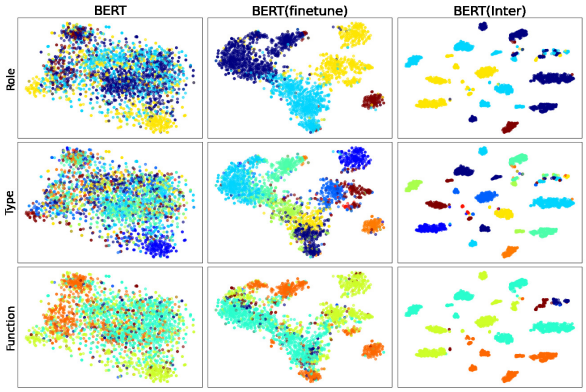

Figure 3: The compressed features by t-SNE of BERT$_{base}$ (Train). The individual rows are features of the pre-trained BERT$_{base}$, the fine-tuned BERT$_{base}$ (Tsunokake and Matsubara, 2022), and our method from the left.

method tends to work especially well in minority classes. However, the model still exhibited a low performance towards labels like *Other*, where the number of samples is extremely small.

In addition, the model demonstrated poor performance on the *Media*. We consider the primary reason for this to be the limited number of samples available. On the other hand, we found that there is a high similarity of proper nouns between *Media* and *Dataset* and think that it also resulted in poor performance. Proper nouns serve as one of the crucial clues for this task, and the overlap in proper nouns between the two classes may disturb the appropriate training and prediction by the model. We think that this high similarity between *Media* and *Dataset* adversely affected the model's ability to effectively distinguish between *Media* and other categories.

4.3. Discussion

4.3.1. Effectiveness of Intermediate Task

To investigate whether the features obtained by training on our intermediate task are effective on the target task, we also assessed the performance of the target task in freezing BERT$_{base}$ parameters after training our intermediate task. Table 4 shows the results of this experiment. While the performance on *role* decreased by 0.024, the performances on *type* and *function* were equal to or better than that without freezing. These competitive results indicate that the output vectors of BERT$_{base}$ corresponding to the [CLS] token, trained only on the intermediate task, can represent valuable features for the target task as the learnable parameter is only a single linear layer of each label.

4.3.2. Class Boundaries Learned by Our Intermediate Task

While creating our intermediate task, we assumed that the model could learn the class boundaries of

Role	F-score		Type	F-score		Function	F-score	
	Tsunokake (2022)	Ours		Tsunokake (2022)	Ours		Tsunokake (2022)	Ours
Method	0.795	0.829	Tool	0.729	0.691	Use	0.773	0.768
Material	0.624	0.692	Code	0.621	0.660	Introduce	0.690	0.756
Supplement	0.712	0.768	Dataset	0.595	0.597	Produce	0.797	0.810
Mixed	0.837	0.864	Website	0.523	0.686	Compare	0.000	0.444
			Knowledge	0.203	0.348	Extend	0.000	0.200
			Paper	0.873	0.842	Other	0.000	0.000
			DataSource	0.439	0.457			
			Document	0.400	0.486			
			Mixed	0.844	0.864			
			Media	0.000	0.000			

Table 3: Macro F-score of each class ($BERT_{base}$). The underlined items indicate the method with the higher score.

	Role	Type	Function
Tsunokake (2022)	0.753	0.518	0.386
Ours(Unfreeze)	**0.767**	0.556	0.479
Ours(Freeze)	0.743	**0.557**	**0.487**

Table 4: Macro F-scores when freezing $BERT_{base}$

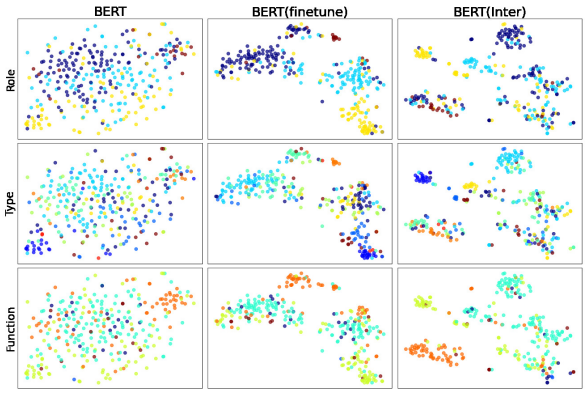

Figure 4: The compressed output vectors by t-SNE of $BERT_{base}$ (Validation). The individual rows are output vectors of the pre-trained $BERT_{base}$, the fine-tuned $BERT_{base}$ (Tsunokake and Matsubara, 2022), and our method from the left.

the target task (Section 3.3). To verify this assumption, we visualized the output vectors of $BERT_{base}$ corresponding to the [CLS] token with t-SNE[8]. Figures 3 and 4 show the outcomes of the training and validation sets, respectively. Columns in the figure correspond to the output vectors of pre-trained $BERT_{base}$ without training the intermediate and target tasks, that of $BERT_{base}$ after fine-tuning only on the target task, and that of $BERT_{base}$ after training on the intermediate task from the left side. Each dot represents a compressed output vector and is colored according to the corresponding class.

