An Automatic Labeling Method for Subword-Phrase Recognition in Effective Text Classification

Yusuke Kimura[1], Takahiro Komamizu[2] and Kenji Hatano[3]
[1]Graduate School of Culture and Information Science, Doshisha University, Japan
[2]Mathematical and Data Science Center, Nagoya University, Japan
[3]Faculty of Culture and Information Science, Doshisha University, Japan
E-mail: usk@acm.org, taka-coma@acm.org, hatano@acm.org

Keywords: text classification, subword-phrase, multi-task learning

Received: March 16, 2023

The deep learning-based text classification methods perform better than traditional ones. In addition to the success of the deep learning technique, multi-task learning (MTL) has come to become a promising approach for text classification; for instance, an MTL approach in text classification employs named entity recognition as an auxiliary task and has showcased that the task helps to improve the text classification performance. Existing MTL-based text classification methods depend on the auxiliary tasks using supervised labels. Obtaining such supervision labels requires additional human and financial costs in addition to those for the main text classification task. To reduce these additional costs, we propose an MTL-based text classification framework on supervised label creation by automatically labeling phrases in texts for the auxiliary recognition task. A basic idea to realize the proposed framework is to utilize phrasal expressions consisting of subwords (called subword-phrases). To the best of our knowledge, no text classification approach has been designed on top of subword-phrases because subwords only sometimes express a coherent set of meanings. The novelty of the proposed framework is in adding subword-phrase recognition as an auxiliary task and utilizing subword-phrases for text classification. It extracts subword-phrases in an unsupervised manner using the statistics approach. To construct labels for effective subword-phrase recognition tasks, extracted subword-phrases are classified based on document classes to ensure that subword-phrases dedicated to some classes can be distinguishable. Experimental evaluation for text classification using five popular datasets showcased the effectiveness of the subword-phrase recognition as an auxiliary task. It also showed that comparing various labeling schemes in recent studies indicated insights for labeling common subword-phrases among several document classes.

Povzetek: Za klasifikacijo besedil je uporabljeno globoko učenje in večopravilno učenje iz uporabo podbesednih fraz za avtomatsko označevanje.

1 Introduction

Text classification is a fundamental technology that has been studied for a long time. Applications that use text classification include speech [7], categorizing daily news articles, and unfair clause detection in terms of services [15]. These text classification applications are achieved by effectively and efficiently retrieving information from large amounts of text [12, 23]. Text classification is a supervised learning task manually assigning labels to documents as classification criteria, such as categories and classes. A classifier learns classification criteria in a feature space based on the dataset. Traditionally, text classification uses hand-crafted features such as term frequency-inverse document frequency. In recent literature, deep learning-based technologies have achieved significantly improved classification performance. A component that has improved text classification performance in recent years is pre-trained neural language models such as BERT, which have been trained on vast amounts of text. Pre-trained neural language models provide semantically rich features for text; therefore, even a simple multi-layer perceptron-based classifier has performs excellently. After the initial success of BERT, many pre-trained models, such as RoBERTa [19] and GPT-3 [5], have been published.

The tokenizers in these pre-trained neural language models typically divide documents into subwords as the smallest unit. Subwords reduce the number of unknown words not in the vocabulary, thus preventing the performance of pre-trained neural language models from being degraded by unknown words. Subword-based tokenization effectively handles out-of-vocabulary (OOV) words by decomposing such words into several subwords. Concatenations of these subwords represent OOV words, while traditional approaches represent them as *unknown* tokens. The subword-based tokenization was initially employed for machine translation [29]; after that, it was used in various natural language processing tasks, including text classification.

Multi-task learning (MTL) [6, 37, 39], which involves one or more auxiliary tasks with the primary task by sharing parameters, is a promising approach to enhance the performance of deep learning models. It has also been applied to text classification [17,35,36]. Learning models with auxiliary tasks positively affect the generalization performance of the main task and reduce over-fitting. Early studies on MTL-based text classification [17,35] focused on methods to combine multiple tasks and combined tasks in different datasets. Recent studies have combined text classification with auxiliary tasks using the same dataset, such as named entity recognition (NER) [2,31] or label co-occurrence prediction [36].

The fact that MTL with NER and text classification improves the accuracy of text classification performance suggests that the recognition of clause representations, such as named entities, is suitable as an auxiliary task to MTL-based text classification. However, to realize NER as an auxiliary task for MTL-based text classification, supervised labels for NER are required in addition to those for text classification. Constructing such training datasets is costly because of additional human costs for NER labeling.

Therefore, in this study, we seek to achieve MTL-based text classification with phrasal expression recognition, which does not require additional human cost to construct a training dataset. Phrasal expressions (or key phrases) for texts have been studied in past decades [27,38]. Applying keyphrase extraction based on the subword-based tokenization of popular pre-trained neural language models is not straightforward. Therefore, we define a phrasal expression based on subwords as a *subword-phrase* and seek its potential usability for the MTL-based text classification. In contrast to phrasal expressions based on words, subword-phrases are not necessarily semantically coherent because a vocabulary of subwords is determined statistically [29]. Owing to such little semantic coherence of subword-phrases, studies have never been conducted on their utilization for text classification.

In this study, we propose a framework for MTL-based text classification with subword-phrase recognition to improve the accuracy of text classification. Our framework comprises unsupervised subword-phrase labeling and MTL-based text classification for the subword-phrase recognition task. Notably, we assume the presence of labels for the classification of a dataset. To implement our framework, we employ a highly primitive approach: frequency-based subword-phrase labeling, in which frequently co-occurring consecutive subwords are merged to form a subword-phrase; various implementations can be realized using this approach. We also employ the concept of byte-pair encoding [29]. We seek labeling schemes to handle commonly appearing subword-phrases among document classes to make the auxiliary task more effective than text classification tasks.

