Text Classification Using Label Names Only: A Language Model Self-Training Approach

Yu Meng[1], **Yunyi Zhang**[1], **Jiaxin Huang**[1], **Chenyan Xiong**[2],
Heng Ji[1], **Chao Zhang**[3], **Jiawei Han**[1]
[1]University of Illinois at Urbana-Champaign, IL, USA
[2]Microsoft Research, WA, USA [3]Georgia Institute of Technology, GA, USA
[1]{yumeng5, yzhan238, jiaxinh3, hengji, hanj}@illinois.edu
[2]chenyan.xiong@microsoft.com [3]chaozhang@gatech.edu

Abstract

Current text classification methods typically require a good number of human-labeled documents as training data, which can be costly and difficult to obtain in real applications. Humans can perform classification without seeing any labeled examples but only based on a small set of words describing the categories to be classified. In this paper, we explore the potential of only using the label name of each class to train classification models on unlabeled data, without using any labeled documents. We use pre-trained neural language models both as general linguistic knowledge sources for category understanding and as representation learning models for document classification. Our method (1) associates semantically related words with the label names, (2) finds category-indicative words and trains the model to predict their implied categories, and (3) generalizes the model via self-training. We show that our model achieves around 90% accuracy on four benchmark datasets including topic and sentiment classification without using any labeled documents but learning from unlabeled data supervised by at most 3 words (1 in most cases) per class as the label name[1].

1 Introduction

Text classification is a classic and fundamental task in Natural Language Processing (NLP) with a wide spectrum of applications such as question answering (Rajpurkar et al., 2016), spam detection (Jindal and Liu, 2007) and sentiment analysis (Pang et al., 2002). Building an automatic text classification model has been viewed as a task of training machine learning models from human-labeled documents. Indeed, many deep learning-based classifiers including CNNs (Kim, 2014; Zhang et al., 2015) and RNNs (Tang et al., 2015a; Yang et al., 2016) have been developed and achieved great success when trained on large-scale labeled documents (usually over tens of thousands), thanks to their strong representation learning power that effectively captures the high-order, long-range semantic dependency in text sequences for accurate classification.

Recently, increasing attention has been paid to semi-supervised text classification which requires a much smaller amount of labeled data. The success of semi-supervised methods stems from the usage of abundant unlabeled data: Unlabeled documents provide natural regularization for constraining the model predictions to be invariant to small changes in input (Chen et al., 2020; Miyato et al., 2017; Xie et al., 2019), thus improving the generalization ability of the model. Despite mitigating the annotation burden, semi-supervised methods still require manual efforts from domain experts, which might be difficult or expensive to obtain especially when the number of classes is large.

Contrary to existing supervised and semi-supervised models which learn from labeled documents, a human expert will just need to understand the label name (*i.e.*, a single or a few representative words) of each class to classify documents. For example, we can easily classify news articles when given the label names such as "sports", "business", and "politics" because we are able to understand these topics based on prior knowledge.

In this paper, we study the problem of weakly-supervised text classification where only the label name of each class is provided to train a classifier on purely unlabeled data. We propose a language model self-training approach wherein a pre-trained neural language model (LM) (Devlin et al., 2019; Peters et al., 2018; Radford et al., 2018; Yang et al., 2019) is used as both the general knowledge source for category understanding and feature representation learning model for classification. The LM

[1]Source code can be found at https://github.com/yumeng5/LOTClass.

creates contextualized word-level category supervision from unlabeled data to train itself, and then generalizes to document-level classification via a self-training objective.

Specifically, we propose the **LOTClass** model for **L**abel-Name-**O**nly **T**ext **Class**ification built in three steps: (1) We construct a category vocabulary for each class that contains semantically correlated words with the label name using a pre-trained LM. (2) The LM collects high-quality category-indicative words in the unlabeled corpus to train itself to capture category distinctive information with a contextualized word-level category prediction task. (3) We generalize the LM via document-level self-training on abundant unlabeled data.

LOTClass achieves around 90% accuracy on four benchmark text classification datasets, *AG News*, *DBPedia*, *IMDB* and *Amazon* corpora, *without* learning from any labeled data but only using at most 3 words (1 word in most cases) per class as the label name, outperforming existing weakly-supervised methods significantly and yielding even comparable performance to strong semi-supervised and supervised models.

The contributions of this paper are as follows:

- We propose a weakly-supervised text classification model **LOTClass** based on a pre-trained neural LM without any further dependencies[2]. **LOTClass** does not need any labeled documents but only the label name of each class.
- We propose a method for finding category-indicative words and a contextualized word-level category prediction task that trains LM to predict the implied category of a word using its contexts. The LM so trained generalizes well to document-level classification upon self-training on unlabeled corpus.
- On four benchmark datasets, **LOTClass** outperforms significantly weakly-supervised models and has comparable performance to strong semi-supervised and supervised models.

2 Related Work

2.1 Neural Language Models

Pre-training deep neural models for language modeling, including autoregressive LMs such as ELMo (Peters et al., 2018), GPT (Radford et al., 2018) and XLNet (Yang et al., 2019) and autoencoding LMs such as BERT (Devlin et al., 2019) and its variants (Lan et al., 2020; Lewis et al., 2020; Liu et al., 2019b), has brought astonishing performance improvement to a wide range of NLP tasks, mainly for two reasons: (1) LMs are pre-trained on large-scale text corpora, which allow the models to learn generic linguistic features (Tenney et al., 2019) and serve as knowledge bases (Petroni et al., 2019); and (2) LMs enjoy strong feature representation learning power of capturing high-order, long-range dependency in texts thanks to the Transformer architecture (Vaswani et al., 2017).

