X-Class: Text Classification with Extremely Weak Supervision

Zihan Wang [1] **Dheeraj Mekala** [1] **Jingbo Shang** [1,2]

[1] Department of Computer Science and Engineering, University of California San Diego, CA, USA

[2] Halıcıoğlu Data Science Institute, University of California San Diego, CA, USA

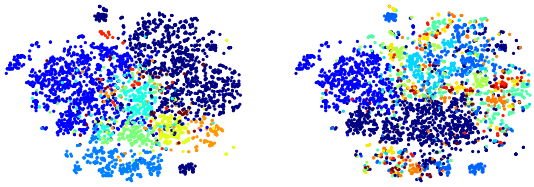
{ziw224, dmekala, jshang}@ucsd.edu

Abstract

In this paper, we explore text classification with *extremely weak supervision*, i.e., only relying on the surface text of class names. This is a more challenging setting than the seed-driven weak supervision, which allows a few seed words per class. We opt to attack this problem from a representation learning perspective—ideal document representations should lead to nearly the same results between clustering and the desired classification. In particular, one can classify the same corpus differently (e.g., based on topics and locations), so document representations should be adaptive to the given class names. We propose a novel framework X-Class to realize the adaptive representations. Specifically, we first estimate class representations by incrementally adding the most similar word to each class until inconsistency arises. Following a tailored mixture of class attention mechanisms, we obtain the document representation via a weighted average of contextualized word representations. With the prior of each document assigned to its nearest class, we then cluster and align the documents to classes. Finally, we pick the most confident documents from each cluster to train a text classifier. Extensive experiments demonstrate that X-Class can rival and even outperform seed-driven weakly supervised methods on 7 benchmark datasets.

1 Introduction

Weak supervision has been recently explored in text classification to save human effort. Typical forms of weak supervision include a few labeled documents per class (Meng et al., 2018; Jo and Cinarel, 2019), a few seed words per class (Meng et al., 2018, 2020a; Mekala and Shang, 2020; Mekala et al., 2020), and other similar open-data (Yin et al., 2019).Though much weaker than a fully annotated corpus, these forms still require non-trivial, corpus-specific knowledge from experts. For example, nominating seed words requires experts to consider their relevance to not only the desired classes but also the input corpus; To acquire a few labeled documents per class, unless the classes are balanced, one needs to sample and annotate a much larger number of documents to cover the minority class.



(a) NYT-Topics (b) NYT-Locations

Figure 1: Visualizations of the same news corpus using Average BERT Representations on two criteria. Colors denote different classes.

In this paper, we focus on *extremely weak supervision*, i.e., only relying on the surface text of class names. This setting is much more challenging than the ones above, and can be considered as almost-unsupervised text classification.

We opt to attack this problem from a representation learning perspective—ideal document representations should lead to nearly the same result between clustering and the desired classification. Recent advances in contextualized representation learning using neural language models have demonstrated the capability of clustering text to domains with high accuracy (Aharoni and Goldberg, 2020). Specifically, a simple average of word representations is sufficient to group documents on the same topic together. However, the same corpus could be classified using various criteria other than topics, such as locations and sentiments. As visualized in Figure 1, such class-invariant representations separate topics well but mix up locations. Therefore, it is a necessity to make document representations adaptive to the user-specified class names.

We propose a novel framework X-Class to conduct text classification with extremely weak supervision, as illustrated in Figure 2. Firstly, we esti-

Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 3043–3053
June 6–11, 2021.

