OW-Class: Open-world Semi-supervised Text Classification

Anonymous ACL submission

Abstract

Open-world semi-supervised classification is a problem where unlabeled samples come from both seen and unseen classes. Existing methods mainly regularize the representation space of all unlabeled samples and solely rely on clustering methods to identify the new classes. We introduce this task in the text domain and argue that class-indicative words may exist in the unlabeled samples, offering a unique opportunity that can help discover the unseen classes. To this end, we propose a novel method OW-Class that jointly performs class name prediction and document clustering, mutually enhancing each other in an iterative manner. Specifically, we first construct an overestimated number of classes through clustering. Then, we extract a list of class-indicative words from the clusters and use them to identify similar clusters and nominate class names. These refined class names further guide us to adjust the document representations, and from here, the iterative loop follows along. We conduct experiments on four popular text classification datasets by setting the most infrequent half of classes as unseen, which emphasizes the imbalanced and emerging nature of real-world scenarios. Results demonstrate the power of OW-Class in both classifying the unlabeled samples and identifying the names of unseen classes.

1 Introduction

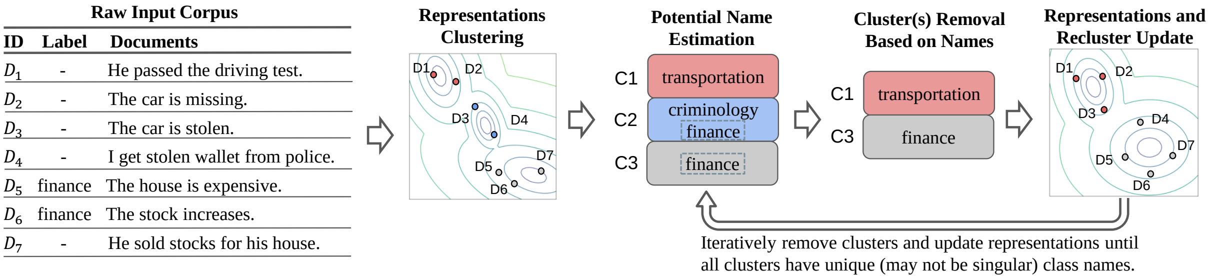
Recent advances in neural networks have achieved state-of-the-art performance in many close-world classification problems where all test samples share the same set of classes as in the training set (LeCun et al., 2015; Silver et al., 2016; Esteva et al., 2017; Devlin et al., 2019). Whereas classical semi-supervised learning settings reduce human efforts by only requiring a subset of examples to be labeled in the dataset (Zhu, 2005; Lee et al., 2013; Kingma et al., 2014; Goldberger and Ben-Reuven, 2017a), making sure these labeled examples have covered all the classes in the dataset is never a trivial effort, especially in the dynamic and emerging real world that is typically open and with limited supervision.

Open-world semi-supervised learning (Cao et al., 2021) is a setting where the labeled training examples only come from a subset of all classes. This subset of classes is called *seen classes* and the rest classes are called *unseen classes*. An open-world semi-supervised method shall learn the semantics of the labeled training samples from the seen classes and generalize the semantics to the unlabeled test set, which contains samples from both seen and unseen classes. Successful solutions to this problem can lift the requirement that the labeled examples have to cover all the classes in the dataset, thus saving tremendous human effort.

In this paper, we study open-world semi-supervised classification in the text domain, which none of the existing text classification methods can handle. Whereas the existing method (Cao et al., 2021) can be extended from images to documents, text classification has its uniqueness because the text is composed of words, some of which reflect the semantics of the classes, giving another kind of supervision signals (Tao et al., 2018; Mekala and Shang, 2020; Wang et al., 2021b). These class-indicative words, upon successfully detected, can help discover the unseen classes. As a concrete example, if the underlying unseen class is *sports*, class-indicative words such as *football* and *Olympics* can help identify sports-related documents.

We thus propose a novel framework OW-Class, which leverages this naturally shared connection among documents, class-indicative words, and class names. It brings up the potential to extend the representation learning and clustering by iteratively refining the clusters of documents and the names of the classes, through class-indicative words.

Figure 1 illustrates the general idea of OW-Class. Specifically, we first make an overestimation of the



Raw Input Corpus
Representations Clustering
D1
D2
D3
D4
D5
D6
D7
Potential Name Estimation
C1
transportation
C2
criminology
finance
C3
finance
Cluster(s) Removal Based on Names
C1
transportation
C3
finance
Representations and Recluster Update
D1
D2
D3
D4
D5
D6
D7
Iteratively remove clusters and update representations until all clusters have unique (may not be singular) class names.

ID	Label	Documents
D_1	-	He passed the driving test.
D_2	-	The car is missing.
D_3	-	The car is stolen.
D_4	-	I get stolen wallet from police.
D_5	finance	The house is expensive.
D_6	finance	The stock increases.
D_7	-	He sold stocks for his house.

Figure 1: An overview of OW-Class framework. Given the corpus and a part of labels, we first estimate document representations and construct the initial clusters. And then, we perform an iterative cluster refinement to remove redundant clusters. At the end of each iteration, we will update the document representations and recluster them.

number of classes and construct initial clusters of documents. Then, we employ an iterative process to refine the clusters and their names. To label clusters with names, we learn a classifier that can identify class-indicative words. The classifier is trained by the given seen-class supervision. When there is redundancy among these clusters, the same class-indicative words for the clusters will overlap, in which case we know one cluster is redundant. To re-estimate clusters with class names, we estimate class representations with the help of class-indicative words and tailor a document representation learning guided by class names to re-construct the clusters. We repeat this iterative process till the number of classes no longer decreases.

Extensive experiments on four popular datasets have shown the strong performance of OW-Class. For example, on the NYT dataset, OW-Class can outperform the best-compared methods by 15.39%. It is worth mentioning that we specifically design challenging experiments by setting the most infrequent half of classes as unseen, which emphasizes the unbalanced and emerging nature of real-world scenarios. Further tests show that OW-Class is robust to (extremely) imbalanced data distributions. Moreover, the class names our method detects are highly related to and sometimes even the same as the ground truth class names.

To the best of our knowledge, this is the first work for open-world semi-supervised text classification. Our contributions are as follows.

- We identify the unique opportunity of leveraging class names and class-indicative words for unseen class discovery.
- We propose a novel method OW-Class that jointly performs class name prediction and document clustering in a mutually improving manner.
- Extensive experiments demonstrate that OW-Class outperforms the previous benchmark in various manners. Ablation studies also verify the necessity of the components in OW-Class.

Reproducibility. We will release the code and datasets on Github[1].

2 Preliminaries

In this section, we formally define the problem of open-world semi-supervised text classification. And then, we brief on some preliminaries about CGExpan and X-Class, two crucial building blocks that we will use in our method.