The class boundary of the output vectors becomes obvious in the training set by the training model on our intermediate task. This result demonstrates that our intermediate task contributes to learning the class boundaries required for the target task.

However, these characteristics are not clear in the validation set. Therefore, we calculated the average silhouette coefficient by each class to measure cohesion. Table 5 shows the result. The higher coefficient means that the cluster of a corresponding class is in one place. The silhouette coefficient of our method is higher than that of Tsunokake and Matsubara (2022) in almost all classes. From this result, we confirmed that the output vectors of the validation set have a similar trend to those of the training set.

4.3.3. Case Study

We investigated the drawbacks of our method. Table 6 shows examples where our methods have failed to predict correctly.

In the first example, the model predicted the *type* label as *Code*, but the actual label is *Mixed*. This resulted from a misunderstanding regarding the number of referenced resources. The underlined text in the first row refers to multiple resources such as code and Reddit discussion identifier by the URL. The model only predicted *Code* and missed other resources. The model may have focused on the "code" in the input text and missed the words indicating other resources.

In the second example, the model predicted the *type* and the *function* labels as *Dataset* and *Produce*, but the actual labels are *Tool* and *Use*. This resulted from a misinterpretation of what the URL refers to. The whole input text is about the dataset the authors produced, but the URL refers to the tagger used to develop a dataset. Therefore, the

[8] https://scikit-learn.org/stable/modules/generated/sklearn.manifold.TSNE.html

Role	Silhouette cor		Type	Silhouette cor		Function	Silhouette cor	
	Tsunokake (2022)	Ours		Tsunokake (2022)	Ours		Tsunokake (2022)	Ours
Method	0.275	0.828	Tool	0.317	0.825	Use	0.547	0.838
Material	0.232	0.705	Code	0.337	0.820	Introduce	0.523	0.832
Supplement	0.278	0.799	Dataset	0.502	0.776	Produce	0.584	0.553
Mixed	-0.048	0.725	Website	0.387	0.825	Compare	0.620	0.828
			Knowledge	0.492	0.801	Extend	0.691	0.769
			Paper	-0.169	0.480	Other	0.444	0.429
			DataSource	0.335	0.319			
			Document	0.443	0.785			
			Mixed	-0.048	0.725			
			Media	0.461	0.678			

Table 5: Silhouette coefficient by each class. The higher the value, the more condensed the cluster of the corresponding class. The underlined items indicate the method with the higher silhouette coefficient.

Input text	Role		Type		Function	
	Gold	Predicted	Gold	Predicted	Gold	Predicted
1 Introduction [SEP] First, we propose a novel reinforcement learning task with both states and combinatorial actions defined by natural language, [CITE] which is introduced in section 2. [SEP] 1 Simulator code and Reddit discussion identifiers are released at https://github.com/jvking/reddit-RL-simulator	**Mixed**	**Method**	**Mixed**	**Code**	Produce	Produce
3.1 Corpus [SEP] Finally, we automatically annotated all utterances with part-of-speech tags using TreeTagger (Schmid, 1994), which we've trained on the switchboard corpus of spoken language (Godfrey et al., 1992), because it contains, just like our corpus, speech disfluencies. [CITE] [SEP] 6 The tagger is available free for academic research from http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/DecisionTreeTagger.html.	**Method**	**Material**	**Tool**	**Dataset**	**Use**	**Produce**

Table 6: Examples where our methods have failed to predict correctly. The input text contains a section title, citation sentences, and a bibliography/footnote text. Citation sentences are truncated due to a page limit. The bold text is a label the model could not predict correctly.

model was influenced by the meaning of the whole input text rather than one sentence that explains a resource cited by the URL.

