Incomplete Supervision: Text Classification based on a Subset of Labels

Luning Yang
l4yang@ucsd.edu

Yacun Wang
yaw006@ucsd.edu

Abstract

Many text classification models rely on the assumption that requires users to provide the model with a full set of class labels. This is not realistic, as users may not be aware of all possible classes in advance, or may not be able to obtain an exhaustive list. These models also forced to predict any new document to one of the existing classes, where none of the existing labels is a good fit. Thus, we explore the **Inc**omplete **T**ext **C**lassification (IC-TC) setting: Models mine patterns in a small labeled set which only contains existing labels, apply patterns to predict into existing labels in an unlabeled set, and detect out-of-pattern clusters for potential new label discoveries. We experiment with the potential of weakly supervised ML to detect class labels that humans may not recognize, thus facilitating more accurate classification. From the document and class embeddings and unconfident documents generated, we found that both the baseline and the final model had some capability of detecting unseen classes, and label generation techniques help produce reasonable new class labels.

1 Introduction

In recent years, with the growing complexity and scale of neural network models, they also require more high-quality human-annotated training data to achieve satisfactory performances. These actions usually require extensive domain expertise and are extremely time-consuming. Researchers have strived to develop models in the weak supervision setting that aim to gradually alleviate the human burden in creating such annotations for the documents. In particular, researchers have approached the problem of text classification by developing models that only require the class labels and a little extra information for each class label such as (1) a few representative words (i.e. seed words); (2) authors, publication date, etc. (i.e. metadata). Researchers have shown that models are capable of obtaining reliable results without full human annotation.

However, the problem setting for these models all depend on one key assumption: users need to provide the model with a full set of desired class labels for the model to consider. This is less realistic as users might not know all possible classes in advance; users are also unable to obtain an exhaustive list of class names without carefully reading and analyzing the documents. If some documents happen to fall outside of the given list, the models will be forced to predict one of the existing classes based on normalized probability (e.g. the last softmax layer for a neural network).

For example, an online article database might contain thousands of user-uploaded articles labeled with their domains: news, sports, computer science, etc., and the labels are only limited to existing articles. When trying to classify new documents, there might be some classes existing in our documents whose labels are not provided by our database. For instance, we may have a group of articles in the domain of chemistry, while we don't have the exact label "chemistry" in the database yet.

In this paper, we explore the **Inc**omplete **T**ext **C**lassification (IC-TC) setting: Models mine patterns in a small labeled set which only contains existing labels, apply patterns to predict into existing labels in an unlabeled set, and detect out-of-pattern clusters for potential new label discoveries. We try to explore the possibility of utilizing the power of machines to detect class labels that humans fail to recognize and classify documents to more reasonable labels. In particular, we proposed a baseline model and an advanced model that both leverage semi-supervised and unsupervised learning methods to extract information from the labeled part of the dataset, learn patterns from the unlabeled part, and generate new labels based on documents that have lower similarity between their representation and existing class labels. From the experiments on a well-balanced dataset, both models are performing relatively well in learning high-quality seed words, word embeddings, class and document embeddings, and detecting unseen clusters of classes. With the help of the modern large language model ChatGPT, the models are also capable of finding generic labels for the new classes.

Table 1: Dataset Statistics

Dataset	# Docs	# Classes	Avg # Words
DBPedia	560,000	14	50.01
NYT-Fine	11,527	26	648.24
Reddit	48,407	20	24.11

2 Data

2.1 Datasets

We picked data from 3 categories: news, social media, and Wikipedia. The basic statistics are shown in 1.

- **The New York Times (nyt-fine):** The NYT dataset consists of news articles published by the New York Times, and is reused from the ConWea paper (Mekala and Shang, 2020). There are 26 fine-grained categories which are stemmed from coarse grained labels (omitted), and the number of documents follow a long-tailed distribution.

- **DBPedia:** The articles come from topic classifications based on Wikipedia pages, and is reused from LOTClass (Meng et al., 2020) and X-Class (Wang et al., 2021). There are 14 perfectly balanced classes and a large number of documents.

- **Reddit:** The Reddit dataset contains social media posts from Reddit, which includes the post titles and descriptions. There are 20 classes following a long-tailed distribution.