Problem Formulation. In an open-world setting, there exists a not fully known set of classes $\mathcal{C}$, which follow the same hyper-concept and have the same granularity and a set of documents $\mathcal{D}$. Each document can be uniquely assigned to a class. An open-world semi-supervised model can observe partial information of $\mathcal{C}$. In this work, we assume that partial information is given as a labeled dataset $\mathcal{D}_s = \{x_i, y_i\}_{i=1}^n, y_i \in \mathcal{C}_s$, where $\mathcal{C}_s \subset \mathcal{C}$. The goal of the model is to classify the remainder of the dataset, $\mathcal{D}_u = \mathcal{D} \backslash \mathcal{D}_s$, where some of the labels in $\mathcal{C}_u = \mathcal{C} \backslash \mathcal{C}_s$ is completely unknown to the model. Therefore, the model needs to discover the number of them, the names of them, and finally the attribution of documents to them.

CGExpan. Entity set expansion aims to expand a set of seed keywords (e.g., *United Sates*, *China*) to new keywords (e.g., *Japan*) following the same hyper-concept (i.e., *Country*). Leveraging this technique, we can expand the seen class names to more potential class names, helping to capture the semantics of unseen classes. However, traditional methods typically give duplicated and semantically-shifted entities even at the top of the rank list. In our method, we employ CGExpan (Zhang et al., 2020), one of the current state-of-the-art methods for set expansion. CGExpan selects automatically generated hyper-concept words by probing a pre-trained language model (e.g., BERT), and further ranks all possible words guided by selected hyper-

[1] https://github.com/anonymous

concept. In our work, we utilize CGExpan to find semantically related words to the user-given class names as candidates for the class-indicative words. This is because CGExpan, or any set expansion methods, tends to propose a superset of the true underlying class names, including many too fine-grained names for classes. Our method resolves this problem by training a classifier that can identify class-indicative words, which is an important reason why we need some labeled user supervision.

X-Class. X-Class is an extremely weakly supervised text classification method that works without any supervision of labeled documents and only relies on the class names (Wang et al., 2021b). It proposes a framework that first learns class representations from class names and then estimates class-oriented document representations. It also integrates clustering to refine the class boundaries for each class. While X-Class showed promising performance in close-world classification settings with minimal supervision, it cannot work in open-world settings. We use X-Class to identify clusters from class names, however, in our use case, the class names might be noisy and contain more fine-grained or similar names. We note that the original version of X-Class fails to solve this, because of the unstable class representation in its very first step. We propose a stable class representation estimation that can work in such a noisy case, by leveraging (a portion of) the class-indicative words again.

3 Our OW-Class Method

In this section, we first present the overall design of OW-Class (Figure 1) and then discuss its two key components: (1) *Clusters* → *Class names* and (2) *Class names* → *Clusters*.

3.1 Overall Design

Similar to previous work on open-world classification (Cao et al., 2021), our method OW-Class also first gives an initial overestimation of classes, thereby transforming the problem into reducing extra classes and assigning documents to the remaining. However, different from works in the image domain, we rely on the names of the classes to cohesively group similar documents. And as we will demonstrate later, these class names along with class-indicative words will aid in refining the clusters, allowing our method to propose a very accurate number of classes in the end.

Following above, OW-Class breaks the open-world class identification and document classification into two sub-problems: (1) the removal of similar clusters and identification of a set of similar-granularity class names when given (possibly too) fine-grained clusters of documents, and (2) the clustering of documents when given a list of (partially correct) class names.

The first problem is challenging, because the initial clusters are noisy and possibly too fine-grained compared with the seen class names (leading to duplicates). We heavily leverage the concept of class-indicative words, words that are semantically related to the clusters, to identify and eliminate clusters that are too similar. The class-indicative words are obtained by ranking high-potential words according to their similarity to the corresponding clusters, where the similarity is estimated through a trained compact network with user-provided supervision. The class-indicative words are also used to suggest the class name for the remaining clusters.

The second problem is also not easy as we need to cluster documents given partially correct class names. While there are existing extremely weak supervision works that can classify documents given class names (Aharoni and Goldberg, 2020; Meng et al., 2020; Wang et al., 2021b), they focus on a perfectly given list of class names, and as we show in our ablations, do not perform well when the class names have redundancy. We show here the power of (a part of) class-indicative words that are used to solve the first problem. They can be integrated with an existing extremely weak supervision method X-Class (Wang et al., 2021b) to stabilize the clustering[2].

We propose to integrate the solutions to these two sub-problems together, so we can refine the class names and clusters interleavingly, enjoying the mutual enhancement loop. This loop naturally stops when we don't see any redundant clusters.

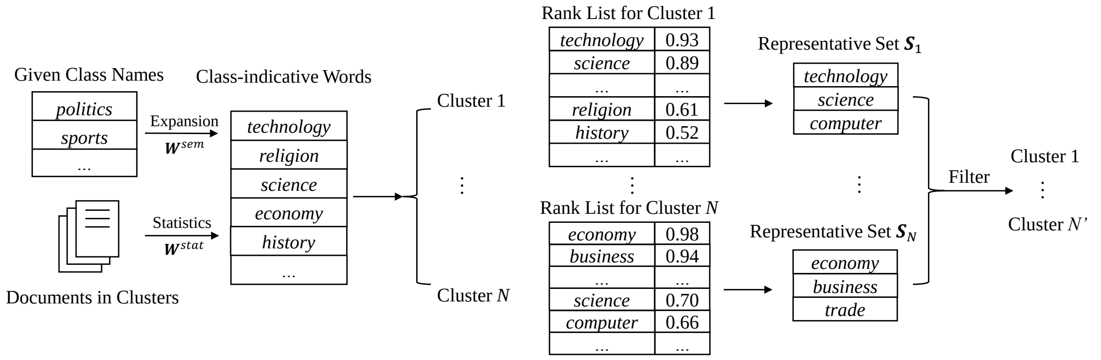
The pseudo-code of the algorithm is summarized in Algorithm 1. And more subtle implementation details can also be found in Appendix B.

3.2 Clusters → Class Names

Figure 2 shows an overview of this subsection.

Proposing High-potential Words. The first step to constructing the class-indicative words is to limit the possible such words to consider. We consider words of two types: (1) words in the same semantic

[2] We do not differentiate the naming of clustering and classification too much here, since it is a noisy setting.



(1) Proposing High-potential Words (2) Indicativeness Ranking (3) Removal of Redundant Clusters

Figure 2: An overview of *Cluster* → *Class Names*.

class as the given class names, and (2) words that are statistically representative in the clusters. For the semantically related words $\mathbf{W}^{sem}$, we employ CGExpan (Zhang et al., 2020), a set expansion method designed to propose semantically and granularly similar words. For the statistically outstanding words $\mathbf{W}^{stat} := \mathbf{W}_i^{stat}, 1 \leq i \leq C$, where C is the size of the clusters in the current iteration, we follow Mekala and Shang (2020) to find these words within each cluster (details in Appendix B).