4.3.4. Comparison with Alternative Intermediate Task

According to Figure 3, URL citations in the same classes are near, and those in the different classes are far in the [CLS] embedding space. It may have allowed the model to capture the class boundary. Therefore, metric learning methods (Zhang et al., 2022; Gunel et al., 2021; Gajjar et al., 2022) can be an alternative intermediate task to acquire the [CLS] embeddings that differentiate the classes in the target task. To compare our approach and metric learning approach, we further assessed the alternative intermediate task with triplet loss (Vasileios Balntas and Mikolajczyk, 2016).

Table 2 shows the result. For reasons of computational cost, we only report the results for the $\text{BERT}_{\text{base}}$. Alternative intermediate task (Triplet+$\text{BERT}_{\text{base}}$) exhibited lower performances for *role* and *type* than our method (Ours+$\text{BERT}_{\text{base}}$). In the function classification, Triplet+$\text{BERT}_{\text{base}}$ outperformed our method on the macro–averaged F-score. This is because the class distribution of *function* is highly skewed. Our intermediate task sometimes does not contain any samples in the minority class because the pairs used in our intermediate task are created randomly. However, Triplet contains all classes as all pairs are generated from a batch. To address this issue, we can consider better sampling strategies based on a class distribution or the performance of the simple fine-tuning strategy. We leave this for future work.

Task group	Task name	#Class
News Categorization	r8 (Cardoso-Cachopo, 2007)	8
	r52 (Cardoso-Cachopo, 2007)	52
	ag-news (Zhang et al., 2015)	4
Question Categorization	trec-coarse (Hovy et al., 2001; Li and Roth, 2002)	6
	trec-fine (Hovy et al., 2001; Li and Roth, 2002)	50
	banking77 (Casanueva et al., 2020)	77
Sentiment	sst2 (Socher et al., 2013)	2
	sst5 (Socher et al., 2013)	5
	emotion (Saravia et al., 2018)	6
EDOS (Kirk et al., 2023)	Binary Sexism Detection	2
	Category of Sexism	4
	Fine-grained Vector of Sexism	11
Others	Clickbait Spoiling (Fröbe et al., 2023)	3
	CoLA (Warstadt et al., 2018)	2

Table 7: The 14 text classification tasks for an additional experiment. #Class denotes the number of different categories that the data can be classified into.

Our method also has computational efficiency. Triplet executes a forward calculation 3 times to obtain vectors of anchor, positive, and negative. Thus, the total number of forward calculations is 9 because the target task has 3 labels. However, our intermediate task needs only one forward calculation. Our method performed better than the alternative methods on *role* and *type* with fewer forward calculations.

4.3.5. Effectiveness of Our Method for Other Text Classification Tasks

Our method can also be applied to other text classification tasks. We performed additional experiments with other text classification tasks to investigate the characteristics of the tasks on which our method effectively works. The table 7 shows a list of 14 tasks for this experiment. We selected general tasks from previous studies and the SemEval 2023 tasks[9]. We used the $\text{BERT}_{\text{base}}$ as an encoder. The number of samples for the intermediate task training is 100,000 due to computational cost. We take an average of the evaluation results with 3 seeds for each task.

Figure 5 and 6 show the difference in macro and weighted F-scores between our method and simple fine-tuning strategy. Our method obtained higher performances in 10 out of 14 tasks for the macro averaged F-score and 11 out of 14 tasks for the weighted F-score. These findings demonstrate that our method tends to be efficient for other NLP tasks.

To analyze when our method is particularly effective, we conducted a correlation study. We calculated the Pearson correlation coefficients from some numerical characteristics and differences in scores between our method and the simple fine-tuning strategy. Table 8 shows the result. The standard deviation of scores on the target task with a simple fine-tuning strategy revealed a positive correlation for the macro and weighted F-scores. The high standard deviation means that the model's performance is highly dependent on random seeds. Thus, this positive correlation indicates that our method is effective when the performance of the simple fine-tuning strategy is unstable.

On the other hand, the average length of the input texts of the target task exhibited a weak negative correlation. The BERT-like model used in this study has a token length limitation. Therefore, if an input text is too long, the text will be truncated. This truncation occurs more often because our method concatenates a pair of input texts in the intermediate training step. A negative correlation with the average length of the input texts indicates that vital information may be lost due to truncation.