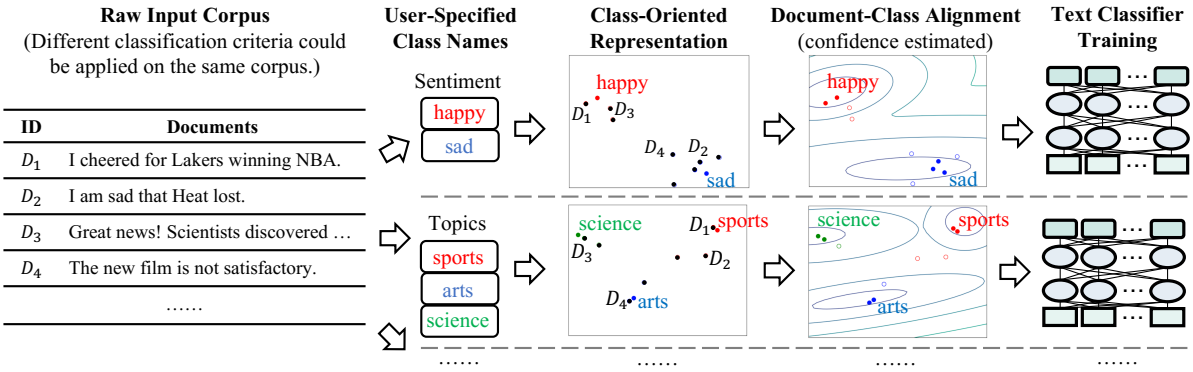


Figure 2: An overview of our X-Class. Given a raw input corpus and user-specified class names, we first estimate a class-oriented representation for each document. And then, we align documents to classes with confidence scores by clustering. Finally, we train a supervised model (e.g., BERT) on the confident document-class pairs.

mate class representations by incrementally adding the most similar word to each class and recalculating its representation. Following a tailored mixture of class attention mechanisms, we obtain the document representation via a weighted average of contextualized word representations. These representations are based on pre-trained neural language models, and they are supposed to be in the same latent space. We then adopt clustering methods (e.g., Gaussian Mixture Models) to group the documents into K clusters, where K is the number of desired classes. The clustering method is initialized with the prior knowledge of each document assigned to its nearest class. We preserve this assignment so we can easily align the final clusters to the classes. In the end, we pick confident documents from each cluster to form a pseudo training set, based on which, we can train any document classifier. In our implementation, we use BERT as both the pre-trained language model and the text classifier. Compared with existing weakly supervised methods, X-Class has a stronger and more consistent performance on 7 benchmark datasets, despite some of them using at least 3 seed words per class. It is also worth mentioning that X-Class has a much more mild requirement on the existence of class names in the corpus, whereas existing methods rely on the variety of contexts of the class names.

Our contributions are summarized as follows.

- We advocate an important but not-well-studied problem of text classification with extremely weak supervision.
- We develop a novel framework X-Class to attack this problem from a representation learning perspective. It estimates high-quality, class-oriented document representations based on pre-trained neural language models so that the confident clustering examples could form pseudo training set for any document classifiers to train on.
- We show that on 7 benchmark datasets, X-Class achieves comparable and even better performance than existing weakly supervised methods that require more human effort.

Reproducibility. We will release both datasets and codes on Github[1].

2 Preliminaries

In this section, we formally define the problem of text classification with extremely weak supervision. And then, we brief on some preliminaries about BERT (Devlin et al., 2019), Attention (Luong et al., 2015) and Gaussian Mixture Models.

Problem Formulation. The extremely weak supervision setting confines our input to only a set of documents $D_i, i \in \{1, ..., n\}$ and a list of class names $c_j, j \in \{1, ..., k\}$. The class names here are expected to provide hints about the desired classification objective, considering that different criteria (e.g., topics, sentiments, and locations) could classify the same set of documents. Our goal is to build a classifier to categorize a (new) document into one of the classes based on the class names.

Seed-driven weak supervision requires *carefully designed* label-indicative keywords that concisely define what a class represents. This requires human experts to understand the corpus extensively. One of our motivations is to relax this burdensome requirement. Interestingly, in experiments, our proposed X-Class using extremely weak supervision can offer comparable and even better performance than the seed-driven methods.

[1] https://github.com/ZihanWangKi/XClass

BERT. BERT is a pre-trained masked language model with a transformer structure (Devlin et al., 2019). It takes one or more sentences as input, breaks them up into word-pieces, and generates a contextualized representation for each word-piece. To handle long documents in BERT, we apply a sliding window technique. To retrieve representations for words, we average the representations of the word's word-pieces. BERT has been widely adopted in a large variety of NLP tasks as a backbone. In our work, we will utilize BERT for two purposes: (1) representations for words in the documents and (2) the supervised text classifier.

Attention. Attention mechanisms assign weights to a sequence of vectors, given a context vector (Luong et al., 2015). It first estimates a hidden state $\tilde{h}_j = K(h_j, c)$ for each vector h_j, where K is a similarity measure and c is the context vector. Then, the hidden states are transformed into a distribution via a softmax function. In our work, we use attentions to assign weights to representations, which we then average them accordingly.