Indicativeness Ranking. To quantitatively evaluate how similar any two classes are, we need to rank the high-potential words for each cluster and compare the similarity of their most representative words. Here, we utilize the user-given class names and labels as supervision signals to train a classifier to determine the relevance of a high-potential word to a cluster.

Specifically, we construct features for a high-potential word to a cluster based on the representation similarity [3] of the word to the cluster's statistically outstanding words $\mathbf{W}_i^{stat}$. The labeled documents are virtually clustered and used as positive signals since we know the user-given class names. The negative signals are deemed through a heuristic based on the most dissimilar high-potential word to the current cluster name. We train a Multilayer Perceptron (MLP) binary classifier on the features and signals, and assign a classification probability score to each high-potential word and cluster pair $p(w, i), w \in \mathbf{W}^{sem} \cup \mathbf{W}^{stat}, 1 \leq i \leq C$.

We also propose a post-processing step to remove generic words from the ranking. We follow previous work (Jones, 1972) and design a penalty coefficient $\mu(w, i)$ based on inter-class statistics (details in Appendix B). The final indicativeness ranking is based on the product of two scores:

$$I(w, i) = p(w, i) \times \mu(w, i).$$

Removal of Redundant Clusters. We finally discuss how we remove the clusters that have too similar meanings on the granularity of given class names. In simple terms, we pick the top class-indicative words as the representative set of words $\mathbf{S}_i$ for a cluster i, and remove clusters that have non-empty intersections in the sets. By removal, we do not mean to remove the data points, rather that we do not consider the cluster when determining the final list of class names. We need to address two details, first the size of the set and second the cluster to be removed when two have intersections.

The size of this set for a cluster i is related to the quality of the class-indicative words, which we estimate by two factors: (1) T, the number of iterations passed which reflects the overall improvement of classification; (2) Q, the ratio of the indicativeness score $I(\cdot, i)$ between the highest and lowest in the set, an intra-cluster restriction to prevent the selection of low-quality class-indicative words. For each cluster, we add the class-indicative words to the representative set $\mathbf{S}_i$ one by one, until either $|\mathbf{S}_i| = T$ or $Q < \beta$. β is a hyper-parameter that controls the looseness of class-indicative words in the set.

When two sets $\mathbf{S}_i$ and $\mathbf{S}_j$ have overlapped, we would like to retain the cluster that contains more coherent documents since we believe it means the cluster is more robust and therefore the class-indicative words are of higher quality. We introduce the representation similarity η_i to denote how coherent a cluster i is,

$$\eta_i = \frac{1}{|\mathbf{R}_i|} \sum_{\mathbf{r} \in \mathbf{R}_i} \cos\left(\mathbf{r}, \overline{\mathbf{R}}_i\right), \qquad (1)$$

where $\mathbf{R}_i$ is the list of all documents' representations in cluster i[4], and $\overline{\mathbf{R}}_i$ is the average representa-

[3] See Appendix B for the exact definition of similarity.

[4] In the iterative framework, we know the document representation from the sub-problem of obtaining clusters.

tion of the list. When overlap happens, we remove the cluster that has lower coherence η.

Finally, after the removal of all redundant clusters, we re-estimate the indicativeness ranking for the remaining clusters to eliminate the effect of redundant clusters and preserve the word with the top score as the final class name for each cluster.

3.3 Class Names → Clusters

While this sub-problem resembles extremely weak supervision for text classification[5], we note that the noise brought by imperfect class names can be detrimental to traditional methods (Wang et al., 2021b; Meng et al., 2020). We demonstrate that it is possible to adapt a traditional method X-Class (Wang et al., 2021b) with the statistically outstanding words $\mathbf{W}_i^{stat}$ to a noisy scenario.

Importantly, the divergence of X-Class stems from its very first step, which involves estimating a class representation based on each class name. The estimation relies on the assumption of mutual similarity and granularity, which is not the case in our scenario. We will use the class-indicative words to stably estimate class representations, and then apply them for X-Class. It is possible that our stable class representation estimation can help other text classification methods, and we leave that to future research.

Class Representation Estimation Based on Class Name. We slightly diverge to discuss how X-Class estimates the class representation as we are going to use part of it as a subroutine. Specifically, given a class name w, they find its synonyms list $\mathcal{K}_w$ by the similarity of representations. Then they estimate the class representations $\mathbf{x}_w$ by computing an average of representations of words in $\mathcal{K}_w$, weighted by the inverse rank of similarities.

$$\mathbf{x}_w = \frac{\sum_{i=1}^{\mathcal{K}_w} \frac{1}{i} \cdot \mathbf{s}_{\mathcal{K}_{w,i}}}{\sum_{i=1}^{\mathcal{K}_w} \frac{1}{i}}.$$

Back to our method, we consider the list of class-indicative words for a cluster i. We first initialize a representation $\mathbf{r}_w$ of all words w in $\mathbf{W}_i^{stat}$ through the class representation estimation method in X-Class. The byproduct of that process is a list of synonyms $\mathcal{K}_w$ for each word $w \in \mathbf{W}_i^{stat}$. We use this list to find the relatedness of a word w in $\mathbf{W}_i^{stat}$, as defined by

$$h_w = |\mathcal{K}_w \cap \mathbf{W}_i^{stat}|.$$

Then, we perform a weighted average of $\mathbf{r}_w$ based on the relatedness to obtain the class representation:

$$\mathbf{y}_i = \frac{\sum_w h_w \cdot \mathbf{r}_w}{\sum_w h_w}.$$

We use this stable class representation as input to X-Class and obtain the clusters. The byproduct of X-Class is document representations that we will use in the next iteration of class name nominations.

4 Experiments

4.1 Datasets

We evaluate OW-Class on four popular datasets of different textual sources, including three news article datasets 20News (Lang, 1995), NYT (Meng et al., 2018) and AGNews (Zhang et al., 2015), and a large ontology categorization dataset DBpedia (Zhang et al., 2015) based on 14 ontology classes in DBpedia. Table 1 contains the detailed statistics of the four datasets.

Table 1: An overview of our datasets. The imbalance factor refers to the ratio of sample sizes between the most frequent class and least frequent one in the dataset.

	20News	**NYT**	**AGNews**	**DBpedia**
# of Classes	5	5	4	14
# of Documents	17,871	13,081	120,000	560,000
Imbalance	2.02	16.65	1.0	1.0

Sentiment analysis is also popular in text classification. However, many explored sentiment analysis settings with weak supervision are on the coarse-grained setting (Wang et al., 2021b; Meng et al., 2020) with 2 classes (positive and negative), which is not practical for open-world class detection.

4.2 Compared Methods

We compare our method with ORCA. **ORCA** (Cao et al., 2021), originally proposed for the image domain, is a general method for open-world semi-supervised classification. It utilizes an uncertainty adaptive margin to reduce the learning gap between seen and unseen classes. To transfer ORCA to the text domain, we concatenate the original classifier with BERT.