The contributions of this study can be summarized as follows: MTL-based text classification with low-cost auxiliary task preparation, utilization of phrasal expression for subwords, and superior performance over conventional methods, and comparable performance with the novel methods. The proposed framework comprises an unsupervised labeling module and an MTL-based classification module. Existing MTL-based text classification methods assume the presence of supervision for auxiliary tasks; however, obtaining this supervision requires further human and financial costs. In contrast, the proposed framework does not require these costs as it utilizes unsupervised subword-phrase extraction to obtain labels to create auxiliary tasks.

Our method is the first study that utilizes subword-phrases. As subwords are not necessarily semantically coherent, their phrasal expressions have yet to be considered for any task. In contrast, the co-occurrence of consecutive subwords or subword-phrases could contribute to the text classification task. Such subwords may represent distinguished instances of a class from those of others. In the experimental evaluation of five popular text classification datasets, the proposed framework with subword-phrase recognition auxiliary task demonstrated improved classification performance (micro and macro F-scores) compared to the single-task method. Compared with the state-of-the-art method (BertGCN [14]), the proposed framework also demonstrated superior performance for datasets with more labels, exhibiting comparative classification performance for the other datasets.

The rest of this paper is organized as follows. Section 2 introduces studies concerning MTL-based text classification. Section 3 explains the proposed framework of MTL-based text classification with subword-phrase recognition task. Section 4 then presents the experimental evaluation, which demonstrates the effectiveness of the proposed framework compared to that of the single-task text classification baseline as well as other novel methods; it also discusses the effect of subword-phrases. Finally, Section 5 concludes this paper.

2 Related work

This section introduces literature related to MTL-based text classification. MTL-based text classification methods are categorized into the following three types based on the relationships between the main and auxiliary tasks [35]; Multi-Cardinality, Multi-Domain, and Multi-Objective.

Multi-Cardinality means that the main and auxiliary tasks are of different datasets but are in the same domain; these tasks also differ in cardinality, meaning that they vary in terms of their text lengths and the number of classes, among other parameters.

Multi-Domain means that the main and auxiliary tasks are similar, but their domains differ. For example, Liu et al. [16] and Zhang et al. [35] examined MTL-based movie review classification with classification tasks of reviews for various products, such as books and DVDs [4].

Multi-Objective means that the main and auxiliary tasks have different objectives. For example, Liu et al. [18] combined query classification and search result ranking using an MTL approach, and Zhang et al. [35] attempted MTL-based movie review classification (IMDB [21]) with news article classification (RN [1]) and question type classification (QC [13]) as auxiliary tasks.

In addition, MTL approaches [3, 30, 33, 40] in which the main and auxiliary tasks are in the same dataset have exhibited their effectiveness. Bi et al. [3] improved the performance of news recommendations by using MTL, which combines the news recommendation task with news article classification and named entity recognition. The MTL-based medical query intent classification model, proposed by Tohti et al. [30], was trained together with the named entity recognition, and consequently showed superior classification performance. On another task, Yang et al. [33] and Zhao et al. [40] showed similar observations on polarity classification combined with the aspect term extraction task. In the emotion prediction task, Li et al. [11] dealt with the emotion-cause pair extraction task using the MTL-based approach, which is combined with the emotion clause extraction and the cause clause extraction. Similarly, Qi et al. [24] proposed the MTL-based aspect sentiment classification method, where the auxiliary task was the aspect term extraction; they also demonstrated its effectiveness. In addition to the text classification task, the MTL-based approaches to image classification tasks have also shown its effectiveness [9, 32].

MTL-based text classification, which utilizes the relationship between labels in the same dataset, has also been proposed to solve the multi-label classification problem, where a single text can be classified into multiple labels [36]. Zhang et al. [36] showed improved classification performance by designing an auxiliary task to learn the relationship between labels.

These studies have shown the effectiveness of combining multiple supervised learning. However, in general, creating supervised data is expensive in terms of human and financial costs; thus, lower-cost solutions to design auxiliary tasks are desirable.

Self-supervised learning (SSL) is a training approach that understands data without supervised datasets. It first hides pieces of data and trains the model so that the model can estimate the hidden pieces. Masked language model (MLM) is a popular SSL in the natural language processing domain [8]. A popular pre-trained neural language model, BERT [8], is trained based on two SSL tasks: MLM and next sentence prediction. In the image processing domain, DALL-E [25] showcased the significant performance of SSL, where an area of an image was erased and DALL-E was trained to estimate the erased area. The increasing attention to these models indicates the usefulness of SSL for *data understanding* and *representation learning*.

In contrast to data understanding, text classification is a supervised learning task. In other words, SSL expects models to reconstruct broken pieces of data, while supervised learning expects models to learn dedicated criteria from supervision. Therefore, task settings in SSL are not easily imported to MTL-based text classification.

The proposed framework in this study focuses on creating datasets for auxiliary tasks with no supervision, significantly reducing human efforts and financial costs. To our knowledge, no research has been conducted that aimed to design auxiliary tasks of MTL-based text classification with no supervision. In addition, as subwords are not necessarily semantically coherent, subword-phrases have not been considered for any task. Therefore, this study proposes a novel methodology of MTL-based text classification in two aspects: In addition, since subwords are not necessarily semantically coherent, subword-phrases have not been considered for any task. Therefore, this paper proposes a novel methodology of MTL-based text classification in two aspects: (1) low-cost auxiliary task design and (2) introduction of subword-phrases. The experimental evaluation of this study reveals promising results for these two aspects.