Gaussian Mixture Model. Gaussian Mixture Model (GMM) is a traditional clustering algorithm (Duda and Hart, 1973). It assumes that each cluster is generated through a Gaussian process. Given an initialization of the cluster centers and the co-variance matrix, it iteratively optimizes the point-cluster memberships and the cluster parameters following an Expectation–Maximization framework. Unlike K-Means, it does not restrict clusters to have a perfect ball-like shape. Therefore, we apply GMM to cluster our document representations.

3 Our X-Class Framework

As shown in Figure 2, our X-Class framework contains three modules: (1) class-oriented document representation estimation, (2) document-class alignment through clustering, and (3) text classifier training based on confident labels.

3.1 Class-oriented Document Representation

Ideally, we wish to have some document representations such that clustering algorithms can find k clusters very similar to the k desired classes.

We propose to estimate the document representations and class representations based on pre-trained neural language models. Algorithm 1 is an overview. In our implementation, we use BERT as an example. For each document, we want its document representation to be similar to the class representation of its desired class.

Algorithm 1: Class-Oriented Document Representation Estimation

Input: n documents D_i, k class names c_j, max number of class-indicative words T, and attention mechanism set $\mathcal{M}$
Compute $\mathbf{t}_{i,j}$ (contextualized word rep.)
Compute $\mathbf{s}_w$ for all words (Eq. 1)
// class rep. estimation
for $l = 1 \dots k$ **do**
  $\mathcal{K}_l \leftarrow \langle\, c_l \,\rangle$
  for $i = 2 \dots T$ **do**
    Compute $\mathbf{x}_l$ based on $\mathcal{K}_l$ (Eq. 2)
    $w = \arg\max_{w \notin \mathcal{K}_l} sim(\mathbf{s}_w, \mathbf{x}_l)$
    Compute $\mathbf{x}'_l$ based on $\mathcal{K}_l \oplus \langle w \rangle$
    // consistency check
    if $\mathbf{x}'_l$ *changes the words in* $\mathcal{K}_l$ **then**
      break
    else
      $\mathcal{K}_l \leftarrow \mathcal{K}_l \oplus \langle w \rangle$
// document rep. estimation
for $i = 1 \dots n$ **do**
  for *attention mechanism* $m \in \mathcal{M}$ **do**
    Rank $D_{i,j}$ according to m
    $r_{m,j} \leftarrow$ the rank of $D_{i,j}$
  Rank $D_{i,j}$ according to $\prod_m r_{m,j}$
  $r_j \leftarrow$ the final rank
  Compute $\mathbf{E}_i$ (Eq. 3)
Return All document representations $\mathbf{E}_i$.

Aharoni and Goldberg (2020) demonstrated that contextualized word representations generated by BERT can preserve the domain (i.e., topic) information of documents. Specifically, they generated document representations by averaging contextualized representations of its constituent words, and they observed these document representations to be very similar among documents belonging to the same topic. This observation motivates us to "classify" documents by topics in an unsupervised way. However, this unsupervised method may not work well on criteria other than topics. For example, as shown in Figure 1, such document representations work well for topics but poorly for locations.

We therefore incorporate information from the given class names and obtain *class-oriented* document representations. We break down this module into two parts, (1) class representation estimation and (2) document representation estimation.

Class Representation Estimation. Inspired by seed-driven weakly supervised methods, we argue that a few keywords per class would be enough

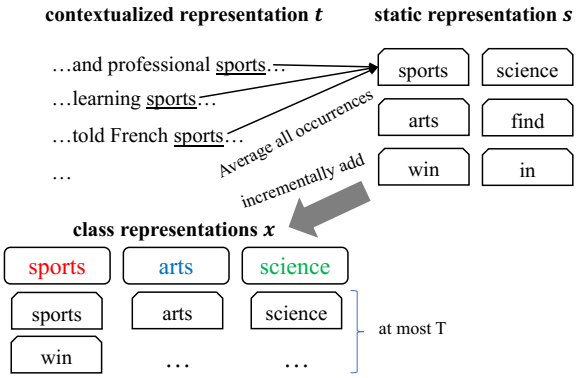


Figure 3: Overview of Our Class Rep. Estimation.

to understand the semantics of the user-specified classes. Intuitively, the class name could be the first keyword we can start with. We propose to incrementally add new keywords to each class to enrich our understanding.