We also propose two strong baselines. BERT is known to capture the domain information of a document well (Aharoni and Goldberg, 2020; Wang et al., 2021a). So we design **BERT+GMM**, which utilizes the CLS token representations after fine-tuning on the partially given dataset to fit a GMM for all classes. **CGExpan+X-Class** takes the high-quality class names from CGExpan and employs

[5] Also known as text classification with class names only.

Table 2: Evaluations of compared methods and OW-Class. The mean macro-F_1 scores over three runs are reported.

Method	Additional Input	20News All	20News Seen	20News Unseen	NYT All	NYT Seen	NYT Unseen	AGNews All	AGNews Seen	AGNews Unseen	DBpedia All	DBpedia Seen	DBpedia Unseen
ORCA	None	19.63	49.07	0	28.23	61.32	18.52	23.13	46.27	0	**98.93**	**99.52**	**98.33**
OW-Class		**79.87**	**90.24**	**72.96**	**91.13**	**90.34**	**91.59**	87.07	84.11	90.04	93.88	96.30	91.46
ORCA (Our estimation)	Number of classes	76.49	87.50	69.16	44.41	67.70	28.89	84.20	88.49	81.01	80.08	99.23	60.94
BERT+GMM (Our estimation)		57.14	71.63	47.49	58.83	76.24	47.22	63.54	67.91	59.16	82.05	96.18	67.93
CGExpan+X-Class (Our estimation)		74.43	80.07	70.66	71.89	89.10	60.42	78.33	81.34	75.35	83.93	93.22	74.63
ORCA (Oracle)		63.45	88.69	46.64	27.82	39.22	20.21	**90.96**	**88.07**	**93.85**	98.93	99.52	98.33
BERT+GMM (Oracle)		39.08	57.95	26.66	46.09	71.01	29.47	47.68	61.88	33.48	75.39	97.11	53.67
CGExpan+X-Class (Oracle)		67.58	78.12	60.56	75.74	87.20	68.09	83.00	79.72	86.27	67.46	88.77	46.15

X-Class on top of the class names (details in Appendix B).

4.3 Experimental Settings

For the basic experiments, we split the classes into half seen and half unseen. Among the seen classes, half of the documents contain labels and the rest are unlabeled (Figure 3).

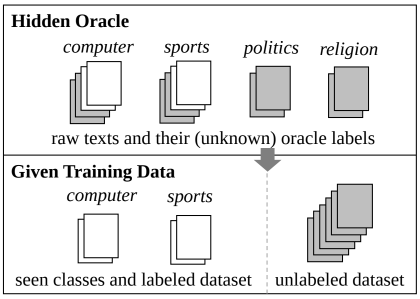


Figure 3: Schematic diagram of the corpus split. Only a certain proportion of samples in popular classes are provided as training labels.

Since all compared methods (except our OW-Class) require knowing the total number of classes, we test them in three ways.

- **Oracle**: Ground truth number of classes is given.
- **Our Estimation**: We give OW-Class's final prediction of classes to the baselines.
- **Estimation in ORCA**: ORCA also introduces a method (Han et al., 2019) to estimate the number of classes, so we test ORCA under this estimation. Since further experiments show this method doesn't work in most our datasets, we do not test other baselines with its estimation.

Evaluation. Since the final number of classes produced by a method may not be equal to the ground truth, a mapping from the prediction to the actual classes is required. We first do a maximum matching between the predicted classes and the ground truth classes to ensure that each ground truth class corresponds to at least one class in our results. In the case when the number of predicted classes is less than the ground truth, we create virtual classes with no documents inside. Then, each remaining predicted class is assigned to the ground truth class that shares the largest overlap. After applying the mapping, we use the F_1 scores as the metric on classification performance for both seen and unseen classes (we show macro-F_1 scores in this section and micro-F_1 scores can be found in Appendix D).

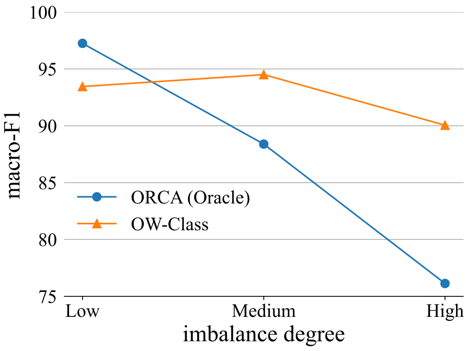


Figure 4: Overall performance of ORCA and OW-Class on different imbalance degrees.

4.4 Experimental Results

OW-Class Performance. We assess the open-world semi-supervised performance of OW-Class versus other baselines. Table 2 contains detailed comparisons. Specifically, OW-Class outperforms the two baselines BERT+GMM and CGExpan+X-Class across all four datasets for both seen and unseen classes, even though they are given the oracle number of classes as input. This strengthens the need for our iterative refinement process since merely applying fine-tuning or X-Class does not bring as good performance as ours. Moreover, while the general method ORCA performs well on DBpedia, its performance is exceptionally poor on the other three datasets as it fails to detect the number of classes anywhere close to the correct number. Even when the correct number is given as input to ORCA, OW-Class still outperforms it significantly in 20News and NYT, with only a small performance margin on AGNews and DBpedia.

Imbalance Tolerance. As a generic solution, ORCA with the ground truth number of classes outperforms OW-Class on the two balanced datasets, AGNews and DBpedia, while underperforming on

Table 3: Ablation study of OW-Class. The mean macro-F_1 scores over three runs are reported.

Method	20News			NYT			AGNews			DBpedia		
	All	Seen	Unseen	All	Seen	Unseen	All	Seen	Unseen	All	Seen	Unseen
OW-Class	79.87	90.24	72.96	91.13	90.34	91.59	87.07	84.11	90.04	93.88	96.30	91.46
OW-Class-noIter	64.90	84.29	51.98	67.84	85.48	56.09	75.20	77.89	72.51	92.59	96.88	88.29
OW-Class-oneIter	79.68	90.37	72.55	89.84	89.47	90.08	84.95	82.25	87.64	95.39	97.50	93.28
OW-Class-X	75.18	84.03	69.22	85.29	88.22	83.33	88.48	84.82	92.14	90.72	95.04	86.40

the two other imbalanced datasets. To gain further insights, we conduct experiments on the tolerance of imbalance for OW-Class and ORCA. As shown in Table 4, we construct three imbalanced DBpedia datasets with different degrees of imbalance. This is achieved through removing the number of samples in each class by a linearly increasing ratio Δ. For example, when $\Delta = 5\%$, the classes have 100%, 95%, $90\%, \ldots$ of its original documents. We choose the ordering of classes randomly but fixed across the Low, Medium and High experiments, and by design, the classes with a larger number of documents are seen classes.

Table 4: An overview of the imbalanced DBpedia datasets with 14 classes.