3 Proposed framework

This section explains our framework of the MTL-based text classification, which generates subword-phrase labels for auxiliary tasks in an unsupervised manner.

3.1 Framework overview

Figure 1 illustrates our framework. It consists of two phases: unsupervised labeling and MTL-based text classification. The basic approach underlying of the framework is that subword-phrase recognition is added as an auxiliary task for MTL-based text classification. To realize the recognition task, unsupervised subword-phrase extraction is employed to create pseudo-supervision. A text classifier based on the framework is trained using the following steps:

1. **Input**: the text classifier receives a training set of text with classification labels;
2. **Tokenization**: the text is tokenized into subwords using a subword-based tokenizer;
3. **Labeling (Phase 1)**: the unsupervised labeling module appends subword-phrase labels to each text in the training set for the auxiliary subword-phrase recognition task;
4. **Training (Phase 2)**: the text classifier is trained in an MTL manner, which is trained together with the auxiliary subword-phrase recognition task based on the appended labels.

Formally, a training set is denoted as $D = \{(T_i, y_i) \mid 1 \leq i \leq N\}$, where T_i represents a sequence of subword tokens of the i-th text, y_i represents the class label corresponding to the i-th text, and N is the number of texts. In the first phase, the unsupervised subword-phrase labeling module receives D and performs subword-phrase extraction on subword token sequences to create another training set $D^{aug} = \{(T_i, Y_i^{aug}) \mid 1 \leq i \leq N\}$ for the auxiliary

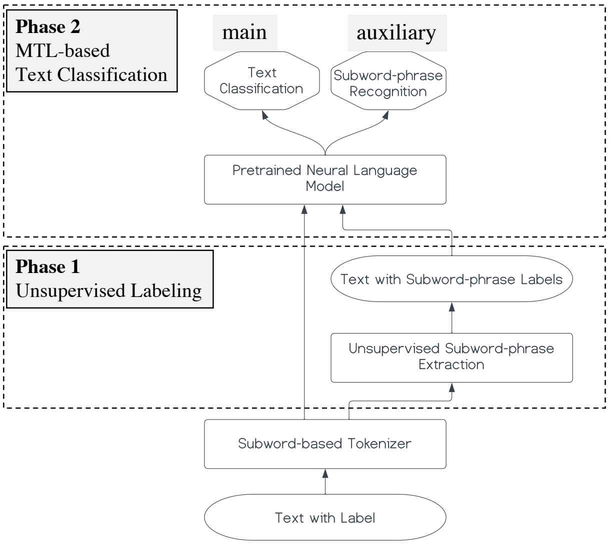

Figure 1: Our MTL-based Text Classification Framework. The framework accepts text with text classification labels and trains an MTL-based text classification model. The framework consists of two phases: the first phase is unsupervised labeling of the input text, and the second phase is the training of the MTL-based text classification model using the text classification labels and labels from the first phase.

task, where Y_i^{aug} is a corresponding sequence of labels for each token in T_i. In the second phase, D and D^{aug} are passed to an MTL-based text classification module based on a pre-trained neural language model; they then train the text classification model in conjunction with the training subword-phrase recognition model.

3.2 Unsupervised subword-phrase labeling

Unsupervised subword-phrase labeling provides a label sequence that corresponds to the input text sequence. This unsupervised labeling is a task formalized as follows:

- **Given**: a sequence of subword tokens T along with a class label y, $(T, y) \in D$
- **Generate**: a sequence of labels Y^{aug} whose length is exactly the same as that of T

The labeling scheme is inspired by NER tasks that employ the inside-outside-beginning (IOB2) tagging scheme [26]. IOB2 tagging is a labeling scheme where the first token of a phrase is tagged with `B` (beginning), the intermediate tokens of a phrase are tagged with `I` (inside), and tokens other than the phrase are tagged with `O` (outside). Besides these tags, semantic types are appended to distinguish types of phrases; for example, `B-PERSON` and `I-PERSON` represent the beginning and intermediate tokens of a token sequence corresponding with a person's name, respectively.

A straightforward labeling scheme for subword-phrase labeling is to treat all phrases equally. In other words, the semantic type is set to `Phrase`. Formally, when an n-length sequence of tokens $S = (s_1, s_2, \ldots, s_n)$ has a phrase which is an m-length sub-sequence $P = (s_k, s_{k+1}, \ldots, s_{k+m})$ of S where $m \leq n$, s_k is labeled as a particular type `B-Phrase`; the rest of the tokens from s_{k+1} to s_{k+m} are labeled as `I-Phrase` and other tokens $s_i \in S \backslash P$ are labeled as `O`.

This approach is so straightforward that subword-phrases appearing in different document classes are treated equally. However, to provide cues to the main text classification model, subword-phrases dependent on document classes should be distinguishable. A simple classification-specific labeling scheme assigns different labels to subword-phrases appearing in other classes. When a subword-phrase $P = (s_k, s_{k+1}, \ldots, s_{k+m})$, which is a sequence of tokens of a text belonging to class y, s_k is labeled as B-y, and the remaining tokens from s_{k+1} to s_{k+m} are labeled as I-y. However, subword-phrases commonly appearing in different classes cannot be handled in this scheme. To handle such common subword-phrases, we propose three labeling schemes, namely, **Disregard**, **Common-Label**, and **Bit-Label**. To compare, the aforementioned straightforward labelling scheme is called **All-Phrase**. Disregard scheme simply ignores the common subword-phrases, in other words, they are labeled by `O` tags. In Common-Label scheme, a special class label $\emptyset$ is used as a special semantic type of labeling in the IOB2 scheme. Specifically, the common subword-phrase P is labeled as B-$\emptyset$ for s_k and I-$\emptyset$ for other tokens. To handle such subword-phrases, this study proposes a bit-encoding-based labeling scheme. Bit-Label scheme still inherits the IOB2 labeling scheme; therefore, suppose that $d = 4$, a subword-phrase $P = (s_k, s_{k+1}, \ldots, s_{k+m})$, which is a sequence of tokens of a text and belongs to the first and third classes, then s_k is labeled as `B-1010`, and the rest of the tokens from s_{k+1} to s_{k+m} are labeled as `I-1010`.

