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

Abstract

- propose a weakly-supervised text classification model **LOTClass** - **L**abel-Name-**O**nly **T**ext **C**lassification
- the goal is to use a pre-trained neural language model (LM) both as general knowlege source for **category understanding** and feature representaion learning model for **document classification**
- three steps to build the **LOTClass** models
    - construct a category vocabulary for each class that contains semantically correlated words
    - word-level tasks - collects high-quality category indicative words in the unlabeled corpus to train itself to capture category distinctive information with a contextualized category prediction task
    - document-level tasks - generalize the LM via self-training on abundant unlabeled data
- achievements and contributions
    - 90% accuracy on four benchmark text classification datasets, outperforming existing weakly-supervised methods significantly and yielding even comparable performance to strong semi-supervised and supervised models

2 Related Work

2.1 Supervised Models

- deep learning classifier such as **CNNs** and **RNNs**, use strong representation learning power that effectively captures the high-order, long-range semantic dependency in text sequences
- pre-training language modeling, including autoregressive models such as **ELMO, GPT, XLNet,** and autoencoding models such as **BERT**, able to learn generic linguistic features and serve as knowledge bases, and use transformer architecturs faciliatating strong feature representation learning power

2.2 semi-supervised and Zero-shot Text Classification

- **Augmentation-based methods** generate new instances and regularize the model's predictions to be invariant to small changes in input. including real text sequences via **back translation** or in the hidden states of the model via **perturbation** or **interpolations**
- **Graph-based methods** build text networks with words, documents and labels, and propagate labeling information along the graph via embeddign learning or graph neural networks
- **Zero-shot text classification** generalizes the classifier trained on a known label set to an unknown one without using any new labeled documents, typically relies on **semantic**

attirbutes and descriptions of all classes, correlations among classes, or joint embeddings of classes and documents. 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

- Aims to categoriaze text documents based on word-level descriptions of each category, eschewign the need of any labeled documents. Since the classifier is learned purely from general knowledge without even requiring any unlabeled domain specific data, these methods are called **dataless classification**
- **distant supervision** such as Wikipedia to interpret the label name semantics and derive document concept relevance via explicit semantic analysis
- **topic models** are exploited for seedguided classification to learn seed word-aware topics by biasing the Dirichlet priors and to infer posterior document-topic assignment.
- **Neural approaches** assign documents pseudo labels to train a neural classifier by either generating pseudo documents or using LMs to detect category-indicative words.

3 Methods

3.1 Category Understanding via Label Name Replacement

- use the **pre-trained BERT masked language model (MLM)** to predict what words can replace the label names under most contexts.
    - Input: for each occurrence of a label name in the corpus, feed its contextualized embedding vector $h \in \mathbb{R}^h$ produced by the BERT encoder
    - output: a probability distribution over the entire vocabulary **V** , indicating the likelihood of each word **w** appearing at this position

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

      where

        - **h** is contextualized embedding vector produced by the BERT encoder
        - $\sigma(\cdot)$ is the activation function
        - $W_1 \in \mathbb{R}^{h \times h}$, $h \in \mathbb{R}^h, b \in \mathbb{R}^h$
- $W_2 \in \mathbb{R}^{|V| \times h}$ learnable parameters that have been pre-trained with the MLM objective of BERT
    - use the threshold of 50 words given by the MLM to define valid replacement for each occurrence of the label names in the corpus
- form the category vocabulary of each class using the top100 words ranked by how many times they can replace the label name in the corpus, discarding stopwords and words that appear in multiple categories

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.

Source code for construct construct category vocabulary - https://github.com/yumeng5/LOTClass/blob/aa65ae2d249605adc28ad9650fdb867447b48375/src/trainer.py#L257-L311

3.2 Masked Category Prediction (MCP)

- reuse the pre-trained LM method to create contextualized **word-level category supervision** for training itself to predict the implied category of a word with the word masked.
    - Assumption: 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.
    - Input: obtain a set of category-indicative words, mask each out with the (MASK) token, and the category labels S_{ind}
    - output: for each category-indicative word **w**, use the model to predict w's indicating category c_w

$$\mathcal{L}_{MCP} = -\sum_{(w,c_w)\in S_{ind}} \log p\left(c_w \mid \boldsymbol{h}_w\right) \tag{2}$$

$$p(c \mid \boldsymbol{h}) = \text{Softmax}(W_c \boldsymbol{h} + \boldsymbol{b}_c) \tag{3}$$

where

- $W_c \in \mathbb{R}^{K\times h}$, $b_c \in \mathbb{R}^K$
- K is the number of classes