2.2 Label Removal

We obtain a fully labelled dataset and remove part of the labels to conform with our task setting. We first obtain n documents $\{D_1, D_2, \ldots, D_n\}$ which are each labelled with one of the classes $c_1, \ldots, c_m$, and let f be a mapping from the documents to the labels. We assume the frequency of class labels follow a long-tailed distribution, for example Zipf's Law. Let the frequency of the labels be $f_i = \#\{k : f(D_k) = i\}$, where the labels are ranked by their frequency in descending order.

In the incomplete setting, we also assume: (1) new labels failed to be provided by users all come from less frequent classes; (2) the less frequent, the more likely that it's missed from the users. To create datasets of such a setting, we sample labels from the less frequent half:

From labels $c_{m/2+1}, \ldots, c_m$, we sample l labels from the discrete distribution for each i in the bottom half:

$$P(X = i) = \frac{1/f(i)}{\sum_{j=m/2+1}^{m} 1/f(j)}$$

Finally, from the remaining labels, we remove the same percentage p of labels from documents to gain our final data set.

3 Problem

We attempt to perform text classification under the setting which requires fewer human effort: the input class label set is only a proper subset of all possible labels being predicted. This setting is more realistic because in most use cases users wish to classify their own unannotated corpus with some expectation and some labeled data, and the end task is to complete the labels for the entire corpus.

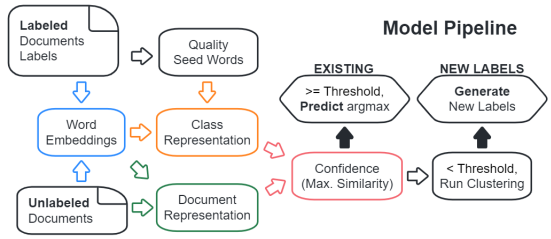
We present the full problem statement. The "incomplete" supervision takes in a set of documents $D = \{D_1, \ldots, D_n\}$, a user-provided set of desired class labels $c^* = \{c_1, \ldots, c_{m^*}\}$, and a set of labels for the first k documents $l_1, \ldots, l_k \in c^*$ The task is to suggest new possible class names $c_{m^*+1}, \ldots, c_m \notin c^*$ to form the full label set $c = \{c_1, \ldots, c_m\}$ and assign a reasonable label $l_j \in c$ for $j \in k+1, \ldots, n$ for the remaining unlabeled documents. In total, we will predict one of the m labels for $n-k$ documents given m^* original labels and k labeled documents.

4 Evaluation

Since the model generates newly suggested labels that are likely to not exist in the original set of full labels, we design the following evaluation process to assess the quality of newly generated labels as well as aggregated classification performance. The term "existing label" refers to the ground truth for documents that exist in the labeled part, and thus will exist in the unlabeled part; in contrary, "newly generated label" refers to ground truths that only exist in the unlabeled part (i.e. is removed during the label removal process).

As described in the models, we allow each document to first select from the existing labels before moving towards generating new labels. Our evaluation process follow a similar multi-step framework to aim on different parts of the model.

Figure 1: Model Pipeline Illustration



4.1 Binary Classification

The model decides on whether to generate new labels for a document based on the confidence of weak supervised document-class representations. The sub-task of predicting whether a document falls outside of existing classes is a binary classification prediction. We evaluate this sub-task using binary precision and recall, with "new labels necessary" as the positive class.

4.2 Existing Label Performance

Based on all documents that have ground truth as existing labels, we evaluate the multi-class classification using the micro- and macro-F1 scores.

4.3 New Label Inspection

After new labels are generated, we inspect the quality of new labels using either manual inspection, and plot word clouds comparing the significant words appeared in the original removed classes and the new clustered classes with generated labels.

5 Method

Figure 1 illustrates the model pipeline for both of the baseline and the final models. The models for the incomplete setting start from a set of labeled documents and another set of unlabeled documents, and mainly contain 4 modules: (1) learning word embeddings from the documents; (2) using word embeddings to find document and class representations; (3) confidence split based on document-class similarity; (4) clustering unconfident documents and generate new labels.

The pipeline utilizes all available documents to find better word representations and makes sure each unlabeled document would have the opportunity to first pick from the existing classes before claiming that it doesn't belong to any of the existing ones. This allows one to substitute high-quality word embedding techniques developed by previous research, and use any supervised learning technique for finding document-class relationships. We choose to use the simple similarity-based method because it helps further reduce human efforts to only provide a subset of class names without any labeled documents, similar to X-Class (Wang et al., 2021), as producing high-quality labeled documents still requires extensive efforts.