	Low	Medium	High
Δ	3%	4%	5%
# of Documents	450,800	41,44,00	378,000
Imbalance	1.64	2.08	2.86

Figure 4 shows the result of OW-Class and ORCA on the three datasets. ORCA is sensitive to imbalanced classes. Its overall performance drops more than 20% as the data distribution get more imbalanced, while OW-Class is rather stable. This experiment shows that OW-Class is more robust to imbalanced classes of text datasets which are common in the real world (e.g., the imbalance ratio of NYT collected from NYT news is 16.65).

Effect of Multiple Iterations. We further explore how the iterations of nomination and classification in Sec. 3.2 and 3.3 influence the performance. OW-Class's performance without iterative refinement is labeled as OW-Class-noIter. As shown in Table 3, it performs better than naive baseline BERT+GMM by more than 7.76%. Though the initial clusters are good, the unseen part is still up to 35.50% lower than OW-Class on NYT, indicating the iteration improves the performance for unseen classes. After one iteration, the performance of OW-Class-oneIter almost converges among all the datasets to OW-Class's best scores and even beats OW-Class on DBpedia. We believe the first iteration is the most critical one in our method. As shown in Table 5, the first iteration can remove the majority of redundant classes. The subsequent iterations further detect remaining redundancy but lead to little performance improvement.

Table 5: Predictions of the number of classes. The average numbers over three runs are reported.

Method	20News	NYT	AGNews	DBpedia
Ground Truth	5	5	4	14
ORCA's baseline	2.0	3.0	2.0	14.0
OW-Class-noIter (initial guess)	10	15	10	35
OW-Class-oneIter	7.7	10.3	7.0	28.0
OW-Class	7.3	9.3	6.0	20.3

Table 6: Examples of the class nomination.

Dataset	Ground Truth	OW-Class
20News	`science`	`encryption, electronics, orbit`
NYT	`business`	`economy`
AGNews	`sports`	`game`
DBpedia	`artist`	`painter, singer`

Effect of Improved X-Class. In Sec. 3.3, we introduce class-indicative words into X-Class to assist in stable class representation computation. We verify how the improved X-Class contributes to the overall performance. OW-Class-X directly utilizes the class names as the input of the original X-Class for clustering. As shown in Table 3, except for AGNews, the overall performance of OW-Class-X drops over 3.16% on the other three datasets. For unseen classes, this drop can reach 8.26% in NYT. One possible reason for the drop is the classification result of the original X-Class highly dependent on the class names. The results get poor if the chosen names are not accurate enough.

Prediction of the Number of Classes. OW-Class starts with an initial guess on the number of classes (details in Appendix C) and removes redundant ones iteratively. The number of the remaining classes is its prediction of the total number of classes. As shown in Table 5, after the first iteration, the number of redundant classes drops from 33% to 60%. OW-Class's final predicted number of classes is around 1.5 times larger than the ground truth, but the estimation turns out to be reasonable as shown in Table 6 (details in Appendix D). In fact, OW-Class overestimates because its predicted classes are the fine-grained version of the ground truth classes. For example, DBpedia's *artist* class can be split into *painter* and *singer*. As a baseline, ORCA can estimate the number of classes. Though its estimation is accurate on 14-classes DBpedia,

Table 7: Study of Hyper-parameters Sensitivity.

Parameter	Tested value	Relative Performance Change	
		NYT	DBpedia
$\lvert\mathbf{W}^{sem}\rvert$	20, 25, 30, 35, 40	-4.93% $\sim$ +1.46%	0%
$\lvert\mathbf{W}_i^{stat}\rvert$	40, 50, 60, 70	-3.65% $\sim$ 0%	-3.26% $\sim$ +0.36%
β	0.55, 0.6, 0.65, 0.7	0%	-0.72% $\sim$ +0.13%

the predictions on the other datasets are almost the same as the number of seen classes. This shows it requires enough seen classes to ensure its accuracy.

Hyper-parameter Sensitivity. OW-Class has three hyper-parameters: $|\mathbf{W}^{sem}|$, $|\mathbf{W}_i^{stat}|$, and β, and we show their default values in Appendix B. To further explore the stability and robustness of OW-Class, we conduct a hyper-parameter sensitivity study on two datasets: NYT and DBpedia, to study how fluctuations in hyper-parameters influence the performance of our method. The experiment is conducted on a fixed random seed (42). We present results on the range of relative performance change in different values of hyper-parameters compared to default settings. As shown in Table 7, These performance changes are within a reasonable range. Our method does not need to fine-tune these hyper-parameters.

Additional results on imbalance tolerance and ablation study are reported in Appendix D.

5 Related Work

Open-world Learning. Traditional open-world recognition methods (Bendale and Boult, 2015; Rudd et al., 2017; Boult et al., 2019) aim to incrementally extend the set of seen classes with new unseen classes. These methods require human involvement to label new classes. Recently, ORCA (Cao et al., 2021) defined open-world semi-supervised classification in the image domain and proposed a general solution which utilized unlabeled data in the learning stage and did not require any human effort. However, this method's performance is not robust enough for the imbalanced data in the text domain. In contrast, our work is applicable for infrequent classes and exploits the fact that the input is words which are class-indicative.

Extremely Weak Supervision in NLP. Aharoni and Goldberg (2020) showed that the average of BERT token representations can preserve documents' domain information. X-Class (Wang et al., 2021b) followed this idea to propose the extremely weak supervision setting where text classification only relies on the name of the class as supervision. However, such methods can not transfer to open-world classification naively as they cannot detect unseen classes. Our method leverages such extremely weak supervision methods as a subroutine to help the clustering of documents. But importantly, we note that such methods cannot be applied straightforwardly as they also are sensitive to noise and too similar classes. We show that our general idea of using class-indicative words can further help an extremely weak supervision method to obtain stable performance.

Joint Clustering with Downstream Tasks. To some sense, our method leverages partly an idea called joint clustering, which some recent works (Caron et al., 2018; Asano et al., 2020) in the image domain achieved high performance through jointly performing clustering and image classification. Their main idea is to utilize clustering to extract the hidden information of image representations and generate pseudo-labels, which in turn provide supervision for classification training and ultimately guide the co-improvement of representation and clustering. However, the crucial difference is that their methods already know the predefined classes and highly depend on strong assumptions like all classes share the same size to obtain excellent performance. Conversely, OW-Class utilizes the general idea of joint clustering in an open-world setting where the classes may be too fine-grained and noisy. We address these unique challenges via the class-indicative words we propose and show that our methodology can not only estimate the precise number of classes but also tolerate imbalanced data distribution.