Figure 3 shows an overview of our class representation estimation. First, for each word, we obtain its *static representation* via averaging the contextualized representations of all its occurrences in the input corpus. For words that are broken into word-piece tokens, we average all the token representations as the word's representation. Then, we define the static representation $\mathbf{s}_w$ of a word w as

$$\mathbf{s}_w = \frac{\sum_{D_{i,j}=w} \mathbf{t}_{i,j}}{\sum_{D_{i,j}=w} 1} \quad (1)$$

where $D_{i,j}$ is the j-th word in the document D_i and $\mathbf{t}_{i,j}$ is its contextualized word representation. Ethayarajh (2019) adopted a similar strategy of estimating a static representation using BERT. Such static representations are used as anchors to initialize our understanding of the classes.

A straightforward way to enrich the class representation is to take a fixed number of words similar to the class name and average them to get a class representation. However, it suffers from two issues: (1) setting the same number of keywords for all classes may hurt the minority classes, and (2) a simple average may shift the semantics away from the class name itself. As an extreme example, when the 99% of documents are talking about *sports* and the rest 1% are about *politics*, it is not reasonable to add as many keywords as *sports* to *politics*—it will diverge the *politics* representation.

To address these two issues, we iteratively find the next keyword for each class and recalculate the class representation by a weighted average on all the keywords found. We stop this iterative process

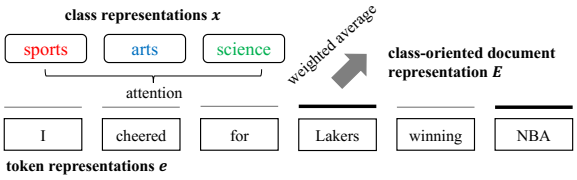


Figure 4: Overview of Our Document Rep. Estimation.

when the new representation is not consistent with the previous one. In this way, different classes will have a different number of keywords adaptively. Specifically, we define a comprehensive representation $\mathbf{x}_l$ for a class l as a weighted average representation based on a ranked list of keywords $\mathcal{K}_l$. The top-ranked keywords are expected to have more similar static representations to the class representation. Assuming that the similarities follow Zipf's laws distribution (Powers, 1998), we define the weight of the i-th keyword as $1/i$. That is,

$$\mathbf{x}_l = \frac{\sum_{i=1}^{|\mathcal{K}_l|} 1/i \cdot \mathbf{s}_{\mathcal{K}_{l,i}}}{\sum_{i=1}^{|\mathcal{K}_l|} 1/i} \quad (2)$$

For a given class, the first keyword in this list is always the class name. In the i-th iteration, we retrieve the out-of-list word with the most similar static representation to the current class representation. We then calculate a new class representation based on all the $i+1$ words. We stop this expansion if we already have enough (e.g., $T = 100$) keywords, or the new class representation cannot yield the same set of top-i keywords in our list. In our experiments, some classes indeed stop before reaching 100 keywords.

Document Representation Estimation. Intuitively, the content of each document should stick to its underlying class. For example, in the sentence "*I cheered for Lakers winning NBA*", its content covers *sports* and *happy* classes, but not *arts*, *politics*, or *sad*. Therefore, we assume that each word in a document is either similar to its desired class's representation or unrelated to all classes. Based on this assumption, we upgrade the simple average of contextualized word representations (Aharoni and Goldberg, 2020) to a weighted average. Specifically, we follow the popular attention mechanisms to assign weights to the words based on their similarities to the class representations.