In both of the models, we utilized the following shared techniques:

- **Seed Words**: Extract the top 10 unique words per label from TF-IDF scores and create a seed-word set. To ensure the quality and accuracy of the seed-word sets, if two labels share a common seed-word, it is removed and replaced with the label's following most frequent words.

- **Class Representation**: Average embedding of extracted seed words:

$$v_l = \frac{1}{|S_l|} \sum_{s \in S_l} v_s$$

  where S_l is the seed word set for label l.

- **Document Representation**: Average embedding of each word in the document. We choose to use averaged words to align with class representations. In particular, we aggregate the vector representations:

$$v_d = \frac{1}{|W_d|} \sum_{w \in W_d} v_w$$

  where W_d is the words in document d.

- **Similarity**: Using the notations above, the cosine similarity score $s_{d,l}$ between the class l and the document d is computed by

$$s_{d,l} = \frac{\langle v_l, v_d \rangle}{\|v_l\| \cdot \|v_d\|}$$

  then for each document d we obtain the maximum similarity $\hat{s}_d$ and the class associated with the maximum $\hat{l}_d$ over all classes $l \in L$:

$$\hat{s}_d = \max_{l \in L} s_{d,l}, \quad \hat{l}_d = \operatorname*{argmax}_{l \in L} s_{d,l}$$

- **Confident Split**: By inspecting the distribution of maximum similarities, we manually set a threshold τ so that any document with $\hat{s}_d \geq \tau$ is considered close to the existing

class and is predicted to be class $\hat{l}_d$. Any other document with $\hat{s}_d < \tau$ is considered to be relatively less relevant to existing classes, and its document representation v_d is extracted to form the unconfident set.

- **Clustering**: Once obtained the unconfident set, we run a Gaussian Mixture clustering model with default 5 classes. Gaussian Mixture is a probabilistic model-based soft clustering algorithm. For our representations $V = \{v_1, \ldots, v_n\}$ of the n unconfident documents, the objective function to maximize is:

$$P(V|C) = \prod_{i=1}^{n} P(v_i|C) = \prod_{i=1}^{n} \sum_{j=1}^{K} w_i f_j(v_i)$$

where $C = \{C_j\}_{j=1}^{K}$ is the set of clusters, where each cluster has a Gaussian density function f_j and a prior distribution w_j. Classically, we find the maximum likelihood estimation using the Expectation-Maximization (EM) approach.

6 Models

6.1 Method

The final model fills in the remaining slots of the model pipeline by using:

- **Word Embedding**: We obtain the contextualized static representations of each word using pre-trained BERT (Kenton and Toutanova, 2019) embeddings and averaging the representations of all occurrences of the word (Wang et al., 2021). We used the pre-trained `bert-base-uncased` model with its default vector dimension 768.

- **Representations**: Since cosine similarity will perform poorly on a high-dimensional vector, we use PCA to reduce all class and document embeddings.

- **Label Generation**: Instead of directly using statistical methods, we use ChatGPT API – a chatbot fine-tuned using reinforcement learning on OpenAI's state-of-the-art language model GPT-3 (Brown et al., 2020) that has shown the ability for text generation and summarization. We prompt ChatGPT to: (1) Generate topics for the top 25 documents in Gaussian probability per cluster; (2) Use the summarized topics to generate a generic class label.

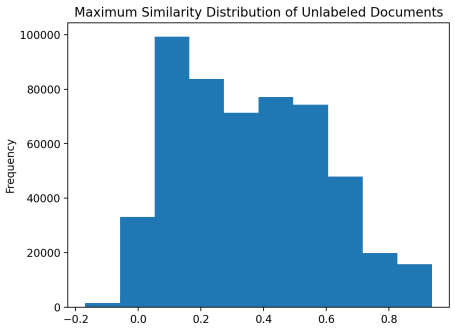


Figure 2: Maximum Similarity Distribution for Unlabeled Documents

6.2 Results and Discussion

We report the results of the final model, by the same order as the pipeline shows.

Seed Words: We present a few example seed words generated from the first supervised TF-IDF module below. The basic TF-IDF scores are able to identify relatively representative seed words.

- Album: studio, band, songs, ...
- Company: founded, headquartered, services, ...
- Film: directed, starring, drama, ...

Similarity Distribution: Figure 2 shows the distribution of the maximum cosine similarity found for all unlabeled documents, and thus provides us with the criteria to get unconfident documents. From the figure, the distribution is roughly normal with a slight right skew, and the value ranges from -0.2 to almost 1.0. This is the ideal distribution, since by the definition of cosine similarity, there will be similarities at 0, indicating the representations are not related; there will also be negative similarities, indicating the representations mean something opposite.