3.3 MTL-based text classification

Our framework uses a text classification model based on MTL and a pre-trained neural language model (NLM). In this method, the NLM performs token encoding, and classification modules for main and auxiliary tasks are appended on top of the encoding. Therefore, NLM is the part shared among tasks and is trained in an MTL manner. A fully connected layer and a softmax non-linear layer design the classification models.

For the main task (i.e., text classification), a representation $\mathbf{h}^{cls}$ for a given input token sequence is obtained from NLM. It is passed to a fully connected layer followed by a softmax layer to predict class distribution $\hat{\mathbf{y}}^{cls}$. Formally, $\hat{\mathbf{y}}^{cls}$ for $\mathbf{h}^{cls}$ is calculated by the following equation:

$$\hat{\mathbf{y}}^{cls} = \mathrm{softmax}(W_{cls}^{\top} \cdot \mathbf{h}^{cls} + \mathbf{b}^{cls}), \tag{1}$$

where W_{cls} and $\mathbf{b}^{cls}$ denote the parameter matrix and bias, respectively, for the text classification task.

For the auxiliary tasks (i.e., subword-phrase recognition), a representation $\mathbf{h}_j^{spr}$ for the j-th token of a given input sequence is obtained from NLM. It is passed to a fully connected layer followed by a softmax layer to predict token label distribution $\hat{\mathbf{y}}^{spr}$. Formally, $\hat{\mathbf{y}}_j^{spr}$ for $\mathbf{h}_j^{spr}$ is calculated by the following equation:

$$\hat{\mathbf{y}}_j^{spr} = \mathrm{softmax}(W_{spr}^{\top} \cdot \mathbf{h}_j^{spr} + \mathbf{b}^{spr}), \tag{2}$$

where W_{spr} and $\mathbf{b}^{spr}$ denote the parameter matrix and bias, respectively, for the subword-phrase recognition task.

These main and auxiliary tasks are multi-class classification tasks; therefore, using the cross-entropy loss as a loss function is straightforward. The following equation calculates the loss L_{cls} for the text classification task:

$$L_{cls} = -\sum_{i=1}^{N} \sum_{c \in C} y_{i,c} \log \hat{y}_{i,c}^{cls}, \tag{3}$$

where N is the number of training sample texts, C denotes a set of classes, $y_{i,c} \in \{0, 1\}$ denotes a true label for the i-th text where $y_{i,c} = 1$ if the true label of the text is c and 0 otherwise, and $\hat{y}_{i,c}^{cls}$ denotes the predicted probability of class c for the text.

Similarly, the following equation calculates the loss L_{spr} for the subword-phrase recognition task:

$$L_{spr} = -\sum_{i=1}^{N} \sum_{j=1}^{M_i} \sum_{c \in C} y_{i,j,c} \log \hat{y}_{i,j,c}^{cls}, \tag{4}$$

where N denotes the number of training sample texts, M_i denotes the number of tokens in the i-th text, C denotes a set of classes, $y_{i,j,c} \in \{0, 1\}$ denotes a true label for the j-th token of the i-th text where $y_{i,j,c} = 1$ if the true label of the token is c and 0 otherwise, and $\hat{y}_{i,j,c}^{spr}$ denotes the predicted probability of class c for that token.

To train both tasks simultaneously, feedback from results on these tasks is fed to the NLM model to fine-tune its parameters. Therefore, joint loss L_{joint} of losses for these tasks are calculated using the following equation and used for parameter optimization.

$$L_{joint} = L_{cls} + L_{spr} \tag{5}$$

We note that the weighting scheme in MTL approaches to involve the importance of individual tasks has been studied [22, 28]. Although considering the weighting scheme in our framework is promising, the purpose of this study is to show the capability of MTL-based text classification in conjunction with subword-phrase recognition, whoselabels for auxiliary tasks are created in an unsupervised manner. Therefore, employing the weighting scheme in our framework can be the focus of future studies.

4 Experimental evaluation

To evaluate the proposed framework, we conducted an experimental evaluation to answer the following items: (1) Whether or not our MTL-based text classification methods that create auxiliary tasks in an unsupervised manner improve classification performance compared to single-task text classification methods?, (2) Whether or not our MTL-based text classification can outperform state-of-the-art (SOTA) text classification methods?, (3) Whether or not the subword-phrase technique contributes to text classification?, and (4) Whether or not there is the best labeling scheme for subword-phrase recognition in terms of common subword-phrases?

The rest of this section is organized as follows: Section 4.1 introduces the implementation of the proposed framework; Section 4.2 explains the SOTA text classification method for comparison; Section 4.3 describes the experimental settings; Section 4.4 showcases the experimental results, and Section 4.5 presents remarks on the experiments by answering items mentioned above.

4.1 Implementation of the proposed framework

In this experiment, we implemented a simple frequency-based subword-phrase extraction method; the labeling scheme used for the extracted subword-phrase was the classification-specific labeling scheme. The frequency-based method expects that frequently co-occurring subwords compose the regular textual expressions for each class. To control the number of subword-phrases, we utilized the byte-pair encoding (BPE) algorithm [29]. The BPE algorithm concatenates consecutive tokens if they frequently co-occur in a corpus and repeats this concatenation until the number of unique tokens equals the expected number. The ability to control the number of subword-phrases was suitable for this experiment because the subword-phrase was newly proposed in this study; therefore, we needed to try variations of evaluation experiments which were realized by creating different numbers of subword-phrases.