6 Conclusions and Future Work

In this paper, we introduce the open-world semi-supervised classification task in the text domain, identify the key challenges and unique opportunities, and then propose OW-Class which can achieve quite decent performance. OW-Class starts with an overestimated number of classes and constructs an iterative refinement framework that jointly performs class nomination and document clustering, leading to iterative mutual enhancement. Extensive experiments demonstrate the effectiveness and stability of OW-Class. In the future, we plan to extend the open-world setting to many other NLP tasks. We also believe that open-world text classification can be conducted with even less human effort, for example, by only requiring user-provided seed words or class names for those seen classes.

7 Ethical Considerations

In this paper, we propose an approach to the text classification problem, a fundamental task in natural language processing to efficiently classify documents such as news reports. We conducted experiments on four publicly available and widely used datasets and did not observe any risky classification information, so we believe that our approach is entirely ethically flawless and will not harm vulnerable groups.

References

Roee Aharoni and Yoav Goldberg. 2020. Unsupervised domain clusters in pretrained language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics.

Dana Angluin and Philip Laird. 1988. Learning from noisy examples. *Machine Learning*, 2(4):343–370.

Yuki M. Asano, Christian Rupprecht, and Andrea Vedaldi. 2020. Self-labelling via simultaneous clustering and representation learning. In *International Conference on Learning Representations (ICLR)*.

Abhijit Bendale and Terrance Boult. 2015. Towards open world recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1893–1902.

Terrance E Boult, Steve Cruz, Akshay Raj Dhamija, Manuel Gunther, James Henrydoss, and Walter J Scheirer. 2019. Learning and the unknown: Surveying steps toward open world recognition. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 9801–9807.

Kaidi Cao, Maria Brbic, and Jure Leskovec. 2021. Open-world semi-supervised learning. In *International Conference on Learning Representations*.

Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. 2018. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European conference on computer vision (ECCV)*, pages 132–149.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Richard O Duda, Peter E Hart, and David G Stork. 1973. *Pattern classification and scene analysis*, volume 3. Wiley New York.

Andre Esteva, Brett Kuprel, Roberto A Novoa, Justin Ko, Susan M Swetter, Helen M Blau, and Sebastian Thrun. 2017. Dermatologist-level classification of skin cancer with deep neural networks. *nature*, 542(7639):115–118.

Jacob Goldberger and Ehud Ben-Reuven. 2017a. Temporal ensembling for semi-supervised learning. In *International Conference on Learning Representations*.

Jacob Goldberger and Ehud Ben-Reuven. 2017b. Training deep neural-networks using a noise adaptation layer. In *International Conference on Learning Representations*.

Kai Han, Andrea Vedaldi, and Andrew Zisserman. 2019. Learning to discover novel visual categories via deep transfer clustering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8401–8409.

Karen Spärck Jones. 1972. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28:11–21.

Durk P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. 2014. Semi-supervised learning with deep generative models. *Advances in neural information processing systems*, 27.

Ken Lang. 1995. Newsweeder: Learning to filter netnews. In *Machine Learning Proceedings 1995*, pages 331–339. Elsevier.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *nature*, 521(7553):436–444.

Dong-Hyun Lee et al. 2013. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, page 896.

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.

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 2018, Torino, Italy, October 22-26, 2018*, pages 983–992. ACM.

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 2020, Online, November 16-20, 2020, pages 9006–9017. Association for Computational Linguistics.

Ethan M Rudd, Lalit P Jain, Walter J Scheirer, and Terrance E Boult. 2017. The extreme value machine. *IEEE transactions on pattern analysis and machine intelligence*, 40(3):762–768.

David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. 2016. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489.

Fangbo Tao, Chao Zhang, Xiusi Chen, Meng Jiang, Tim Hanratty, Lance Kaplan, and Jiawei Han. 2018. Doc2cube: Allocating documents to text cube without labeled data. In *2018 IEEE International Conference on Data Mining (ICDM)*, pages 1260–1265.

Zihan Wang, Chengyu Dong, and Jingbo Shang. 2021a. "average" approximates "first principal component"? an empirical analysis on representations from neural language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 5594–5603. Association for Computational Linguistics.

Zihan Wang, Dheeraj Mekala, and Jingbo Shang. 2021b. 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*, pages 3043–3053, Online. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28.

Yunyi Zhang, Jiaming Shen, Jingbo Shang, and Jiawei Han. 2020. Empower entity set expansion via language model probing. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8151–8160, Online. Association for Computational Linguistics.

Xiaojin Zhu. 2005. Semi-supervised learning literature survey. *world*, 10:10.

A Limitations

As an initial attempt in open-world semi-supervised text classification, our method and evaluation metric still have the following limitations.

A.1 Theoretical Guarantees

We are aware that we made quite a few design choices without a solid theoretical guarantee. For example, there might be different choices of similarity functions for our MLP features or a different number of statistically representative words to extract.

Nevertheless, as a pioneering work in open-world text classification, our goal is to design a working algorithm and promote the power of class-indicative words. In addition, in order to make sure that our method is stable and robust under our setting, we also conducted extensive hyper-parameter sensitivity studies.

A.2 Evaluation Criteria

As shown in Sec. 4.4, experimental analysis indicates that OW-Class produces some fine-grained classes, which would not show up on our purity-based evaluation metric. In an extremal If there is no penalty for generating too many subclasses, dividing all documents into individual classes can be a shortcut to a perfect score. However, we also note that the problem of the deficit of a perfect metric stems from finding a good metric for clustering, which has been studied for a long time without a universal solution. Although it is impossible to completely avoid some differences in the metric evaluation, we tried our best to mitigate the differences and conduct a fair comparison.

First, the criterion for evaluation is the most reasonable one we could find. We investigated ORCA and other related work including clustering and topic modeling and did not find a metric that balances the number of predicted classes and the purity of predicted classes. Finding an appropriate metric for this task is a separate research problem.

Also, since the bias of the metric is caused by the potentially different numbers of predicted classes in different methods, we paid attention to this number. In a large portion of our main experiments, the comparisons are all based on the same number of classes, either provided by ground truth or by our method (Table 2). With such a controlled number of classes, we believe the evaluation is fair and reliable. We can see from the predicted class names in Sec. 4.4 and Appendix D that the additional classes our method identified are meaningful and distinct from existing classes in datasets, demonstrating the importance of discovering new classes.

B Implementation Details

B.1 Algorithm

We summarize our iterative refinement framework in Algorithm 1.

Algorithm 1 Iterative Refinement Framework

```
Input: clusters $\mathbf{C}$, document representations $\mathbf{R}$
while there are still redundant clusters do
    Find class-indicative words $\mathbf{W}$
    for each cluster $\mathbf{C}_i$ in $\mathbf{C}$ do
        Train MLP and rank $\mathbf{W}$
        Select possible names $\mathbf{S}_i$ from $\mathbf{W}$
        Compute cluster coherence $\eta_i$ (Eq. 1)
    for each pair $\mathbf{S}_i, \mathbf{S}_j$ do
        if $\mathbf{S}_i \cap \mathbf{S}_j \neq \emptyset$ and $\eta_i \leq \eta_j$ then
            Remove $\mathbf{C}_i$
    Re-estimate class names $\mathbf{S}$
    update $\mathbf{R}, \mathbf{C}$ based on $\mathbf{S}$
```

B.2 Implementation Details of OW-Class

In our experiments, we fine-tune the pre-trained BERT-base-uncased model provided in Huggingface's `Transformers` library (Wolf et al., 2019).