Unconfident Documents: Figure 3 shows the 2D unconfident document representations after applying the t-SNE (van der Maaten and Hinton, 2008) dimensionality reduction technique to visualize the high-dimensional data, color-coded by their original label, with "Other" indicating any existing labels. To generate the 2D representation, we followed the suggestions on `sklearn` t-SNE

Table 2: Final model experiment results: BERT Embeddings

		New Label Binary		Existing Labels	
Threshold	PCA Dimension	Precision	Recall	Micro-F1	Macro-F1
0.05	128	0.924	0.153	0.908	0.905
0.1	128	0.912	0.344	0.905	0.903
0.15	128	0.882	0.539	0.899	0.897
0.05	256	0.924	0.153	0.908	0.905
0.1	256	0.912	0.344	0.905	0.903
0.15	256	0.882	0.539	0.899	0.897
Best Word2Vec Baseline		0.782	0.656	0.856	0.849
ConWea Replication: Best Word2Vec				0.75	0.63

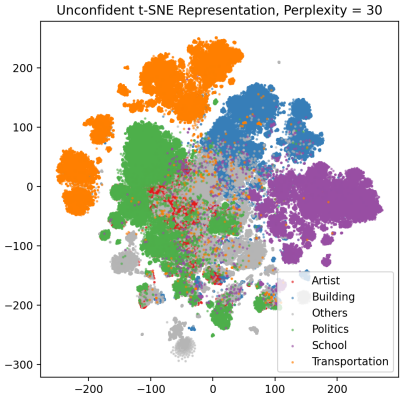


Figure 3: t-SNE Dimensionality Reduction for Unlabeled Documents

to first use Principle Component Analysis (PCA) to reduce to 50 dimensions and apply t-SNE with perplexity 30. From the figure, the unconfident documents follow pretty closely with the original removed labels, and 4 new class distributions are distinctive to be clustered: Transportation, Politics, School, Building. From this distribution, we could expect the Gaussian Mixture method to detect most of the new classes relatively well, with the exception that the "Transportation" class (in orange) has two clear centers, which might not be well detected by the model. There will also be a few noisy documents lying around $(0, 0)$ in t-SNE that confuses clustering and label generation.

Experiment Results: Table 2 shows the results of experimenting with different threshold cutoffs (0.05, 0.1, 0.15) and 2 PCA dimensions (128, 256). From the existing labels prediction, we could see that the supervised + weakly supervised models perform relatively well on classes already known in the dataset, and have improved from the baseline Word2Vec representations. Also, compared to ConWea replication where seed words are all human-chosen, the ability of the model to learn the seed words from the existing labeled documents is helping the understanding of existing classes. Note that sometimes in the ConWea setting, we might not have enough labeled documents to generate the seed words, so human efforts are still useful.

The new label binary classification shows satisfactory results, as in all the experiment settings we observe a precision close to or over 0.9, showing that the similarity cutoff is picking mostly correct out-of-distribution documents. On the other hand, the recall is less optimal, but it increases drastically if we take more documents. This indicates that most out-of-distribution documents lie in the region with positive, non-lowest similarities. It's also possible for documents in the same class to have opposite meanings, which makes the simple split less practical.

However, the existing label performance show stable and robust results, confirming the ability of BERT embeddings to find existing classes. One drawback is that the DBPedia dataset is well-balanced, so micro- and macro-F1 will co-change with a similar trend; when in real-world scenarios when less popular classes are more likely to be unseen, it's unclear how the model will perform.

Finally, we present the labels generated from ChatGPT, compared with the removed labels:

- Generated Labels: 'School', 'Navy', 'Biography', 'Institutions', 'Specialized'
- Removed Labels: 'Artist', 'Building', 'Politics', 'School', 'Transportation'

The word cloud of each removed class and clustered class is plotted in Figure 4. From the label comparison and the word cloud, most important words show in aligned clusters, confirming that the clusters found are relatively close to the original classes. It also lays a good foundation for label