In general, the number of texts is skewed among classes; the number of particular texts of a class may be quite large, while that of other classes is very small. This affected the extraction of subword-phrases; therefore, in this experiment, the extraction mentioned above was applied for each set of texts of class. Specifically, we extracted n subword-phrases for each class. n was chosen from $\{10, 100, 1000, 10000\}$ to achieve the best classification performance on the validation data.

4.2 Comparison method: BertGCN

BertGCN [14] is a SOTA method for text classification that combines a pre-trained NLM with the inductive learning of graph neural networks (GNNs). BertGCN follows TextGCN [34] by constructing a graph of the cooccurrence relations between texts and words and between words and words. In BertGCN, vectors of vertices are initialized using the pre-trained NLM. These vectors are updated through graph convolutional neural network (GCN) to involve the co-occurrence relationships between texts and words. Based on the updated vectors, BertGCN performs text classification by adding a fully connected layer followed by a softmax layer. In addition, [14] reported that integrating the output of the NLM-based classification model and that of BertGCN can improve classification performance; specifically, the linear sum of the predicted class distributions Z_{GCN} and Z_{NLM}, which are obtained from BertGCN and the classifier using NLM, respectively, as seen in the following equation:

$$Z = \lambda \cdot Z_{\mathrm{GCN}} + (1-\lambda) \cdot Z_{\mathrm{NLM}}, \tag{6}$$

where $\lambda \in [0,1]$ denotes the weight for BertGCN classification. This experiment used $\lambda = 0.7$ as [14] reported that it was the optimal value. BertGCN can use any pre-trained NLM, and [14] reported that RoBERTa showed the optimal performance. Therefore, RoBERTa was also used in implementing the proposed framework to make the comparison as reasonable as possible.

4.3 Settings

Datasets For the evaluation, the following five popular datasets in the text classification task are used; Movie Review (MR), 20 Newsgroups (20NG), R8, R52 and Ohsumed (OHS). MR is a dataset of movie reviews categorized into binary sentiment classes (i.e., positive and negative). 20NG is a dataset of news texts categorized into 20 categories. R8 is a dataset of news articles from Reuters-21578[1] limited to eight selected classes. R52 is a dataset of news articles from Reuters-21578 limited to 52 selected categories. OHS is a dataset of medical abstracts categorized into 23 medical concepts called MESH categories.

The statistics of the dataset are shown in Table 1. As the table shows, datasets with different classes and variations in the number of instances per class (the standard deviation (Std.) of the number of instances within a class) were used in the experiment. These datasets were expected to reveal the advantages and disadvantages of the proposed method.

Metrics The evaluation metric is F-score which is the harmonic mean of precision and recall scores as shown below.

$$\begin{aligned} Pre &= \frac{TP}{TP+FP} & (7)\\ Rec &= \frac{TP}{TP+FN} & (8)\\ F &= \frac{2 \cdot Pre \cdot Rec}{Prec + Rec} & (9) \end{aligned}$$

The precision, denoted by *Pre* is the ratio of the number of true positives (TP) over the number of instances estimated as positive (i.e., $TP + FP$, where FP is the number of false positives). The recall, denoted by *Rec* is the ratio of TP over the number of positive instances in the evaluation set (i.e., $TP+FN$, where FN is the number of false negatives). To observe various aspects for evaluation, micro and macro averages of F-scores were used in this experiment. The micro average of F-scores, F_{micro}, is the instance-level average of the F-score, and the macro average, F_{macro}, is the class-level average of the F-scores. When the numbers of instances of different classes are highly skewed (class imbalance problem), the F_{micro} is not suitable to evaluate the classification performance; this is because the larger the number of instances of a class, the more it affects this metric. In other words, the classification performance in the instances of minority classes is underestimated. In contrast, the F_{macro} metric can ignore the skewness as the F scores of difference classes are treated independently and averaged.

Parameters For the base model in the proposed method and BertGCN, we employed the RoBERTa-base model [19], available at Huggingface[2]. BertGCN with the RoBERTa model was called RoBERTaGCN in this experiment. In this study, the effect of common subword-phrases was also evaluated; therefore, the proposed method had two variations: one included common subword-phrases (denoted as Proposed w/ cmn) and the other excluded them (denoted as Proposed w/o cmn). In addition, as a baseline method, we also employed a single-task text classification method based on RoBERTa. The baseline method was implemented by adding a fully connected layer and a softmax layer on top of RoBERTa, which is equivalent to Eq. 1 with the loss function shown in Eq. 3. The only difference between the proposed and the baseline methods was the number of tasks on top of RoBERTa. Therefore, the comparison between them was expected to reveal the effectiveness of MTL-based text classification. These models were optimized using the AdamW optimizer (Adam optimizer [10] with decoupled weight decay regularization) [20]. Experiments were conducted with 100 epochs, batch size 64, and a maximum token length of 256. Only the experiment for RoBERTaGCN was conducted with a batch size of 128 and a maximum token length of 128, which yielded better results than the aforementioned hyper parameters.