2.2 Semi-Supervised and Zero-Shot Text Classification

For semi-supervised text classification, two lines of framework are developed to leverage unlabeled data. Augmentation-based methods generate new instances and regularize the model's predictions to be invariant to small changes in input. The augmented instances can be either created as real text sequences (Xie et al., 2019) via back translation (Sennrich et al., 2016) or in the hidden states of the model via perturbations (Miyato et al., 2017) or interpolations (Chen et al., 2020). Graph-based methods (Tang et al., 2015b; Zhang et al., 2020) build text networks with words, documents and labels and propagate labeling information along the graph via embedding learning (Tang et al., 2015c) or graph neural networks (Kipf and Welling, 2017).

Zero-shot text classification generalizes the classifier trained on a known label set to an unknown one without using any new labeled documents. Transferring knowledge from seen classes to unseen ones typically relies on semantic attributes and descriptions of all classes (Liu et al., 2019a; Pushp and Srivastava, 2017; Xia et al., 2018), correlations among classes (Rios and Kavuluru, 2018; Zhang et al., 2019) or joint embeddings of classes and documents (Nam et al., 2016). However, zero-shot learning still requires labeled data for the seen label set and cannot be applied to cases where no labeled documents for any class is available.

2.3 Weakly-Supervised Text Classification

Weakly-supervised text classification aims to categorize text documents based only on word-level descriptions of each category, eschewing the need of any labeled documents. Early attempts rely on distant supervision such as Wikipedia to interpret

[2]Other semi-supervised/weakly-supervised methods usually take advantage of distant supervision like Wikipedia dump (Chang et al., 2008), or augmentation systems like trained back translation models (Xie et al., 2019).

the label name semantics and derive document-concept relevance via explicit semantic analysis (Gabrilovich and Markovitch, 2007). Since the classifier is learned purely from general knowledge without even requiring any unlabeled domain-specific data, these methods are called dataless classification (Chang et al., 2008; Song and Roth, 2014; Yin et al., 2019). Later, topic models (Chen et al., 2015; Li et al., 2016) are exploited for seed-guided classification to learn seed word-aware topics by biasing the Dirichlet priors and to infer posterior document-topic assignment. Recently, neural approaches (Mekala and Shang, 2020; Meng et al., 2018, 2019) have been developed for weakly-supervised text classification. They assign documents pseudo labels to train a neural classifier by either generating pseudo documents or using LMs to detect category-indicative words. While achieving inspiring performance, these neural approaches train classifiers from scratch on the local corpus and fail to take advantage of the general knowledge source used by dataless classification. In this paper, we build our method upon pre-trained LMs, which are used both as general linguistic knowledge sources for understanding the semantics of label names, and as strong feature representation learning models for classification.

3 Method

In this section, we introduce **LOTClass** with BERT (Devlin et al., 2019) as our backbone model, but our method can be easily adapted to other pre-trained neural LMs.

3.1 Category Understanding via Label Name Replacement

When provided label names, humans are able to understand the semantics of each label based on general knowledge by associating with it other correlated keywords that indicate the same category. In this section, we introduce how to learn a category vocabulary from the label name of each class with a pre-trained LM, similar to the idea of topic mining in recent studies (Meng et al., 2020a,b).

Intuitively, words that are interchangeable most of the time are likely to have similar meanings. We use the pre-trained BERT masked language model (MLM) to predict what words can replace the label names under most contexts. Specifically, for each occurrence of a label name in the corpus, we feed its contextualized embedding vector $\boldsymbol{h} \in \mathbb{R}^h$ produced by the BERT encoder to the MLM head, which will output a probability distribution over the entire vocabulary V, indicating the likelihood of each word w appearing at this position:

$$p(w \mid \boldsymbol{h}) = \text{Softmax}\left(W_2\, \sigma\left(W_1 \boldsymbol{h} + \boldsymbol{b}\right)\right), \quad (1)$$

where $\sigma(\cdot)$ is the activation function; $W_1 \in \mathbb{R}^{h \times h}$, $\boldsymbol{b} \in \mathbb{R}^h$, and $W_2 \in \mathbb{R}^{|V| \times h}$ are learnable parameters that have been pre-trained with the MLM objective of BERT.

Table 1 shows the pre-trained MLM prediction for the top words (sorted by $p(w \mid \boldsymbol{h})$) to replace the original label name "sports" under two different contexts. We observe that for each masked word, the top-50 predicted words usually have similar meanings with the original word, and thus we use the threshold of 50 words given by the MLM to define valid replacement for each occurrence of the label names in the corpus. Finally, we form the category vocabulary of each class using the top-100 words ranked by how many times they can replace the label name in the corpus, discarding stopwords with NLTK (Bird et al., 2009) and words that appear in multiple categories. Tables 2, 3, 4 and 9 (Table 9 is in Appendix B) show the label name used for each category and the obtained category vocabulary of *AG News*, *IMDB*, *Amazon* and *DBPedia* corpora, respectively.

3.2 Masked Category Prediction

Like how humans perform classification, we want the classification model to focus on category-indicative words in a sequence. A straightforward way is to directly highlight every occurrence of the category vocabulary entry in the corpus. However, this approach is error-prone because: (1) Word meanings are contextualized; not every occurrence of the category keywords indicates the category. For example, as shown in Table 1, the word "sports" in the second sentence does not imply the topic "sports". (2) The coverage of the category vocabulary is limited; some terms under specific contexts have similar meanings with the category keywords but are not included in the category vocabulary.