Figure 4 shows an overview of our document representation estimation. We propose to employ a mixture of attention mechanisms to make it more robust. For the j-th word in the i-th document

$D_{i,j} = w$, there are two possible representations: (1) the contextualized word representation $\mathbf{t}_{i,j}$ and (2) the static representation of this word $\mathbf{s}_w$. The contextualized representations disambiguate words with multiple senses by considering the context, while the static version accounts for outliers that may exist in documents. Therefore, it is reasonable to use either of them as the word representation $\mathbf{e}$ for attention mechanisms. Given the class representations $\mathbf{x}_c$, we define two attention mechanisms:

- **one-to-one**: $h_{i,j} = \max_c\{\cos(\mathbf{e}, \mathbf{x_c})\}$. It captures the maximum similarity to one class. This is useful for detecting words that are specifically similar to one class, such as *NBA* to *sports*.
- **one-to-all**: $h_{i,j} = \cos\left(\mathbf{e}, \text{avg}_c\{\mathbf{x}_c\}\right)$ which is the similarity to the average of all classes. This ranks words by how related it is to the general set of classes in focus.

Combining 2 choices of $\mathbf{e}$ and 2 choices of attention mechanisms totals 4 ways to compute each word's attention weight. We further fuse these attention weights in an unsupervised way. Instead of using the similarity values directly, we rely on the rankings. Specifically, we sort the words decreasingly based on attention weights to obtain 4 ranked lists. Following previous work (Mekala and Shang, 2020; Tao et al., 2018), we utilize the geometric mean of these ranks for each word and then form a unified ranked list. Like class representation estimation, we follow Zipf's law and assign a weight of $1/r$ to a word ranked at the r-th position in the end. Finally, we obtain the document representation $\mathbf{E}_i$ from $\mathbf{t}_{i,j}$ with these weights.

$$\mathbf{E}_i = \frac{\sum_j \frac{1}{r_j} \cdot \mathbf{t}_{i,j}}{\sum_j \frac{1}{j}} \tag{3}$$

3.2 Document-Class Alignment

One straightforward idea to align the documents to classes is simply finding the most similar class based on their representations. However, document representations not necessarily distribute ball-shape around the class representation—the dimensions in the representation can be correlated freely.

To address this challenge, we leverage the Gaussian Mixture Model (GMM) to capture the covariances for the clusters. Specifically, we set the number of clusters the same as the number of classes k and initialize the cluster parameters based on the prior knowledge that each document D_i is assigned to its nearest class L_i, as follows.

$$L_i = \arg\max_c cos(\mathbf{E}_i, \mathbf{x}_c) \tag{4}$$

We use a tied co-variance matrix across all clusters since we believe classes are similar in granularity. We cluster the documents while remembering the class each cluster is initialized to. In this way, we can align the final clusters to the classes.

Considering the potential redundant noise in these representations, we also apply principal component analysis (PCA) for dimension reduction following the experience in topic clustering (Aharoni and Goldberg, 2020). By default, we fix the PCA dimension $P = 64$.

3.3 Text Classifier Training

The alignment between documents and classes produce high-quality pseudo labels for documents in the training set. To generalize such knowledge to unseen text documents, we train a text classifier using these pseudo labels as ground truth. This is a classical noisy training scenario (Angluin and Laird, 1987; Goldberger and Ben-Reuven, 2017). Since we know how confident we are on each instance (i.e., the posterior probability on its assigned cluster in GMM), we select the most confident ones to train a text classifier (e.g., BERT). By default, we set a confidence threshold $\delta = 50\%$, i.e., the top 50% instances are selected for classifier training.

4 Experiments

We conduct extensive experiments to show and ablate the performance of X-Class.

4.1 Compared Methods

We compare with two seed-driven weakly supervised methods. **WeSTClass** (Meng et al., 2018) generates pseudo-labeled documents via word embeddings of keywords and employs a self-training module to get the final classifier. We use the CNN version of WeSTClass as it is reported to have better performance compared to the HAN version. **ConWea** (Mekala and Shang, 2020) utilizes pre-trained neural language models to make the weak supervision contextualized. In our experiments, we feed at least 3 seed words per class to these two.

We also compare with **LOTClass** (Meng et al., 2020b), which works under the extremely weak supervision setting. In their experiments, it mostly relies on class names but has used a few keywords

Table 1: An overview of our 7 benchmark datasets. They cover various domains and classification criteria. The imbalance factor of a dataset refers to the ratio of its largest class's size to the smallest class's size.