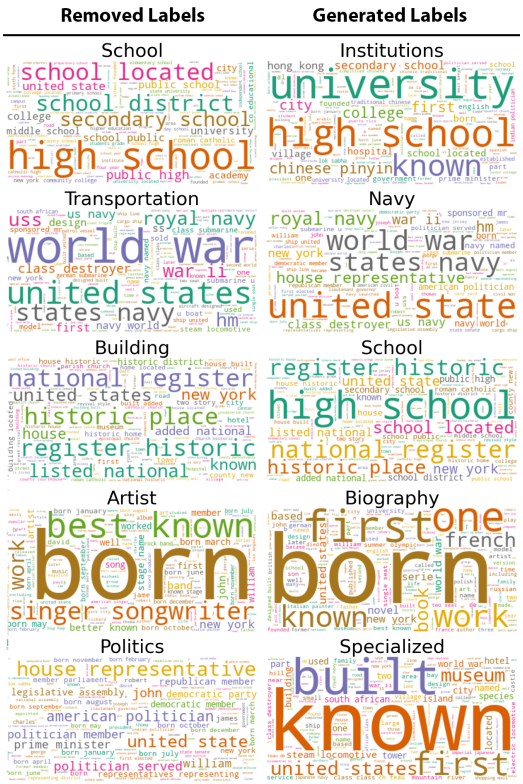


Figure 4: Word cloud for removed (left column) and generated (right column) labels, with manual row alignment. Note they don't necessarily match.

generation. In addition, ChatGPT also produces reasonable generic labels for the clusters.

6.3 Baseline Model

As a comparison, we present the settings for the baseline model, which fills in the remaining slots of the model pipeline by using:

- **Word Embedding**: We train a Word2Vec (Mikolov et al., 2013) model to learn word embedding vectors for each word
- **Label Generation**: We take the clusters obtained in the Gaussian Mixture model, and pick the unique word with the best TF-IDF score for each class for the generated class label.

For the baseline model, we obtain the best results from the following experiment setting:

- **Preprocessing:** Remove punctuation and standard English stopwords;
- **Word2Vec:** Vector Dimension 128, Window 10, Epochs 150;
- **Confidence Split**: Threshold 0.3

The setting gives the following results:

- New Label Binary: Precision 0.782, Recall 0.656.
- Existing Performance: Micro-F1 0.856, Macro-F1 0.849.
- Example Generated Labels: "politician","aircraft","historic","village","navy"

7 Related Work

Research has been conducted to tackle weakly supervised settings to reduce human workload:

Meng et al., 2018; Mekala and Shang, 2020 utilize a few user-provided seed words for the only annotation; Mekala et al., 2020; Zhang et al., 2022 utilize metadata information about the articles such as authors and publication dates which require less human effort to obtain; more advanced methods such as Meng et al., 2020; Wang et al., 2021; Zeng et al., 2022; Shen et al., 2021 only require class label names. All of the models leverage the limited information provided by humans with learned representations including knowledge graphs, large pre-trained language models, etc. to achieve satisfactory classification accuracy. Almost none of the previous research has mentioned this setting, including papers we have covered in Quarter 1, but similar methods could be tweaked in minor ways and applied to this new setting.

There were also attempts to detect out-of-distribution data points and perform open-world classification:

Shu et al., 2017 utilized the 1-vs-Rest layer as the last layer in inference-time model to avoid using softmax to normalize the probabilities and find documents that are out of the existing distribution. Shu et al., 2018 leverages pair-wise classification network in seen and unseen documents for unseen class detection, and utilizes hierarchical clustering to find new clusters of classes. Zhang et al., 2020; Shu et al., 2021 leverages the labeled documents to learn adaptive boundaries for each existing class and used the boundaries to classify unseen documents as part of unseen classes. Most methods above focus on the single *rejection* class, and none help generate new class labels.

Finally, Grootendorst, 2022 uses pre-trained BERT (Kenton and Toutanova, 2019) embeddings on an entirely unsupervised dataset, performs UMAP (McInnes et al., 2018) dimensionality reduction, leverages HDBSCAN (Campello et al.,

2013) clustering technique to find patterns from the documents, and used class-based TF-IDF for label generation. This work is closely related to part of our setting, but focuses on the entirely unsupervised setting, which could find good general topics but might perform less ideally when users have a customized set of classes.

8 Future Work

From the discussions above, although the current final model is capable of finding quality representations, performing reasonable similarity-based confidence splits, and generating labels based on clusters, there are plenty of drawbacks that this model failed to address:

- **Confidence Split**: In some literature (Shu et al., 2017), the split threshold is automatically learned from each existing class, which requires less manual instructions and could lead to potential better splits targeted at individual classes;

- **Clustering**: In the current model, we have to specify the number of clusters beforehand, which requires prior assumptions. We can replace Gaussian Mixture with density-based clustering or LDA to automatically detect the potential number of new classes. Hierarchical clustering can also be applied so that examples of multiple centers can be included as well.