To address the aforementioned challenge, we introduce a new task, Masked Category Prediction (MCP), as illustrated in Fig. 1, wherein a pre-trained LM creates *contextualized* word-level category supervision for training itself to predict the implied category of a word with the word masked.

To create contextualized word-level category supervision, we reuse the pre-trained MLM method in

Sentence	Language Model Prediction
The oldest annual US team **sports** competition that includes professionals is not in baseball, or football or basketball or hockey. It's in soccer.	sports, baseball, handball, soccer, basketball, football, tennis, sport, championship, hockey, ...
Samsung's new SPH-V5400 mobile phone **sports** a built-in 1-inch, 1.5-gigabyte hard disk that can store about 15 times more data than conventional handsets, Samsung said.	has, with, features, uses, includes, had, is, contains, featured, have, incorporates, requires, offers, ...

Table 1: BERT language model prediction (sorted by probability) for the word to appear at the position of "sports" under different contexts. The two sentences are from *AG News* corpus.

Label Name	Category Vocabulary
politics	politics, political, politicians, government, elections, politician, democracy, democratic, governing, party, leadership, state, election, politically, affairs, issues, governments, voters, debate, cabinet, congress, democrat, president, religion, ...
sports	sports, games, sporting, game, athletics, national, athletic, espn, soccer, basketball, stadium, arts, racing, baseball, tv, hockey, pro, press, team, red, home, bay, kings, city, legends, winning, miracle, olympic, ball, giants, players, champions, boxing, ...
business	business, trade, commercial, enterprise, shop, money, market, commerce, corporate, global, future, sales, general, international, group, retail, management, companies, operations, operation, store, corporation, venture, economic, division, firm, ...
technology	technology, tech, software, technological, device, equipment, hardware, devices, infrastructure, system, knowledge, technique, digital, technical, concept, systems, gear, techniques, functionality, process, material, facility, feature, method, ...

Table 2: The label name used for each class of *AG News* dataset and the learned category vocabulary.

Section 3.1 to understand the contextualized meaning of each word by examining what are valid replacement words. As shown in Table 1, the MLM predicted words are good indicators of the original word's meaning. As before, we regard the top-50 words given by the MLM as valid replacement of the original word, and we consider a word w as "category-indicative" for class c_w if more than 20 out of 50 w's replacing words appear in the category vocabulary of class c_w. By examining every word in the corpus as above, we will obtain a set of category-indicative words and their category labels $\mathcal{S}_{\text{ind}}$ as word-level supervision.

For each category-indicative word w, we mask it out with the `[MASK]` token and train the model to predict w's indicating category c_w via cross-entropy loss with a classifier (a linear layer) on top of w's contextualized embedding $\boldsymbol{h}$:

$$\mathcal{L}_{MCP} = -\sum_{(w,c_w)\in\mathcal{S}_{\text{ind}}} \log p(c_w \mid \boldsymbol{h}_w), \quad (2)$$

$$p(c \mid \boldsymbol{h}) = \text{Softmax}\left(W_c\boldsymbol{h} + \boldsymbol{b}_c\right), \quad (3)$$

where $W_c \in \mathbb{R}^{K\times h}$ and $\boldsymbol{b}_c \in \mathbb{R}^K$ are learnable parameters of the linear layer (K is the number of classes).

We note that it is crucial to mask out the category-indicative word for category prediction, because this forces the model to infer categories based on the word's *contexts* instead of simply memorizing context-free category keywords. In this way, the BERT encoder will learn to encode category-discriminative information within the sequence into the contextualized embedding $\boldsymbol{h}$ that is helpful for predicting the category at its position.

3.3 Self-Training

After training the LM with the MCP task, we propose to self-train the model on the entire unlabeled corpus for two reasons: (1) There are still many unlabeled documents not seen by the model in the MCP task (due to no category keywords detected) that can be used to refine the model for better generalization. (2) The classifier has been trained on top of words to predict their categories with them masked, but have not been applied on the `[CLS]` token where the model is allowed to see the entire

Label Name	Category Vocabulary
good	good, excellent, fair, wonderful, sound, high, okay, positive, sure, solid, quality, smart, normal, special, successful, quick, home, brilliant, beautiful, tough, fun, cool, amazing, done, interesting, superb, made, outstanding, sweet, happy, old, ...
bad	bad, badly, worst, mad, worse, sad, dark, awful, rotten, rough, mean, dumb, negative, nasty, mixed, thing, much, fake, guy, ugly, crazy, german, gross, weird, sorry, like, short, scary, way, sick, white, black, shit, average, dangerous, stuff, ...

Table 3: The label name used for each class of *IMDB* dataset and the learned category vocabulary.

Label Name	Category Vocabulary
good	good, excellent, fine, right, fair, sound, wonderful, high, okay, sure, quality, smart, positive, solid, special, home, quick, safe, beautiful, cool, valuable, normal, amazing, successful, interesting, useful, tough, fun, done, sweet, rich, suitable, ...
bad	bad, terrible, horrible, badly, wrong, sad, worst, worse, mad, dark, awful, mean, rough, rotten, much, mixed, dumb, nasty, sorry, thing, negative, funny, far, go, crazy, weird, lucky, german, shit, guy, ugly, short, weak, sick, gross, dangerous, fake, ...

Table 4: The label name used for each class of *Amazon* dataset and the learned category vocabulary.

sequence to predict its category.