5. Conclusion and Future Work

This paper proposed a novel classification method for the *role*, *type*, and *function* of URL citations in scholarly papers utilizing intermediate task training. We set the novel intermediate task of identifying whether sample pairs belong to the same class, which reuses a dataset for the target task. The experimental results demonstrated the efficacy of our method with different model sizes and architectures. Furthermore, we discussed the characteristics of our intermediate task. Visualization results and silhouette coefficients based on the output vectors from $\text{BERT}_{\text{base}}$ indicated that the model could learn the class boundaries through training on our intermediate task. We also compared our intermediate task with an alternative task that uses triplet

[9] https://semeval.github.io/SemEval2023/tasks

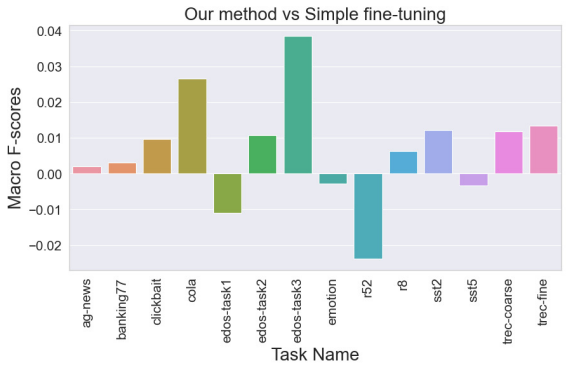

Figure 5: Difference in macro F-scores between our method and a simple fine-tuning strategy. A positive value means that our method outperformed a simple fine-tuning strategy.

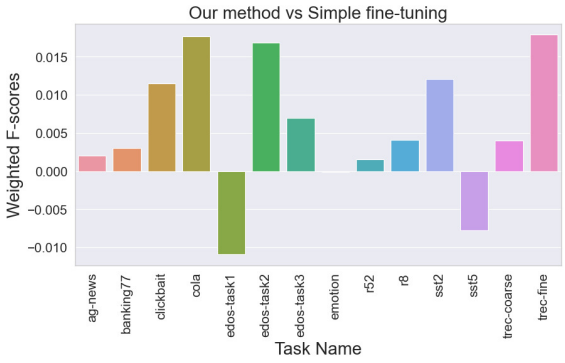

Figure 6: Difference in weighted F-scores between our method and a simple fine-tuning strategy. A positive value means that our method outperformed a simple fine-tuning strategy.

Characteristic	Pearson	
	macro F1	weighted F1
The number of samples in the dataset for the target task	-0.15	-0.19
The number of classes of the target task	-0.26	0.05
The entropy of the class distribution of the target task	-0.04	0.03
The average length of input texts of the target task	**-0.47**	**-0.21**
The average scores on the target task with a simple fine-tuning	-0.35	-0.08
The standard deviation of scores on the target task with a simple fine-tuning	**0.42**	**0.57**

Table 8: The Pearson correlation coefficients from some numerical characteristics and differences in scores between our method and simple fine-tuning strategy.

loss, demonstrating our intermediate task's effectiveness and computational efficiency. In addition, we applied our method to 14 NLP tasks to determine whether our method is effective for other tasks. Consequently, our methods outperformed a simple fine-tuning strategy on 10∼11 tasks, and we discovered that our method tends to be effective when a simple fine-tuning strategy does not stably work.

In future work, further analysis is required to clarify the effect of a sampling strategy for the intermediate task. In addition, we will consider employing a model that can process a longer text because truncation of an input text may cause performance degradation. The datasets used in this study are created from only papers in the field of natural language processing. However, results may differ when using URL citations in datasets from different domains because the form of URL citations may vary depending on the domain. Validation of our method in other domains using DBLP and Arxiv data remains one of the future work.

6. Ethical Consideration

The primary objective of this study is to classify URL citations in a paper based on their type and purpose. Resources cited by URLs are usually provided with a license for their use. Users are required to follow the license. The task we have undertaken in this study does not consider such licenses. In order to prevent license violations, it is necessary to list license information for practical use.

7. Acknowledgements

This research was partially supported by the Grant-in-Aid for Scientific Research (B) (No. 21H03773) of JSPS.

8. Bibliographical References

Ana Cardoso-Cachopo. 2007. Improving Methods for Single-label Text Categorization. PhD Thesis, Instituto Superior Tecnico, Universidade Tecnica de Lisboa.

Iñigo Casanueva, Tadas Temčinas, Daniela Gerz, Matthew Henderson, and Ivan Vulić. 2020. Efficient intent detection with dual sentence encoders. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, pages 38–45, Online. Association for Computational Linguistics.