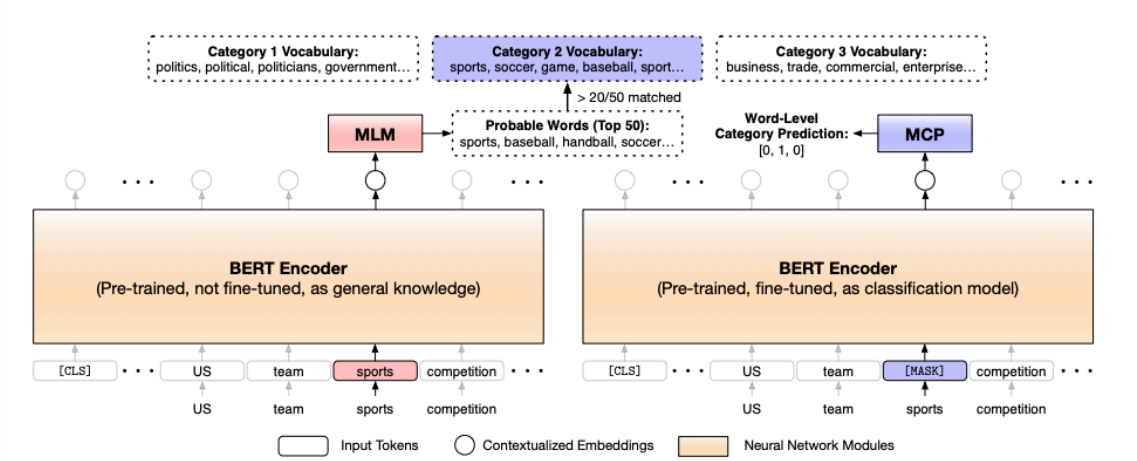


Figure 1: Overview of Masked Category Prediction (MCP). The Masked Language Model (MLM) head first predicts what are probable words to appear at each token's position. A token is considered as "category-indicative" if its probable replacement words highly overlap with the category vocabulary of a certain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

Source code for masked category prediction - https://github.com/yumeng5/LOTClass/blob/aa65ae2d249605adc28ad9650fdb867447b48375/src/trainer.py#L313-L454

3.3 Self-Training

- 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:

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

where N is the number of instances, K is the number of classes.

- two major choices of the target distribution **Q**:
  - hard labeling - converts high-confidence predictions over a threshold τ to one-hot labels, i.e.,

    $q_{ij} = I\left(p_{ij} > \tau\right)$, where $I(\cdot)$ is the indicator function
  - soft labeling - derives **Q** by enhancing high-confidence predictions while demoting low-confidence ones via squaring and normalizing the current predictions

$$ \begin{equation} q_{i j}=\frac{p_{i j}^{2} / f_{j}}{\sum_{j^{\prime}}\left(p_{i j^{\prime}}^{2} / f_{j^{\prime}}\right)}, f_{j}=\sum_{i} p_{i j} \tag{5} \end{equation}

$where the model prediction is made by applying the classifier trained via MCP to the [CLS] token of the entire document as below$

\begin{equation} p_{i j}=p\left(c_{j} \mid \boldsymbol{h}_{d_{i}:(\mathrm{CLS})}\right) \end{equation} \tag{6} $$

- update the target distribution $\mathbf{Q}$ via soft labeling every 50 batches and train the model via $\mathcal{L}_{ST}$.

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 $\leftarrow$ Section 3.1;
$\mathcal{S}_{\text{ind}} \leftarrow$ Section 3.2;
Train M with Eq. (2);
$B \leftarrow$ Total number of batches;
for $i \leftarrow 0$ *to* $B-1$ **do**
  if $i \bmod 50 = 0$ **then**
    $Q \leftarrow$ Eq. (5);
  Train M on batch i with Eq. (4);
Return M;

Source code for self-training - https://github.com/yumeng5/LOTClass/blob/aa65ae2d249605adc28ad9650fdb867447b48375/src/trainer.py#L456-L615

4 Experiments

4.1 Datasets

- AG News (Zhang et al., 2015) - maximum sequence lengths set to be 200
- DBPedia (Lehmann et al., 2015) - maximum sequence lengths set to be 200
- IMDB (Maas et al., 2011) - maximum sequence lengths set to be 512
- Amazon (McAuley and , 2013) - maximum sequence lengths set to be 200

4.2 Compared Methods

- fully supervised methods use the entire training set for model training

    - **char-CNN** (Zhang et al., 2015): encodes the text sequences into characters and applies 6-layer **CNNs** for feature learning and classification.
    - **BERT** (Devlin et al., 2019): use the pretrained BERT-base-uncased model and fine-tune it with the training data for classification.
- semi-supervised method uses 10 labeled documents per class from the training set and the rest as unlabeled data
    - Unsupervised data augmentation - **UDA** (Xie et al., 2019): use back translation and TF-IDF word replacing for **augmentation** to enforces the model to make consistent predictions over the augmentations.
- weakly-supervised methods use the training set as unlabeled data
    - **Dataless** (Chang et al., 2008): 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): generates pseudo documents to pre-train a **CNN classifier** and then **bootstraps** the model on unlabeled data with self-training.
    - **BERT with simple match**: 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**: train LOTClass only with the MCP task, without performing self-training on the entire unlabeled data
- all methods are evaluated on the test set.