The idea of self-training (ST) is to iteratively use the model's current prediction P to compute a target distribution Q which guides the model for refinement. The general form of ST objective can be expressed with the KL divergence loss:

$$\mathcal{L}_{ST} = \mathrm{KL}(Q\|P) = \sum_{i=1}^{N}\sum_{j=1}^{K} q_{ij} \log \frac{q_{ij}}{p_{ij}}, \quad (4)$$

where N is the number of instances.

There are two major choices of the target distribution Q: Hard labeling and soft labeling. Hard labeling (Lee, 2013) converts high-confidence predictions over a threshold τ to one-hot labels, *i.e.*, $q_{ij} = \mathbb{1}(p_{ij} > \tau)$, where $\mathbb{1}(\cdot)$ is the indicator function. Soft labeling (Xie et al., 2016) derives Q by enhancing high-confidence predictions while demoting low-confidence ones via squaring and normalizing the current predictions:

$$q_{ij} = \frac{p_{ij}^2/f_j}{\sum_{j'}\left(p_{ij'}^2/f_{j'}\right)},\ f_j = \sum_i p_{ij}, \quad (5)$$

where the model prediction is made by applying the classifier trained via MCP (Eq. (3)) to the `[CLS]` token of each document, *i.e.*,

$$p_{ij} = p(c_j \mid \boldsymbol{h}_{d_i:\texttt{[CLS]}}). \quad (6)$$

In practice, we find that the soft labeling strategy consistently gives better and more stable results than hard labeling, probably because hard labeling treats high-confident predictions directly as ground-truth labels and is more prone to error propagation. Another advantage of soft labeling is that the target distribution is computed for every instance and no confidence thresholds need to be preset.

We update the target distribution Q via Eq. (5) every 50 batches and train the model via Eq. (4). The overall algorithm is shown in Algorithm 1.

Algorithm 1: LOTClass Training.

Input: An unlabeled text corpus $\mathcal{D}$; a set of label names $\mathcal{C}$; a pre-trained neural language model M.
Output: A trained model M for classifying the K classes.

```
Category vocabulary ← Section 3.1;
S_ind ← Section 3.2;
Train M with Eq. (2);
B ← Total number of batches;
for i ← 0 to B − 1 do
    if i mod 50 = 0 then
        Q ← Eq. (5);
    Train M on batch i with Eq. (4);
Return M;
```

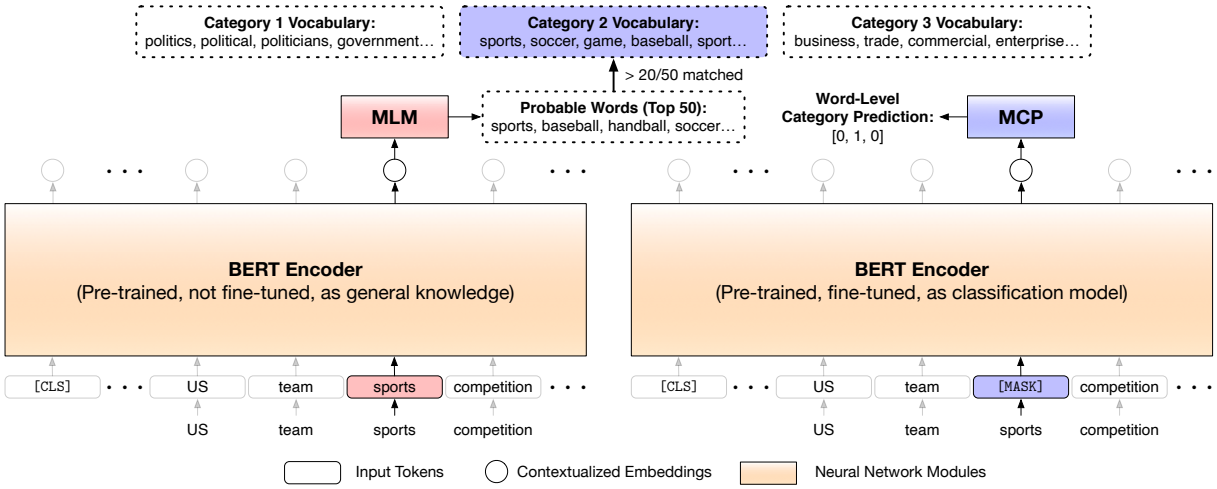


Figure 1: Overview of Masked Category Prediction (MCP). The Masked Language Model (MLM) head first predicts what are probable words to appear at each token's position. A token is considered as "category-indicative" if its probable replacement words highly overlap with the category vocabulary of a certain class. The MCP head is trained to predict the implied categories of the category-indicative words with them masked.

Dataset	Classification Type	# Classes	# Train	# Test
AG News	News Topic	4	120,000	7,600
DBPedia	Wikipedia Topic	14	560,000	70,000
IMDB	Movie Review Sentiment	2	25,000	25,000
Amazon	Product Review Sentiment	2	3,600,000	400,000

Table 5: Dataset statistics. Supervised models are trained on the entire training set. Semi-supervised models use 10 labeled documents per class from the training set and the rest as unlabeled data. Weakly-supervised models are trained by using the entire training set as unlabeled data. All models are evaluated on the test set.

4 Experiments

4.1 Datasets

We use four benchmark datasets for text classification: *AG News* (Zhang et al., 2015), *DBPedia* (Lehmann et al., 2015), *IMDB* (Maas et al., 2011) and *Amazon* (McAuley and Leskovec, 2013). The dataset statistics are shown in Table 5. All datasets are in English language.