4.4 Results

Table 2 shows the experimental results of F_{micro} (Table 2(a)) and F_{macro} (Table 2(b)), and showcases the following three observations. (1) The proposed method performed better than the baseline method in both metrics except the simple binary classification on the MR dataset. (2) The proposed method outperformed RoBERTaGCN for three of the five datasets in terms of the F_{micro} metric and four of the five datasets in terms of the F_{macro} metric. (3) In terms of labeling schemes, the Bit-Label and the Disregard approaches

[1] Reuters-21578, `http://www.daviddlewis.com/resources/testcollections/reuters21578/`, visited on Aug. 4, 2022

[2] `https://huggingface.co/roberta-base`

Table 1: Statistics of datasets. The number of instances in train-valid-test splits, number of classes, and average (Avg.) and standard deviation (Std.) of the number of instances across classes.

	MR	20NG	R8	R52	OHS
#Train	6,398	10,183	4,937	5,879	3,022
#Valid	710	1,131	548	653	335
#Test	3,554	7,532	2,189	2,568	4,043
#Class	2	20	8	52	23
Avg. #Instances/Class	5,331	942	959	175	321
Std. #Instances/Class	0	94	1,309	613	305

Table 2: Evaluation results. The best score in each column (i.e., dataset) is bold-faced. RoBERTaGCN is the SOTA text classification method and Baseline is the single-task text classification based on the RoBERTa model. The proposed method has two variations: one, denoted as Proposed w/ cmn, includes common subword-phrases in the labeling scheme, and the other, denoted as Proposed w/o cmn, excludes them. (a) and (b) showcase the results of F_{micro} and F_{macro}, respectively.

(a) F_{micro}

Model	MR	20NG	R8	R52	OHS
RoBERTaGCN	0.880	**0.894**	**0.979**	0.944	**0.736**
Baseline (RoBERTa)	0.881	0.831	0.977	0.962	0.690
Proposed - All-Phrase	**0.888**	0.838	**0.979**	0.967	0.705
Proposed - Common-Label	0.860	0.850	0.978	0.967	0.704
Proposed - Bit-Label	0.882	0.846	**0.979**	0.968	0.711
Proposed - Disregard	0.866	0.851	**0.979**	**0.969**	0.711

(b) F_{macro}

Model	MR	20NG	R8	R52	OHS
RoBERTaGCN	0.880	**0.861**	0.925	0.756	0.605
Baseline (RoBERTa)	0.881	0.825	0.943	0.836	0.594
Proposed - All-Phrase	**0.888**	0.832	0.948	0.842	0.622
Proposed - Common-Label	0.860	0.845	0.947	0.841	0.610
Proposed - Bit-Label	0.882	0.840	0.953	**0.866**	0.636
Proposed - Disregard	0.866	0.845	**0.955**	0.851	**0.637**

performed better than other schemes in terms of the F_{macro} metric.

The comparison between the proposed method and the baseline method in both F_{micro} and F_{macro} revealed the effectiveness of the MTL-based approach, in which the auxiliary task was systematically constructed. In addition to insights from existing literature that MTL-based approaches using auxiliary tasks with supervision are effective, this experiment showcased the effectiveness of an MTL approach in which training data for an auxiliary task was generated in an unsupervised manner. The results showcase that low-cost auxiliary tasks for MTL-based text classification now demonstrate promising performance.

While the results of MR and R8 datasets showed comparable performances between the proposed and the baseline methods, these datasets were composed of smaller numbers of classes. These results suggest that the proposed method did not perform effectively when the number of classes was small.

A notable fact from the results was the proposed method achieved significantly better performance than RoBERTaGCN in terms of F_{macro} on the R8, R52, and OHS datasets. Simultaneously, the proposed method was also more accurate than RoBERTaGCN in terms of F_{micro}. These facts indicate that the proposed method achieved state-of-the-art classification performance on these datasets. Recalling the statistics of these datasets from Table 1, the numbers of classes in each R8, R52 and OHS dataset are larger than those of other datasets and the number of instances per class is highly skewed. These facts indicate that the proposed method is good for highly skewed datasets. Though 20NG dataset had similar number of classes to the OHS dataset and was less skewed than the OHS dataset, the performance in terms of F_{micro} and F_{macro} of the proposed

Table 3: Evaluation results: Accuracy of auxiliary tasks

(a) F_{micro}

Model	MR	20NG	R8	R52	OHS
Proposed - All-Phrase	**0.971**	**0.975**	**0.978**	**0.998**	0.971
Proposed - Common-Label	0.922	0.968	0.974	0.972	**0.978**
Proposed - Bit-Label	0.918	0.974	0.965	0.975	0.977
Proposed - Disregard	0.922	0.851	0.962	0.975	**0.978**

(b) F_{macro}

Model	MR	20NG	R8	R52	OHS
Proposed - All-Phrase	**0.960**	**0.975**	**0.945**	0.796	**0.953**
Proposed - Common-Label	0.761	0.889	0.869	0.853	0.725
Proposed - Bit-Label	0.756	0.852	0.764	**0.864**	0.762
Proposed - Disregard	0.761	0.845	0.731	0.847	0.725

method was worse than RoBERTaGCN. Consequently, the proposed method performed better than the SOTA method when datasets were composed of large classes and highly skewed in the number of instances across classes.

The comparison among variations of the proposed method in terms of the labeling schemes for commonly appearing subword-phrases among document classes showed that the proposed method with different schemes had similar performances, each with their pros and cons for different datasets. The All-Phrase scheme had all phrases labeled by the IOB2 tagging scheme regardless of document classes. Compared with other schemes that take document classes into account, its performance was inferior. This indicates that class-specific labeling (the Common-Label, Bit-Label, and Disregard schemes) is effective, except for the MR dataset, which is a binary classification dataset; thus, subword-phrases are merely *class-specific*. For the comparison of labeling common subword-phrases among the Common-Label, Bit-Label, and Disregard schemes, their classification performances were comparable, and the Disregard scheme had relatively better performance.