4.3 Experiment Settings

- use the pre-trained BERT-base-uncased model as the base neural LM
- the training batch size is 128
- 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.

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

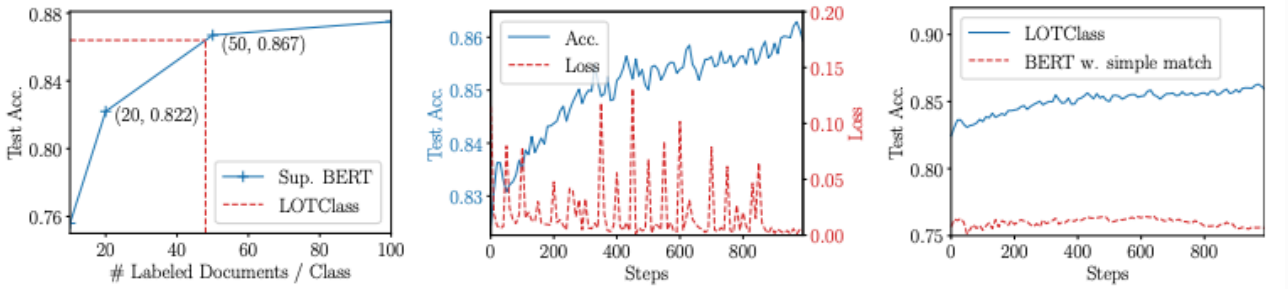
Table 6: Test accuracy of all methods on four datasets.

- 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.
- LOTClass consistently outperforms all weakly-supervised methods by a large margin.
- With the help of self-training, LOTClass's performance becomes comparable to stateof-the-art semi-supervised and supervised models.

4.5 Study of Category Understanding

- We study the characteristics of the method introduced in Section 3.1 from the following two aspects.
    - **sensitivity to different words as label names**. observe that despite the change in label name, around half of terms in the resulting category vocabulary overlap with the original one, the other half also indicate very similar meanings. This guarantees the **robustness** of the method since it is the category vocabulary rather than the original label name that is used in subsequent steps.
    - advantages over pre-trained 300-d GloVe (Pennington et al., 2014) embeddings - 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



(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.

- (b) when training LOTClass on the first 1, 000 steps (batches) of unlabeled documents, 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.
- (c) This is probably because documents containing label names may not be actually about the category; 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 Discussion

- weakly-supervised classification has not been fully explored
    - only use the BERT-base-uncased model rather than more advanced and recent :
    - use at most 3 words per class as label names
    - refrain from using other dependencies like back translation systems for augmentation
- applicability of weak supervision in other NLP tasks
    - could apply similar methods as introduced in this paper in other NLP tasks, eg. name entity, aspect-based sentiment analysis
- limitation of weakly-supervised classification
    - some difficult cases where label names are not sufficient to teach the model for correct classification, which can try active learning
- easily integrate weakly-supervised methods with semi-supervised methods in different scenarios
    - when no training documents are available, the high-confidence predictions of weakly-supervised methods can be used as groundtruth labels for initializing semi-

supervised methods.

- 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)**.

Notes:

Supervised Learning {(x, y)} with label,

Semi-supervised learning {(x, y)}^R with label, {x}^U without label, U>>R

Generative Model - Soft label

Self Learning - Hard label (low density seperation)

Repeat:

Train model from labeled dataset

Applied f to the unlabeled dataset to collect spud-label

Choose the dataset from unlabel dataset, provide weights to each dataset

Remove a set of data from unable dataset, and add them into unable dataset

Distribution entropy - evaluate how concentrate the distriubtion is E(y) = - sum(y * ln(yhat))/len(y)

ML system design frameworks -

The candidate implements a solution to the problem: build an ML system to annotate tweets with topics. Use this to explore how the candidate sets goals and success metrics, how much they understand about systems design, and their level of applied ML knowledge.

Graph-based Approach

The labelled data influence their neighbors, which propagate through the graph.