4.2 Compared Methods

We compare **LOTClass** with a wide range of weakly-supervised methods and also state-of-the-art semi-supervised and supervised methods. The label names used as supervision on each dataset for the weakly-supervised methods are shown in Tables 2, 3, 4 and 9. (Table 9 can be found in Appendix B.) Fully supervised methods use the entire training set for model training. Semi-supervised method **UDA** uses 10 labeled documents per class from the training set and the rest as unlabeled data. Weakly-supervised methods use the training set as unlabeled data. All methods are evaluated on the test set.

Weakly-supervised methods:

- **Dataless** (Chang et al., 2008): Dataless classification maps label names and each document into the same semantic space of Wikipedia concepts. Classification is performed based on vector similarity between documents and classes using explicit semantic analysis (Gabrilovich and Markovitch, 2007).
- **WeSTClass** (Meng et al., 2018): WeSTClass generates pseudo documents to pre-train a CNN classifier and then bootstraps the model on unlabeled data with self-training.
- **BERT w. simple match**: We treat each document containing the label name as if it is a labeled document of the corresponding class to train the BERT model.
- **LOTClass w/o. self train**: This is an ablation version of our method. We train **LOTClass**

only with the MCP task, without performing self-training on the entire unlabeled data.

Semi-supervised method:

- **UDA** (Xie et al., 2019): Unsupervised data augmentation is the state-of-the-art semi-supervised text classification method. Apart from using a small amount of labeled documents for supervised training, it uses back translation (Sennrich et al., 2016) and TF-IDF word replacing for augmentation and enforces the model to make consistent predictions over the augmentations.

Supervised methods:

- **char-CNN** (Zhang et al., 2015): Character-level CNN was one of the state-of-the-art supervised text classification models before the appearance of neural LMs. It encodes the text sequences into characters and applies 6-layer CNNs for feature learning and classification.

- **BERT** (Devlin et al., 2019): We use the pre-trained BERT-base-uncased model and fine-tune it with the training data for classification.

4.3 Experiment Settings

We use the pre-trained BERT-base-uncased model as the base neural LM. For the four datasets *AG News*, *DBPedia*, *IMDB* and *Amazon*, the maximum sequence lengths are set to be 200, 200, 512 and 200 tokens. The training batch size is 128. We use Adam (Kingma and Ba, 2015) as the optimizer. The peak learning rate is $2e-5$ and $1e-6$ for MCP and self-training, respectively. The model is run on 4 NVIDIA GeForce GTX 1080 Ti GPUs.

4.4 Results

The classification accuracy of all methods on the test set is shown in Table 6. **LOTClass** consistently outperforms all weakly-supervised methods by a large margin. Even without self-training, **LOTClass**'s ablation version performs decently across all datasets, demonstrating the effectiveness of our proposed category understanding method and the MCP task. With the help of self-training, **LOTClass**'s performance becomes comparable to state-of-the-art semi-supervised and supervised models.

How many labeled documents are label names worth? We vary the number of labeled documents per class on *AG News* dataset for training **Supervised BERT** and show its corresponding performance in Fig. 2(a). The performance of **LOTClass** is equivalent to that of **Supervised BERT** with 48 labeled documents per class.

4.5 Study of Category Understanding

We study the characteristics of the method introduced in Section 3.1 from the following two aspects. (1) Sensitivity to different words as label names. We use "commerce" and "economy" to replace "business" as the label name on *AG News* dataset. Table 7 shows the resulting learned category vocabulary. We observe that despite the change in label name, around half of terms in the resulting category vocabulary overlap with the original one (Table 2 "business" category); the other half also indicate very similar meanings. This guarantees the robustness of our method since it is the category vocabulary rather than the original label name that is used in subsequent steps. (2) Advantages over alternative solutions. We take the pre-trained 300-d GloVe (Pennington et al., 2014) embeddings and use the top words ranked by cosine similarity with the label names for category vocabulary construction. On *Amazon* dataset, we use "good" and "bad" as the label names, and the category vocabulary built by **LOTClass** (Table 4) accurately reflects the sentiment polarity, while the results given by GloVe (Table 8) are poor—some words that are close to "good"/"bad" in the GloVe embedding space do not indicate sentiment, or even the reversed sentiment (the closest word to "bad" is "good"). This is because context-free embeddings only learn from local context windows, while neural LMs capture long-range dependency that leads to accurate interpretation of the target word.

4.6 Effect of Self-Training

We study the effect of self-training with two sets of experiments: (1) In Fig. 2(b) we show the test accuracy and self-training loss (Eq. (4)) when training **LOTClass** on the first $1,000$ steps (batches) of unlabeled documents. It can be observed that the loss decreases within a period of 50 steps, which is the update interval for the target distribution Q—when the self training loss approximates zero, the model has fit the previous Q and a new target distribution is computed based on the most recent predictions. With the model refining itself on unlabeled data iteratively, the performance gradually improves. (2) In Fig. 2(c) we show the performance of **LOTClass** vs. **BERT w. simple match** with the same self-

Supervision Type	Methods	AG News	DBPedia	IMDB	Amazon
Weakly-Sup.	**Dataless** (Chang et al., 2008)	0.696	0.634	0.505	0.501
	WeSTClass (Meng et al., 2018)	0.823	0.811	0.774	0.753
	BERT w. simple match	0.752	0.722	0.677	0.654
	LOTClass w/o. self train	0.822	0.860	0.802	0.853
	LOTClass	**0.864**	**0.911**	**0.865**	**0.916**
Semi-Sup.	**UDA** (Xie et al., 2019)	0.869	0.986	0.887	0.960
Supervised	**char-CNN** (Zhang et al., 2015)	0.872	0.983	0.853	0.945
	BERT (Devlin et al., 2019)	0.944	0.993	0.945	0.972

Table 6: Test accuracy of all methods on four datasets.