To show the difficulties of subword-phrase recognition tasks with different labeling schemes, Table 3 displays the F scores of the auxiliary tasks. In general, the number of classes in a sequence labeling problem is related to its difficulty. Thus, the All-Phrase scheme was expected to be the easiest and the Bit-Label scheme the most difficult. As shown in the results in the table, the F scores of the All-Phrase scheme are the highest among these schemes, thereby confirming their easiness in terms of a sequence labeling problem. In contrast, F scores of the other schemes were inferior, but still high enough to aid the generalization performance of the main text classification model.

4.5 Remarks

This section summarizes the findings from our experiment by answering the abovementioned items and introduces the limitations of the proposed method.

(1) The proposed method outperformed the baseline method when the number of classes of a dataset was large and was comparable to them when the number was small. However, datasets with a few classes were also less skewed in the number of instances per class. Therefore, the frequency-based subword-phrase extraction for constructing auxiliary tasks was suitable when datasets had many classes, and the number of instances per class was skewed. A promising outcome is that an auxiliary recognition task in which (pseudo) supervision is generated unsupervised is effective in the MTL-based classification. Therefore, this outcome opens up new possibilities for constructing auxiliary tasks for the MTL-based classification methods on tasks other than text classification.

(2) The proposed method was superior to the SOTA method, RoBERTaGCN, for the R52 and OHS datasets, which contained many classes and where the number of instances per class was skewed. A promising direction to overcome the inferiority of the proposed method in the other datasets is to utilize RoBERTaGCN as a base model for the proposed method.

(3) The subword-phrase recognition task as an auxiliary task improves text classifications in various datasets. A promising outcome is the usage of phrasal expressions for subwords, which which needs more attention in the literature.

(4) To handle common subword-phrases among document classes, the Bit-Label scheme, which encodes dependence of subword-phrases in a bit sequence that can represent all combinations of appearing classes, and the Disregard scheme, which ignores common subword-phrases, were the best. The higher the number of classes (e.g., R52), the better the classification performance using the Bit-Label scheme. Contrastingly, the smaller the number of classes (e.g., R8 and OHS), the better the Disregard scheme performance.

Consequently, when the number of classes is large, and the number of instances for document classes is skewed, the MTL-based text classification suffers from the class imbalance problem, which is still an open problem in the general text classification tasks domain. This domain showcases some promising results by using subword-phrase recognition tasks, whose labels are obtained in an unsupervised manner. However, at the same time, the classification performance still leaves a lot to be desired. Therefore, future studies should seek more effective auxiliary tasks to deal with the class imbalance problem.

5 Conclusion

We proposed an MTL-based text classification framework using auxiliary tasks with lower human and financial costs by creating auxiliary task labels unsupervised. We also sought to ascertain the possibility of phrasal expressions of subwords called subword-phrases to utilize subword-based neural language pre-trained models. As an implementation of our framework, we extracted subword-phrases in terms of their frequency of occurrence and labeled them into documents in three different ways. Our experimental evaluation for text classification using five popular datasets highlighted the effectiveness of the subword-phrase recognition as an auxiliary task. It also showed comparative results with RoBERTaGCN which is the state-of-the-art method.

The main conclusions of this paper are: an auxiliary recognition task in which pseudo supervision is generated in an unsupervised manner is effective in MTL-based classification, and opens up the possibility of constructing auxiliary tasks for MTL-based classification methods for classification tasks other than text classification, and phrasal expressions for subwords (subword-phrase) can be helpful in text classification.

Acknowledgment

This work was partly supported by the Grants-in-Aid for Academic Promotion, Graduate School of Culture and Information Science, Doshisha University, JSPS KAKENHI Grant Number 19H01138, 19H04218, and 21H03555, and JST, the establishment of university fellowships towards the creation of science and technology innovation, Grant Number JPMJFS2145.

References

[1] C. Apté, F. Damerau, and S. M. Weiss. Automated Learning of Decision Rules for Text Categorization. *ACM Transactions on Information Systems*, 12(3):233–251, 1994.

[2] A. Benayas, R. Hashempour, D. Rumble, S. Jameel, and R. C. De Amorim. Unified Transformer Multi-Task Learning for Intent Classification With Entity Recognition. *IEEE Access*, 9:147306–147314, 2021.

[3] Q. Bi, J. Li, L. Shang, X. Jiang, Q. Liu, and H. Yang. MTRec: Multi-Task Learning over BERT for News Recommendation. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2663–2669, May 2022.

[4] J. Blitzer, M. Dredze, and F. Pereira. Biographies, Bollywood, Boom-boxes and Blenders: Domain Adaptation for Sentiment Classification. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 440–447, 2007.

[5] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems*, 2020.

[6] R. Caruana. Multitask Learning. *Machine Learning*, 28(1):41–75, 1997.

[7] O. de Gibert, N. Pérez, A. G. Pablos, and M. Cuadros. Hate Speech Dataset from a White Supremacy Forum. In *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*, pages 11–20, 2018.

[8] J. Devlin, M. Chang, K. Lee, and K. Toutanova. 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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 2 (Industry Papers)*, pages 4171–4186, 2019.

[9] S. Graham, Q. D. Vu, M. Jahanifar, S. Raza, F. A. Afsar, D. R. J. Snead, and N. M. Rajpoot. One model is all you need: Multi-task learning enables simultaneous histology image segmentation and classification. *Medical Image Analysis*, 83:102685, 2023.

[10] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations*, 2015.

[11] C. Li, J. Hu, T. Li, S. Du, and F. Teng. An effective multi-task learning model for end-to-end emotion-cause pair extraction. *Applied Intelligence*, 53(3):3519–3529, 2023.