	AGNews	**20News**	**NYT-Small**	**NYT-Topic**	**NYT-Location**	**Yelp**	**DBpedia**
Corpus Domain	News	News	News	News	News	Reviews	Wikipedia
Class Criterion	Topics	Topics	Topics	Topics	Locations	Sentiment	Ontology
# of Classes	4	5	5	9	10	2	14
# of Documents	120,000	17,871	13,081	31,997	31,997	38,000	560,000
Imbalance	1.0	2.02	16.65	27.09	15.84	1.0	1.0

Table 2: Evaluations of Compared Methods and X-Class. Both micro-/macro-F_1 scores are reported. Supervised provides an upper bound. † indicates the use of at least 3 seed words per class. ‡ indicates the use of only class names. § refers to number coming from other papers. ConWea is too slow on DBpedia, therefore not reported.

Model	**AGNews**	**20News**	**NYT-Small**	**NYT-Topic**	**NYT-Location**	**Yelp**	**DBpedia**
Supervised	93.99/93.99	96.45/96.42	97.95/95.46	94.29/89.90	95.99/94.99	95.7/95.7	98.96/98.96
WeSTClass†	82.3/82.1§	71.28/69.90	91.2/83.7§	68.26/57.02	63.15/53.22	81.6/81.6§	81.42/81.19
ConWea†	74.6/74.2	75.73/73.26	95.23/90.79	**81.67/71.54**	85.31/83.81	71.4/71.2	N/A
LOTClass‡	**86.89/86.82**	73.78/72.53	78.12/56.05	67.11/43.58	58.49/58.96	87.75/87.68	86.66/85.98
X-Class ‡	85.74/85.66	**78.62/77.76**	**97.18/94.02**	79.02/68.55	**91.8/91.98**	**90.0/90.0**	**91.32/91.17**
Ablations							
X-Class-Rep‡	77.86/76.84	75.37/73.7	92.13/83.69	77.06/65.05	86.36/88.1	78.0/77.19	74.05/71.74
X-Class-Align‡	83.32/83.28	79.19/78.46	96.42/92.32	79.12/67.76	90.09/90.63	87.19/87.13	87.36/87.27
X-Class-ExactT‡	84.85/84.76	73.95/74.13	97.18/94.02	79.18/68.96	88.94/88.02	90.0/90.0	88.48/88.37
X-Class-KMeans‡	81.29/81.08	70.79/71.18	94.96/89.66	72.83/64.79	93.88/92.94	80.6/80.56	65.76/66.94

to elaborate on some difficult classes. In our experiments, we only feed the class names to it.

We denote our method as **X-Class**. To further understand the effects of different modules, we have four ablation versions. **X-Class-Rep** refers to the prior labels L_i derived based on class-oriented document representation. **X-Class-Align** refers to the labels obtained after document-class alignment. **X-Class-ExactT** refers to not doing consistency check when estimating class representations, and having exactly T class words. **X-Class-KMeans** refers to using K-Means (Lloyd, 1982) of GMM during document class alignment.

We present the performance of supervised models, serving as an upper-bound for X-Class. Specifically, **Supervised** refers to a BERT model cross-validated on the training set with 2 folds (matching our confidence selection threshold).

4.2 Datasets

Many different datasets have been adopted to evaluate weakly supervised methods in different works. This makes it hard for systematic comparison.

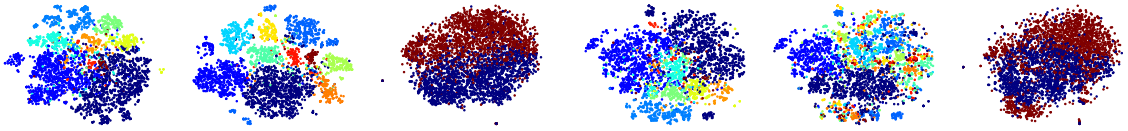
In this paper, we pool the most popular datasets to establish a benchmark on weakly supervised text classification. Table 1 provides an overview of our carefully selected 7 datasets, covering different text sources (e.g., news, reviews, and Wikipedia articles) and different criteria of classes (e.g., topics, locations, and sentiment).