Label Name	Category Vocabulary
commerce	commerce, trade, consumer, retail, trading, merchants, treasury, currency, sales, commercial, market, merchant, economy, economic, marketing, store, exchange, transactions, marketplace, businesses, investment, markets, trades, enterprise, ...
economy	economy, economic, economies, economics, currency, trade, future, gdp, treasury, sector, production, market, investment, growth, mortgage, commodity, money, markets, commerce, economical, prosperity, account, income, stock, store, ...

Table 7: Different label names used for class "business" of *AG News* dataset and the learned category vocabulary.

Label Name	Category Vocabulary
good	good, better, really, always, you, well, excellent, very, things, think, way, sure, thing, so, n't, we, lot, get, but, going, kind, know, just, pretty, i, 'll, certainly, 're, nothing, what, bad, great, best, something, because, doing, got, enough, even, ...
bad	bad, good, things, worse, thing, because, really, too, nothing, unfortunately, awful, n't, pretty, maybe, so, lot, trouble, something, wrong, got, terrible, just, anything, kind, going, getting, think, get, ?, you, stuff, 've, know, everything, actually, ...

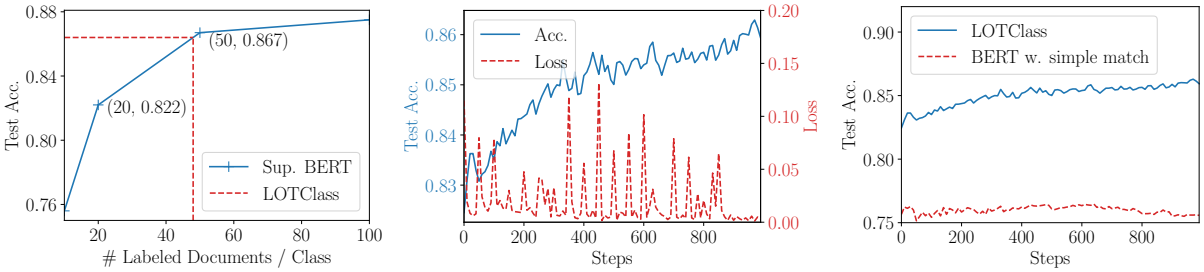
Table 8: GloVe 300-d pre-trained embedding for category understanding on *Amazon* dataset.

training strategy. **BERT w. simple match** does not seem to benefit from self-training as our method does. This is probably because documents containing label names may not be actually about the category (*e.g.*, the second sentence in Table 1); the noise from simply matching the label names causes the model to make high-confidence wrong predictions, from which the model struggles to extract correct classification signals for self-improvement. This demonstrates the necessity of creating word-level supervision by understanding the contextualized word meaning and training the model via MCP to predict the category of words instead of directly assigning the word's implied category to its document.

5 Discussions

The potential of weakly-supervised classification has not been fully explored. For the simplicity and clarity of our method, (1) we only use the BERT-base-uncased model rather than more advanced and recent LMs; (2) we use at most 3 words per class as label names; (3) we refrain from using other dependencies like back translation systems for augmentation. We believe that the performance will become better with the upgrade of the model, the enrichment in inputs and the usage of data augmentation techniques.

Applicability of weak supervision in other NLP tasks. Many other NLP problems can be formulated as classification tasks such as named en-



(a) **Supervised BERT**: Test acc. vs. number of labeled documents.

(b) **LOTClass**: Test accuracy and self-training loss.

(c) **LOTClass** vs. **BERT w. simple match** during self-training.

Figure 2: (On *AG News* dataset.) (a) The performance of **LOTClass** is close to that of **Supervised BERT** with 48 labeled documents per class. (b) The self-training loss of **LOTClass** decreases in a period of 50 steps; the performance of **LOTClass** gradually improves. (c) **BERT w. simple match** does not benefit from self-training.

tity recognition and aspect-based sentiment analysis (Huang et al., 2020). Sometimes a label name could be too generic to interpret (*e.g.*, "person", "time", etc). To apply similar methods as introduced in this paper to these scenarios, one may consider instantiating the label names with more concrete example terms like specific person names.

Limitation of weakly-supervised classification. There are difficult cases where label names are not sufficient to teach the model for correct classification. For example, some review texts implicitly express sentiment polarity that goes beyond word-level understanding: *"I find it sad that just because Edward Norton did not want to be in the film or have anything to do with it, people automatically think the movie sucks without even watching it or giving it a chance."* Therefore, it will be interesting to improve weakly-supervised classification with active learning where the model is allowed to consult the user about difficult cases.

Collaboration with semi-supervised classification. One can easily integrate weakly-supervised methods with semi-supervised methods in different scenarios: (1) When no training documents are available, the high-confidence predictions of weakly-supervised methods can be used as ground-truth labels for initializing semi-supervised methods. (2) When both training documents and label names are available, a joint objective can be designed to train the model with both word-level tasks (*e.g.*, MCP) and document-level tasks (*e.g.*, augmentation, self-training).

6 Conclusions

In this paper, we propose the **LOTClass** model built upon pre-trained neural LMs for text classification with label names as the only supervision in three steps: Category understanding via label name replacement, word-level classification via masked category prediction, and self-training on unlabeled corpus for generalization. The effectiveness of **LOTClass** is validated on four benchmark datasets. We show that label names is an effective supervision type for text classification but has been largely overlooked by the mainstreams of literature. We also point out several directions for future work by generalizing our methods to other tasks or combining with other techniques.