In each iteration, we use 25 times the number of seen classes as $|\mathbf{W}^{sem}|$. For each cluster, we fix $|\mathbf{W}_i^{stat}| = 50$ to get sufficient statistically outstanding words.In class nomination, the quality ratio threshold is $\beta = 0.6$. The analysis of hyper-parameter sensitivity is shown in Sec. 4.4.

Initial Clusters. The iterative refinement framework should start with a reasonable (over-) estimation of initial clusters based on the initial class names. We first make a bold guess (refer to Appendix C) on the number of classes as previous work (Cao et al., 2021) did. The guess can be several times larger than the actual number of classes. Afterwards, we utilize CGExpan to propose a large number of possible class names, so as to cover the semantics of all possible classes. We take the static representations of these names as the initial class representations. Then, we use these class representations as input to X-Class and obtain the document representations. We fine-tune a BERT (Devlin et al., 2019) classifier with the given labeled documents. The CLS token representation after fine-tuning is concatenated with the X-Class representation for each document to form the initial document representations we use. We then run a GMM (Duda et al., 1973) on these representations to identify the clusters, where the number of clusters is our overestimation.

Details of MLP Features. To construct the MLP features, we first follow Wang et al. (2021b) and obtain *static representation* $\mathbf{s}_w$ for each word w in the input corpus, by averaging BERT's contextualized representations of all its appearances:

$$\mathbf{s}_w = \frac{\sum_{w'=w} \mathbf{t}_{w'}}{\sum_{w'=w} 1},$$

where w' are occurrences of the word in the corpus and $\mathbf{t}_{w'}$ is its contextualized word representation. This static representation is used as an anchor to measure the semantic similarity between words. Then we define quality features for a high-potential word to a cluster as the mean and variance of Euclidean distance and cosine similarity of static representations between them.

Statistical Metric. In Sec. 3.2, we follow Mekala and Shang (2020) to find statistically outstanding words within cluster i:

$$score_i(w) = \frac{s_i(w)}{size_i} \cdot \tanh\left(\frac{t_i(w)}{size_i}\right) \cdot \log\left(\frac{size_{all}}{s_{all}(w)}\right),$$

where $t_i(w)$ is the number of occurrences of the word w in documents belonging to cluster i, $s_i(w)$ indicates how many documents in cluster i contain the word w while $size_i$ indicates how many documents are in cluster i.

In the measurement, the first term tells how indicative a word is to a cluster, the second term measures how frequent this word is, and the third is a normalization based on the inverse document frequency.

Penalty Coefficient. In Sec. 3.2, we use a penalty coefficient to punish the words that are too generic to be semantically meaningful. In this section, we give our definition of the penalty coefficient $\mu(w, i)$ for each candidate class name w in cluster i,

$$\mu(w, i) = \log\left(\frac{Med\{rank_j(w) \mid 1 \leq j \leq C\}}{1 + rank_i(w)}\right),$$

where $rank_i(w)$ is the absolute rank number of w in cluster i based on MLP's prediction and $Med\{S\}$ is the median value of the set S.

The main idea of this formula is to obtain a coefficient to penalize those generic words (e.g., life, which might rank high in most clusters) from being selected as class names. The numerator of the fraction shows how the word behaves across all clusters while the denominator shows how it behaves in a specific cluster. The median rank of a generic word will be very close to the specific rank. Note that we allow one word as the class name of several clusters because of the initial overestimation, but if a word ranks high in more than half of the clusters, it is considered a generic word that must be filtered.

Such penalization and normalization are similar to the inverse document frequency term in Information Retrieval. Therefore, we follow the design and choose to divide the two values and take the logarithm. Similar to the inverse document frequency, this penalty coefficient lowers the chance of selecting a generic word but will not harm proper words.

Iterative Framework. The iterative framework is solving the aforementioned two sub-problems one by one, obtaining class names from clusters and clusters from class names. The first sub-problem entails removing redundant clusters and the natural stopping criteria for the iteration is when no clusters are removed.

Final Text Classifier. The iteration ends with the determined number of classes and document distribution. The clusters in the final iteration provide high-quality pseudo labels for the unlabeled dataset. Following many previous works in (extremely) weak supervision (Meng et al., 2018; Mekala and Shang, 2020; Meng et al., 2020; Wang et al., 2021b), we train a classifier based on given labels and these pseudo labels to generalize such knowledge to new documents. This is a typical noisy training task (Angluin and Laird, 1988; Goldberger and Ben-Reuven, 2017b). The clusters in Sec. 3.3 are obtained through GMM (Duda et al., 1973) following X-Class, and the final text classifier is trained on a selected dataset that has high posterior probability in the final GMM. Experiments in Appendix D show the usefulness of this extra classifier.

B.3 Implementation Details of baselines

For ORCA, We migrate its method to the text domain by using BERT to obtain the feature representations. For CGExpan + X-Class, We first expand a large number of class names via CGExpan and employs X-Class to give each document a class-oriented representation. Then, we train a GMM and take the class name closest to the center of each cluster as the final name of the corresponding cluster. Finally we use these names to get the classification result by X-Class.

B.4 Computational Budget

All experiments are conducted using a 32-Core Processor and a single RTX A6000 GPU. The expected running time of OW-class is less than one hour, for a dataset with a sample size of 10,000 (e.g., 20News, NYT). A larger dataset may require more time; for DBpedia (a massive dataset with 560,000 samples), 10 hours is expected. Note, however, that ORCA cannot provide same-condition results in a single day.

C Initial Guessed Number of Classes

In initial clustering, OW-Class makes a bold guess on the number of classes at the initial stage. We treat this guessed number of classes as a hyper-parameter. Here we give our default scheme. We choose 5 times the number of seen classes as the estimation of total classes and also plus a imbalance factor P of the labeled dataset as compensation for GMM, since we assume that all classes will split into clusters of almost equal size in the initial clustering. That is,

$$P = \sum_{c \in \mathcal{C}_s} \left\lfloor \frac{NUM(c)}{\min_{c \in \mathcal{C}_s} NUM(c)} - 1 \right\rfloor,$$

where $NUM(c)$ is the number of labeled texts in class c. In a (almost) balanced dataset, $P = 0$. But for NYT (extremely imbalanced) in our setting, we have $P = 5$.

We further test OW-Class with different initial estimations. Figure 5 shows that when the initial number of classes is in a reasonable range, OW-Class can deliver a steady result that can outperform baselines in most instances. And its performance can get even better with more initial clusters. But OW-Class may suffer a significant performance drop when the initial class is too small (especially for extreme imbalanced datasets like NYT).