- **AGNews** from (Zhang et al., 2015) (used in WeSTClass and LOTClass) is for topic categorization in news from AG's corpus.
- **20News** from (Lang, 1995)[2] (used in WeSTClass and ConWea) is for topic categorization in news.
- **NYT-Small** (used in WeSTClass and ConWea) is for classifying topic in New York Times news.
- **NYT-Topic** (used in (Meng et al., 2020a)) is another larger dataset collected from New York Times for topic categorization.
- **NYT-Location** (used in (Meng et al., 2020a)) is the same corpus as NYT-Topic but for locations. It is noteworthy to point out that many documents from this dataset talk about several countries simultaneously, so simply checking the location names will not lead to satisfactory results.
- **Yelp** from (Zhang et al., 2015) (used in WeSTClass) is for sentiment analysis in reviews.
- **DBpedia** from (Zhang et al., 2015) (used in LOTClass) is for topic classification based on titles and descriptions in DBpedia.

4.3 Experimental Settings

For all X-Class experiments, we report the performance under one fixed random seed. By default, we set $T = 100, P = 64, \delta = 50\%$. For contextualized token representations $\mathbf{t}_{i,j}$, we use the `BERT-base-uncased` to group more occur-

[2] http://qwone.com/~jason/20Newsgroups/



(a) Our Class-Oriented Document Representations (b) Simple Average of BERT Representations

Figure 5: t-SNE Visualizations of Representations. From left to right: NYT-Topics, NYT-Locations, Yelp.

Table 3: Example seed words and class names for methods for NYT-Small.

class	Seed words	Class name for X-Class
arts	dance,art, ballet,museum	arts
business	shares,stocks, markets,trading	business

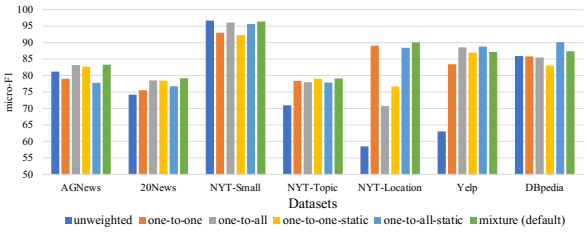


Figure 6: Effects of Attention Mechanisms. We focus on X-Class-Align to show their direct effects.

rences of the same word. For supervised model training, we follow BERT fine-tuning (Wolf et al., 2019) with all hyper-parameters unchanged.

For both WeSTClass and ConWea, we have tried our best to find keywords for the new datasets. Table 3 shows an example on the seed words selected for them on the NYT-Small dataset. For LOTClass, we tune their hyper-parameters *match_threshold* and *mcp_epoch*, and report the best performance during their self-train process.

4.4 Performance Comparison and Analysis

From Table 2, one can see that X-Class achieves the best overall performance. It is only 1% to 2% away from LOTClass and ConWea on AGNews and NYT-Topics, respectively. Note that, ConWea consumes at least 3 keywords per class.

It is noteworthy that X-Class can approach the supervised upper bound to a small spread, especially on the NYT-Small dataset.

Ablation on Modules. X-Class-Rep has achieved high scores (e.g., on both NYT-Topics and NYT-Locations) showing success of our class-oriented representations. The improvement of X-Class-Align over X-Class-Rep demonstrates the usefulness of our clustering module. It is also clear that the classifier training is beneficial by comparing X-Class and X-Class-Align.

Ablation on Consistency Check. The consistency check in class representation estimation allows an adaptive number of keywords for each class. Without it leads to a diverged class understanding and degrading performance, as shown in Table 2.

Ablation on Clustering Methods. Table 2 also shows that K-Means performs poorly on most datasets. This matches our previous analysis as K-Means assumes a hard spherical boundary, while GMM models the boundary softly like an ellipse.

4.5 Effect of Attention

In Figure 5, we visualize our class-oriented document representations and the unweighted variants using t-SNE (Rauber et al., 2016). We can see that while the simple-average representations are well-separated like class-oriented representations in NYT-Topics, they are much mixed up in NYT-Locations and Yelp. We conjecture that this is because BERT representations has topic information as its most significant feature.