[12] Q. Li, H. Peng, J. Li, C. Xia, R. Yang, L. Sun, P. S. Yu, and L. He. A Survey on Text Classification: From Traditional to Deep Learning. *ACM Transactions on Intelligent Systems and Technology*, 13(2):31:1–31:41, 2022.

[13] X. Li and D. Roth. Learning Question Classifiers. In *COLING 2002: The 19th International Conference on Computational Linguistics*, 2002.

[14] Y. Lin, Y. Meng, X. Sun, Q. Han, K. Kuang, J. Li, and F. Wu. BertGCN: Transductive Text Classification by Combining GNN and BERT. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1456–1462, Online, Aug. 2021.

[15] M. Lippi, P. Palka, G. Contissa, F. Lagioia, H. Micklitz, G. Sartor, and P. Torroni. CLAUDETTE: an automated detector of potentially unfair clauses in online terms of service. *Artifcial Intelligence and Law*, 27(2):117–139, 2019.

[16] P. Liu, X. Qiu, and X. Huang. Deep Multi-Task Learning with Shared Memory for Text Classification. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 118–127, 2016.

[17] P. Liu, X. Qiu, and X. Huang. Recurrent Neural Network for Text Classification with Multi-Task Learning. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, pages 2873–2879, 2016.

[18] X. Liu, J. Gao, X. He, L. Deng, K. Duh, and Y. Wang. Representation Learning Using Multi-Task Deep Neural Networks for Semantic Classification and Information Retrieval. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 912–921, 2015.

[19] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *CoRR*, abs/1907.11692, 2019.

[20] I. Loshchilov and F. Hutter. Decoupled Weight Decay Regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

[21] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150. Association for Computational Linguistics, 2011.

[22] Y. Mao, Z. Wang, W. Liu, X. Lin, and P. Xie. MetaWeighting: Learning to Weight Tasks in Multi-Task Learning. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3436–3448, 2022.

[23] S. Minaee, N. Kalchbrenner, E. Cambria, N. Nikzad, M. Chenaghlu, and J. Gao. Deep Learning-based Text Classification: A Comprehensive Review. *ACM Computing Surveys*, 54(3):62:1–62:40, 2021.

[24] R. Qi, M. Yang, Y. Jian, Z. Li, and H. Chen. A Local context focus learning model for joint multi-task using syntactic dependency relative distance. *Applied Intelligence*, 53(4):4145–4161, 2023.

[25] A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, and I. Sutskever. Zero-Shot Text-to-Image Generation. In *Proceedings of the 38th International Conference on Machine Learning*, pages 8821–8831, 2021.

[26] L. Ramshaw and M. Marcus. Text Chunking using Transformation-Based Learning. In *Third Workshop on Very Large Corpora*, 1995.

[27] F. Sebastiani. Machine Learning in Automated Text Categorization. *ACM computing surveys*, 34(1):1–47, mar 2002.

[28] O. Sener and V. Koltun. Multi-Task Learning as Multi-Objective Optimization. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 525–536, 2018.

[29] R. Sennrich, B. Haddow, and A. Birch. Neural Machine Translation of Rare Words with Subword Units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany, Aug. 2016. Association for Computational Linguistics.

[30] T. Tohti, M. Abdurxit, and A. Hamdulla. Medical QA Oriented Multi-Task Learning Model for Question Intent Classification and Named Entity Recognition. *Information*, 13(12):581, 2022.

[31] C. Wu, G. Luo, C. Guo, Y. Ren, A. Zheng, and C. Yang. An attention-based multi-task model for named entity recognition and intent analysis of Chinese online medical questions. *Journal of Biomedical Informatics*, 108:103511, 2020.

[32] M. Xu, K. Huang, and X. Qi. A Regional-Attentive Multi-Task Learning Framework for Breast Ultrasound Image Segmentation and Classification. *IEEE Access*, 11:5377–5392, 2023.

[33] H. Yang, B. Zeng, J. Yang, Y. Song, and R. Xu. A multi-task learning model for Chinese-oriented aspect polarity classification and aspect term extraction. *Neurocomputing*, 419:344–356, 2021.

[34] L. Yao, C. Mao, and Y. Luo. Graph Convolutional Networks for Text Classification. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence*, pages 7370–7377, 2019.

[35] H. Zhang, L. Xiao, Y. Wang, and Y. Jin. A Generalized Recurrent Neural Architecture for Text Classification with Multi-Task Learning. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, pages 3385–3391, 2017.

[36] X. Zhang, Q. Zhang, Z. Yan, R. Liu, and Y. Cao. Enhancing Label Correlation Feedback in Multi-Label Text Classification via Multi-Task Learning. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1190–1200. Association for Computational Linguistics, 2021.

[37] Y. Zhang and Q. Yang. A Survey on Multi-Task Learning. *IEEE Transactions on Knowledge and Data Engineering*, 34(12):5586–5609, 2022.

[38] Y. Zhang, N. Zincir-Heywood, and E. Milios. Narrative Text Classification for Automatic Key Phrase Extraction in Web Document Corpora. In *Proceedings of the 7th Annual ACM International Workshop on Web Information and Data Management*, WIDM '05, page 51–58, 2005.

[39] Z. Zhang, W. Yu, M. Yu, Z. Guo, and M. Jiang. A Survey of Multi-task Learning in Natural Language Processing: Regarding Task Relatedness and Training Methods. *CoRR*, abs/2204.03508, 2022.

[40] M. Zhao, J. Yang, and L. Qu. A multi-task learning model with graph convolutional networks for aspect term extraction and polarity classification. *Applied Intelligence*, 53(6):6585–6603, 2023.