Acknowledgments

Research was sponsored in part by US DARPA KAIROS Program No. FA8750-19-2-1004 and SocialSim Program No. W911NF-17-C-0099, National Science Foundation IIS 19-56151, IIS 17-41317, IIS 17-04532, IIS 16-18481, and IIS-2008334, and DTRA HDTRA11810026. Any opinions, findings, and conclusions or recommendations expressed herein are those of the authors and should not be interpreted as necessarily representing the views, either expressed or implied, of DARPA or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for government purposes notwithstanding any copyright annotation hereon. We thank anonymous reviewers for valuable and insightful feedback.

References

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.".

Ming-Wei Chang, Lev-Arie Ratinov, Dan Roth, and Vivek Srikumar. 2008. Importance of semantic representation: Dataless classification. In *AAAI*.

Jiaao Chen, Zichao Yang, and Diyi Yang. 2020. Mixtext: Linguistically-informed interpolation of hidden space for semi-supervised text classification. In *ACL*.

Xingyuan Chen, Yunqing Xia, Peng Jin, and John A. Carroll. 2015. Dataless text classification with descriptive lda. In *AAAI*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*.

Evgeniy Gabrilovich and Shaul Markovitch. 2007. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *IJCAI*.

Jiaxin Huang, Yu Meng, Fang Guo, Heng Ji, and Jiawei Han. 2020. Weakly-supervised aspect-based sentiment analysis via joint aspect-sentiment topic embedding. In *EMNLP*.

Nitin Jindal and Bing Liu. 2007. Review spam detection. In *WWW*.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *EMNLP*.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR*.

Thomas Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *ICLR*.

Zhen-Zhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. Albert: A lite bert for self-supervised learning of language representations. In *ICLR*.

Dong-Hyun Lee. 2013. Pseudo-label : The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*.

Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N. Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick van Kleef, Sören Auer, and Christian Bizer. 2015. Dbpedia - a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web*, 6:167–195.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2020. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *ACL*.

Chenliang Li, Jian Xing, Aixin Sun, and Zongyang Ma. 2016. Effective document labeling with very few seed words: A topic model approach. In *CIKM*.

Hongmei Liu, Xiaotong Zhang, Lu Fan, Xuandi Fu, Qimai Li, Xiao ming Wu, and Albert Y. S. Lam. 2019a. Reconstructing capsule networks for zero-shot intent classification. In *EMNLP*.

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

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *ACL*.

Julian J. McAuley and Jure Leskovec. 2013. Hidden factors and hidden topics: understanding rating dimensions with review text. In *RecSys '13*.

Dheeraj Mekala and Jingbo Shang. 2020. Contextualized weak supervision for text classification. In *ACL*.

Yu Meng, Jiaxin Huang, Guangyuan Wang, Zihan Wang, Chao Zhang, Yu Zhang, and Jiawei Han. 2020a. Discriminative topic mining via category-name guided text embedding. In *WWW*.

Yu Meng, Jiaming Shen, Chao Zhang, and Jiawei Han. 2018. Weakly-supervised neural text classification. In *CIKM*.

Yu Meng, Jiaming Shen, Chao Zhang, and Jiawei Han. 2019. Weakly-supervised hierarchical text classification. In *AAAI*.

Yu Meng, Yunyi Zhang, Jiaxin Huang, Yu Zhang, Chao Zhang, and Jiawei Han. 2020b. Hierarchical topic mining via joint spherical tree and text embedding. In *KDD*.

Takeru Miyato, Andrew M. Dai, and Ian J. Goodfellow. 2017. Adversarial training methods for semi-supervised text classification. In *ICLR*.

Jinseok Nam, Eneldo Loza Mencía, and Johannes Fürnkranz. 2016. All-in text: Learning document, label, and word representations jointly. In *AAAI*.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up? sentiment classification using machine learning techniques. In *EMNLP*.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *NAACL-HLT*.

Fabio Petroni, Tim Rocktäschel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, Alexander H. Miller, and Sebastian Riedel. 2019. Language models as knowledge bases? In *EMNLP*.

Pushpankar Kumar Pushp and Muktabh Mayank Srivastava. 2017. Train once, test anywhere: Zero-shot learning for text classification. *ArXiv*, abs/1712.05972.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100, 000+ questions for machine comprehension of text. In *EMNLP*.

Anthony Rios and Ramakanth Kavuluru. 2018. Few-shot and zero-shot multi-label learning for structured label spaces. In *EMNLP*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *ACL*.

Yangqiu Song and Dan Roth. 2014. On dataless hierarchical text classification. In *AAAI*.

Duyu Tang, Bing Qin, and Ting Liu. 2015a. Document modeling with gated recurrent neural network for sentiment classification. In *EMNLP*.

Jian Tang, Meng Qu, and Qiaozhu Mei. 2015b. Pte: Predictive text embedding through large-scale heterogeneous text networks. In *KDD*.

Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. 2015c. Line: Large-scale information network embedding. In *WWW*.

Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R. Thomas McCoy, Najoung Kim, Benjamin Van Durme, Samuel R. Bowman, Dipanjan Das, and Ellie Pavlick. 2019. What do you learn from context? probing for sentence structure in contextualized word representations. In *ICLR*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*.

Congying Xia, Chenwei Zhang, Xiaohui Yan, Yi Chang, and Philip S. Yu. 2018. Zero-shot user intent detection via capsule neural networks. In *EMNLP*.

Junyuan Xie, Ross B. Girshick, and Ali Farhadi. 2016. Unsupervised deep embedding for clustering analysis. In *ICML*.