D Additional Results

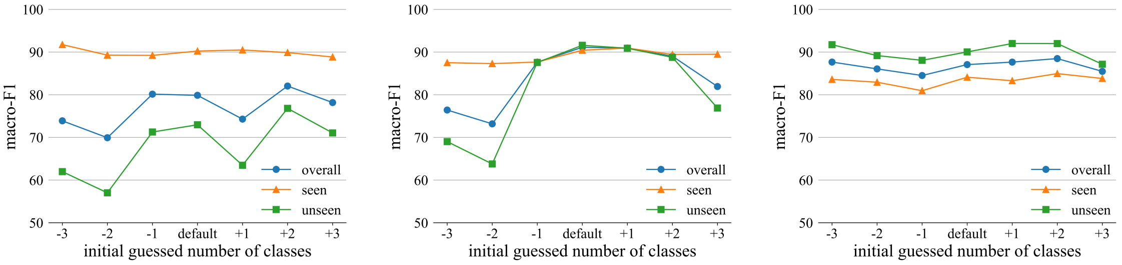
Effect of Extra Classifier. To verify the effectiveness of the extra classifier (see details in Appendix B) after the iterative refinement, we design OW-Class-GMM which obtains the labels from the GMM in the last iteration. The improvement of OW-Class over OW-Class-GMM in Table 8 shows the usefulness of the classifier training.

Imbalance Tolerance. We further show the performance of ORCA and OW-Class on imbalanced DBpedia for both seen and unseen classes.

As shown in Figure 6, ORCA's performance drops more than 40% on unseen classes, demonstrating its intolerance to imbalanced data distributions.

Micro-F$_1$ Scores. Tables 9 and 10 show the micro-F$_1$ scores of experiments in Sec. 4.4.

Examples of Class Nomination. We additionally show the full list of the class nomination in Table 11. Our class names are highly related to ground truth class names and human-understandable.

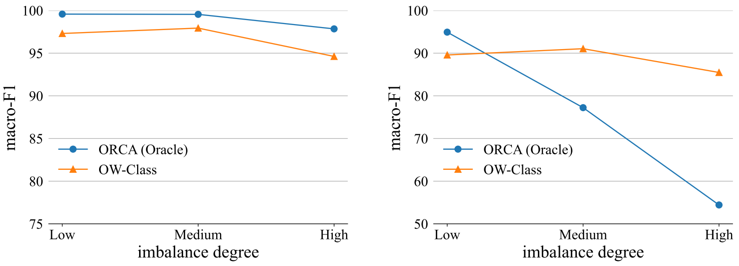


(a) 20News

(b) NYT

(c) AGNews

Figure 5: Sensitivity on the Initial Guessed Number of Classes for 20News, NYT and AGNews. The mean macro-F_1 scores over three runs are reported. OW-Class is slow on DBpedia, therefore not reported.



(a) seen

(b) unseen

Figure 6: Performance of ORCA and OW-Class on different imbalance degrees.

Table 8: Effectiveness Study of Extra Classifier. The mean macro-F_1 scores over three runs are reported.

Method	**20News**			**NYT**			**AGNews**			**DBpedia**		
	All	**Seen**	**Unseen**	**All**	**Seen**	**Unseen**	**All**	**Seen**	**Unseen**	**All**	**Seen**	**Unseen**
OW-Class	79.87	90.24	72.96	91.13	90.34	91.59	87.07	84.11	90.04	93.88	96.30	91.46
OW-Class-GMM	77.27	88.77	69.62	90.38	89.98	90.65	86.82	83.50	90.13	93.43	96.05	90.80

Table 9: Evaluations of compared methods and OW-Class. The mean micro-F_1 scores over three runs are reported.

Method	**Additional Input**	**20News**			**NYT**			**AGNews**			**DBpedia**		
		All	**Seen**	**Unseen**	**All**	**Seen**	**Unseen**	**All**	**Seen**	**Unseen**	**All**	**Seen**	**Unseen**
ORCA	None	32.57	48.97	0	63.82	79.36	18.52	30.82	46.27	0	**98.93**	**99.52**	**98.33**
OW-Class		**78.36**	**89.65**	**72.04**	**94.63**	**96.46**	**91.28**	87.91	84.08	90.09	93.01	95.72	91.58
ORCA (Our estimation)	Number of classes	75.80	87.17	69.66	68.12	84.20	38.38	83.89	88.31	81.56	79.59	99.22	69.70
BERT+GMM (Our estimation)		52.38	69.94	41.77	75.49	89.89	51.34	61.84	67.75	59.17	77.24	96.25	68.20
CGExpan+X-Class (Our estimation)		72.87	81.59	68.76	85.47	96.41	66.84	78.52	81.33	77.07	84.90	92.97	76.65
ORCA (Oracle)		63.04	88.29	49.82	32.91	40.67	22.08	**91.88**	**88.08**	**93.88**	98.93	99.52	98.33
BERT+GMM (Oracle)		42.55	56.71	29.97	66.02	80.74	36.06	48.46	61.86	33.30	75.39	97.08	54.34
CGExpan+X-Class (Oracle)		67.10	79.09	61.09	85.54	93.45	73.26	84.22	79.72	86.55	70.78	89.23	52.41

Table 10: Ablation study of OW-Class. The mean micro-F_1 scores over three runs are reported.

Method	**20News**			**NYT**			**AGNews**			**DBpedia**		
	All	**Seen**	**Unseen**	**All**	**Seen**	**Unseen**	**All**	**Seen**	**Unseen**	**All**	**Seen**	**Unseen**
OW-Class	78.36	89.65	72.04	94.63	96.46	91.28	87.91	84.08	90.09	93.01	95.72	91.58
OW-Class-noIter	63.95	83.96	53.93	81.27	94.59	58.87	74.49	77.89	72.84	91.28	96.61	88.51
OW-Class-oneIter	78.18	89.84	71.77	93.77	96.25	89.30	85.76	82.35	87.50	94.74	96.90	93.50
OW-Class-X	74.25	82.79	68.96	91.63	95.57	83.80	89.57	84.79	92.21	91.11	93.77	88.04

Table 11: Examples of the class nomination.

Dataset	Ground Truth	OW-Class
20News	science	encryption, electronics, orbit
	religion	Christianity
	computer	computer
	politics	history
	sports	sports, baseball
NYT	sport	sport, tennis, quarterback
	business	economy
	science	scientist
	art	theater
	politics	politics
AGNews	politics	executive, Iraq
	sports	game
	technology	technology
	business	business
DBpedia	athlete	footballer, Olympics
	artist	painter, singer
	company	company
	school	school
	politics	politician
	transportation	aircraft, locomotive
	building	church, bridge
	river	river
	village	village
	animal	animal, snail
	plant	plant
	album	album
	book	book
	film	film