Ting-Yun Chang and Chi-Jen Lu. 2021. Rethinking why intermediate-task fine-tuning works. In

Findings of the Association for Computational Linguistics: EMNLP 2021, pages 706–713, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Miryam de Lhoneux, Miguel Ballesteros, and Joakim Nivre. 2019. Recursive subtree composition in LSTM-based dependency parsing. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1566–1576, Minneapolis, Minnesota. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Maik Fröbe, Tim Gollub, Benno Stein, Matthias Hagen, and Martin Potthast. 2023. SemEval-2023 Task 5: Clickbait Spoiling. In *Proceedings of the 17th International Workshop on Semantic Evaluation*, pages 2278–2289, Toronto, Canada. Association for Computational Linguistics.

Pranshav Gajjar, Pooja Shah, Akash Vegada, and Jainish K. Savalia. 2022. Triplet loss for chromosome classification. *Journal of Innovative Image Processing*, 4(1):1–15.

Beliz Gunel, Jingfei Du, Alexis Conneau, and Veselin Stoyanov. 2021. Supervised contrastive learning for pre-trained language model fine-tuning. In *the Ninth International Conference on Learning Representations*.

Eduard Hovy, Laurie Gerber, Ulf Hermjakob, Chin-Yew Lin, and Deepak Ravichandran. 2001. Toward semantics-based answer pinpointing. In *Proceedings of the First International Conference on Human Language Technology Research*, pages 1–7, Stroudsburg, PA, USA. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv:1412.6980*.

Hannah Kirk, Wenjie Yin, Bertie Vidgen, and Paul Röttger. 2023. SemEval-2023 task 10: Explainable detection of online sexism. In *Proceedings of the 17th International Workshop on Semantic Evaluation*, pages 2193–2210, Toronto, Canada. Association for Computational Linguistics.

Xin Li and Dan Roth. 2002. Learning question classifiers. In *Proceedings of the 19th International Conference on Computational Linguistics*, pages 1–7, Taipei, Taiwan. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv:1907.11692*.

Min Sook Park and Hyoungjoo Park. 2019. An examination of metadata practices for research data reuse: Characteristics and predictive probability of metadata elements. *Malaysian Journal of Library & Information Science*, 24:61–75.

Jason Phang, Thibault Févry, and Samuel R Bowman. 2018. Sentence encoders on STILTs: Supplementary training on intermediate labeled-data tasks. *arXiv:1811.01088*.

Clifton Poth, Jonas Pfeiffer, Andreas Rücklé, and Iryna Gurevych. 2021. What to pre-train on? Efficient intermediate task selection. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10585–10605, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yada Pruksachatkun, Jason Phang, Haokun Liu, Phu Mon Htut, Xiaoyi Zhang, Richard Yuanzhe Pang, Clara Vania, Katharina Kann, and Samuel R. Bowman. 2020. Intermediate-task transfer learning with pretrained language models: When and why does it work? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5231–5247, Online. Association for Computational Linguistics.

Elvis Saravia, Hsien-Chi Toby Liu, Yen-Hao Huang, Junlin Wu, and Yi-Shin Chen. 2018. CARER: Contextualized affect representations for emotion recognition. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3687–3697, Brussels, Belgium. Association for Computational Linguistics.

Ozan Sener and Vladlen Koltun. 2018. Multi-task learning as multi-objective optimization. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 525–536, Red Hook, NY, USA. Curran Associates Inc.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep

models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Masaya Tsunokake and Shigeki Matsubara. 2022. Classification of URL citations in scholarly papers for promoting utilization of research artifacts. In *Proceedings of the First Workshop on Information Extraction from Scientific Publications*, pages 8–19, Online. Association for Computational Linguistics.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605.

Daniel Ponsa Vassileios Balntas, Edgar Riba and Krystian Mikolajczyk. 2016. Learning local feature descriptors with triplets and shallow convolutional neural networks. In *Proceedings of the British Machine Vision Conference*, pages 119.1–119.11. BMVA Press.

Alex Warstadt, Amanpreet Singh, and Samuel R Bowman. 2018. Neural network acceptability judgments. *arXiv:1805.12471*.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, pages 649–657, Cambridge, MA, USA. MIT Press.

Zhenyu Zhang, Yuming Zhao, Meng Chen, and Xiaodong He. 2022. Label anchored contrastive learning for language understanding. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1437–1449, Seattle, United States. Association for Computational Linguistics.

He Zhao, Zhunchen Luo, Chong Feng, Anqing Zheng, and Xiaopeng Liu. 2019. A context-based framework for modeling the role and function of on-line resource citations in scientific literature. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 5206–5215, Hong Kong, China. Association for Computational Linguistics.

Mengnan Zhao, Erjia Yan, and Kai Li. 2018. Data set mentions and citations: A content analysis of full-text publications. *Journal of the Association for Information Science and Technology*, 69(1):32–46.