Qizhe Xie, Zihang Dai, Eduard H. Hovy, Minh-Thang Luong, and Quoc V. Le. 2019. Unsupervised data augmentation. *ArXiv*, abs/1904.12848.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *NeurIPS*.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alexander J. Smola, and Eduard H. Hovy. 2016. Hierarchical attention networks for document classification. In *NAACL-HLT*.

Wenpeng Yin, Jamaal Hay, and Dan Roth. 2019. Benchmarking zero-shot text classification: Datasets, evaluation and entailment approach. In *EMNLP*.

Jingqing Zhang, Piyawat Lertvittayakumjorn, and Yike Guo. 2019. Integrating semantic knowledge to tackle zero-shot text classification. In *NAACL-HLT*.

Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *NIPS*.

Yu Zhang, Yu Meng, Jiaxin Huang, Frank F. Xu, Xuan Wang, and Jiawei Han. 2020. Minimally supervised categorization of text with metadata. In *SIGIR*.

A Implementation Details for MCP Training

When training the model with the MCP loss (Eq. (2)), we apply a simple weighting trick that slightly improves the model's performance: Intuitively, the more replacing words (out of top 50 MLM predictions) of a word w overlap with the category vocabulary of c_w, the more likely w is "category-indicative" for c_w. To reflect this, we use the square of overlapping word count $n(w, c_w)$ between the replacing words of w and the category vocabulary of class c_w as the multiplier for the MCP loss of each category-indicative term:

$$\mathcal{L}_{MCP} = - \sum_{(w,c_w)\in\mathcal{S}_{\text{ind}}} n(w, c_w)^2 \log p(c_w \mid \boldsymbol{h}_w).$$

B Label Names Used and Category Vocabulary Obtained for DBPedia

We show the label names used for *DBPedia* corpora in Table 9. In most cases, only one word as the label name will be sufficient; however, sometimes the semantics of the label name might be too general so we instead use 2 or 3 keywords of the class to represent the label name. For example, we use "school" and "university" to represent the class "educational institution"; we use "river", "lake" and "mountain" to represent the class "natural place"; we use "book", "novel" and "publication" to represent the class "paper work".

Label Name	Category Vocabulary
company	companies, co, firm, concern, subsidiary, brand, enterprise, division, partnership, manufacturer, works, inc, cooperative, provider, corp, factory, chain, limited, holding, consortium, industry, manufacturing, entity, operator, product, giant ...
school university	academy, college, schools, ecole, institution, campus, university, secondary, form, students, schooling, standard, class, educate, elementary, hs, level, student, tech, academic, universities, branch, degree, universite, universidad, ...
artist	artists, painter, artistic, musician, singer, arts, poet, designer, sculptor, composer, star, vocalist, illustrator, architect, songwriter, entertainer, cm, painting, cartoonist, creator, talent, style, identity, creative, duo, editor, personality, ...
athlete	athletes, athletics, indoor, olympian, archer, events, sprinter, medalist, olympic, runner, jumper, swimmer, competitor, holder, mile, ultra, able, mark, hurdles, relay, amateur, medallist, footballer, anchor, metres, cyclist, shooter, athletic, ...
politics	politics, political, government, politicians, politician, elections, policy, party, affairs, legislature, politically, democracy, democratic, governing, history, leadership, cabinet, issues, strategy, election, religion, assembly, law, ...
transportation	transportation, transport, transit, rail, travel, traffic, mobility, bus, energy, railroad, communication, route, transfer, passenger, transported, traction, recreation, metro, shipping, railway, security, transports, infrastructure, ...
building	buildings, structure, tower, built, wing, hotel, build, structures, room, courthouse, skyscraper, library, venue, warehouse, block, auditorium, location, plaza, addition, museum, pavilion, landmark, offices, foundation, headquarters, ...
river lake mountain	river, lake, bay, dam, rivers, water, creek, channel, sea, pool, mountain, stream, lakes, flow, reservoir, hill, flowing, mountains, basin, great, glacier, flowed, pond, de, valley, peak, drainage, mount, summit, brook, mare, head, ...
village	village, villages, settlement, town, east, population, rural, municipality, parish, na, temple, commune, pa, ha, north, pre, hamlet, chamber, settlements, camp, administrative, lies, township, neighbourhood, se, os, iran, villagers, nest, ...
animal	animal, animals, ape, horse, dog, cat, livestock, wildlife, nature, lion, human, owl, cattle, cow, wild, indian, environment, pig, elephant, fauna, mammal, beast, creature, australian, ox, land, alligator, eagle, endangered, mammals, ...
plant tree	shrub, plants, native, rose, grass, herb, species, jasmine, race, vine, hybrid, bamboo, hair, planted, fire, growing, flame, lotus, sage, iris, perennial, variety, palm, cactus, trees, robert, weed, nonsense, given, another, stand, holly, poppy, ...
album	lp, albums, cd, ep, effort, recording, disc, compilation, debut, appearance, soundtrack, output, genus, installation, recorded, anthology, earth, issue, imprint, ex, era, opera, estate, single, outing, arc, instrumental, audio, el, song, offering, ...
film	films, comedy, drama, directed, documentary, video, language, pictures, miniseries, negative, movies, musical, screen, trailer, acting, starring, filmmaker, flick, horror, silent, screenplay, box, lead, filmmaking, second, bond, script, ...
book novel publication	novel, books, novels, mystery, memoir, fantasy, fiction, novelist, reader, read, cycle, romance, writing, written, published, novella, play, narrative, trilogy, manga, autobiography, publication, literature, isbn, write, tale, poem, year, text, reading, ...

Table 9: The label name used for each class of *DBPedia* dataset and the learned category vocabulary.