- **Data**: The DBPedia dataset is well-balanced, which is less realistic for unseen classes. In the example of the online database, classes that failed to be provided beforehand are more likely to be unpopular classes. We need to improve method heuristics to work for unbalanced and fine-label datasets.

- **Extension**: Since the model has the ability to detect new labels based on out-of-distribution documents, we can naturally extend the model to detect potential mislabels. For example, we can utilize confidence scores to identify and relabel poor human annotations and allow multi-labels.

- **Extension**: As discussed in the Method section, we choose simple similarity-based techniques to have the opportunity to further decrease human effort, which is both error-prone and time-consuming. We can fully utilize extremely weak supervision techniques to only use class names as supervision and learn class-oriented document representations using attention (Wang et al., 2021).

9 Conclusion

In conclusion, the setting of incomplete text classification can leverage classical ML and weakly supervised methods to predict documents to existing labels; by using a few additional steps on the confident documents, new classes could be produced as well. In particular, the new label binary task, existing label performance, and new label performance all showed relatively, if not better, results. However, there is also much room for improvement to make the model more robust and target more realistic yet less ideal situations.

References

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.

Ricardo J. G. B. Campello, Davoud Moulavi, and Joerg Sander. 2013. Density-based clustering based on hierarchical density estimates. In *Advances in Knowledge Discovery and Data Mining*, pages 160–172, Berlin, Heidelberg. Springer Berlin Heidelberg.

Maarten Grootendorst. 2022. Bertopic: Neural topic modeling with a class-based tf-idf procedure. *arXiv preprint arXiv:2203.05794*.

Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186.

Leland McInnes, John Healy, Nathaniel Saul, and Lukas Großberger. 2018. Umap: Uniform manifold approximation and projection. *Journal of Open Source Software*, 3(29):861.

Dheeraj Mekala and Jingbo Shang. 2020. Contextualized weak supervision for text classification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 323–333, Online. Association for Computational Linguistics.

Dheeraj Mekala, Xinyang Zhang, and Jingbo Shang. 2020. META: Metadata-empowered weak supervision for text classification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8351–8361, Online. Association for Computational Linguistics.

Yu Meng, Jiaming Shen, Chao Zhang, and Jiawei Han. 2018. Weakly-supervised neural text classification. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, CIKM '18, page 983–992, New York, NY, USA. Association for Computing Machinery.

Yu Meng, Yunyi Zhang, Jiaxin Huang, Chenyan Xiong, Heng Ji, Chao Zhang, and Jiawei Han. 2020. Text classification using label names only: A language model self-training approach. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9006–9017, Online. Association for Computational Linguistics.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Jiaming Shen, Wenda Qiu, Yu Meng, Jingbo Shang, Xiang Ren, and Jiawei Han. 2021. TaxoClass: Hierarchical multi-label text classification using only class names. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4239–4249, Online. Association for Computational Linguistics.

Lei Shu, Yassine Benajiba, Saab Mansour, and Yi Zhang. 2021. ODIST: Open world classification via distributionally shifted instances. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3751–3756, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Lei Shu, Hu Xu, and Bing Liu. 2017. DOC: Deep open classification of text documents. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2911–2916, Copenhagen, Denmark. Association for Computational Linguistics.

Lei Shu, Hu Xu, and Bing Liu. 2018. Unseen class discovery in open-world classification.

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

Zihan Wang, Dheeraj Mekala, and Jingbo Shang. 2021. X-class: Text classification with extremely weak supervision. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Ziqian Zeng, Weimin Ni, Tianqing Fang, Xiang Li, Xinran Zhao, and Yangqiu Song. 2022. Weakly supervised text classification using supervision signals from a language model. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 2295–2305, Seattle, United States. Association for Computational Linguistics.

Hanlei Zhang, Hua Xu, and Ting-En Lin. 2020. Deep open intent classification with adaptive decision boundary.

Yu Zhang, Shweta Garg, Yu Meng, Xiusi Chen, and Jiawei Han. 2022. Motifclass: Weakly supervised text classification with higher-order metadata information. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, WSDM '22, page 1357–1367, New York, NY, USA. Association for Computing Machinery.