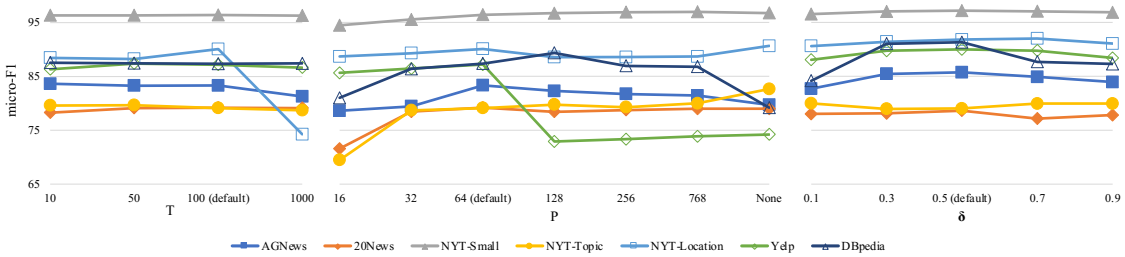
We have also tried using different attention mechanisms in X-Class. From the results in Figure 6, one can see that using a single mechanism, though not under-performing much, is less stable than our proposed mixture. The unweighted case works well on all four datasets that focus on news topics but not good enough on locations and sentiments.

4.6 Hyper-parameter Sensitivity in X-Class

Figure 7 visualizes the performance trend w.r.t. to the three hyper-parameters in X-Class, i.e., the limit of class words T in class representation estimation, the PCA dimension P in document-class alignment, and the confidence threshold δ in text classifier training.

Intuitively, a class doesn't have too many highly relevant keywords. One can confirm this in Figure 7(a) as the performance of X-Class is relatively stable unless T goes too large to 1000.

Choosing a proper PCA dimension could prune out redundant information in the embeddings and improve the running time. However, if P is too small or too large, it may hurt due to information



(a) T in Class Rep. Estimation (b) P in Document-Class Alignment (c) δ in Text Classifier Training

Figure 7: Hyper-parameter Sensitivity in X-Class. For T and P, we report the performance of X-Class-Align to explore their direct effects.

Table 4: Macro-F_1 score changes of methods when removing all but one occurrence of a class name.

Model	20News		NYT-Small	
	Original	Removed	Original	Removed
X-Class	77.76	74.48	94.02	93.29
LOTClass	72.53	8.82	56.05	29.53

loss or redundancy. One can observe this expected trend in Figure 7(b) on all datasets.

Typically, we want to select a reasonable number of confident training samples for the text classifier training. Too few training samples (i.e., too large δ) would lead to insufficient training data. Too many training samples (i.e., too small δ) would lead to too noisy training data. Figure 7(c) shows that $\delta \in [0.3, 0.9]$ is a good choice on all datasets.

4.7 Requirements on Class Names

Compared with previous works (Meng et al., 2018; Mekala and Shang, 2020; Meng et al., 2020b), our X-Class has a significantly more mild requirement on human-provided class names in terms of quantity and quality. We have conducted an experiment in Table 4 for X-Class on 20News and NYT-Small by deleting all but one occurrence of a class name from the input corpus. In other words, the user-provided class name only appears once in the corpus. Interestingly, the performance of X-Class only drops less than 1%, still outperforming all compared methods. In contrast, the most recent work, LOTClass (Meng et al., 2020b), requires a wide variety of contexts of class names from the input corpus to ensure the quality of generated class vocabulary in its very first step.

5 X-Class for Hierarchical Classification

There are two straightforward ways to extend X-Class for hierarchical classification (1) **X-Class-End**: We can give all fine-grained class names as input to X-Class and conduct classification in an end-to-end manner; and (2) **X-Class-Hier**: We can first give only coarse-grained class names to X-Class and obtain coarse-grained predictions. Then, for each coarse-grained class and its predicted documents, we further create a new X-Class classifier based on the fine-grained class names.

Table 5: Micro-/Macro-F_1 scores for Fine-grained Classification on NYT-Small. All compared methods use 3 keywords per class. LOTClass failed to discover documents with category indicative terms, thus not reported. § refers to numbers coming from other papers.

Model	**Coarse (5 classes)**	**Fine (26 classes)**
WeSTClass	91/84§	50/36§
WeSHClass		87.4/63.2§
ConWea	95.23/90.79	91/79§
X-Class-End	96.67/92.98	86.07/75.30
X-Class-Hier		92.66/80.92

We experiment with hierarchical classification on the NYT-Small dataset, which has annotations for 26 fine-grained classes. We also introduce **WeSHClass** (Meng et al., 2019), the hierarchical version of WeSTClass, for comparison. LOTClass is not investigated here due to its poor coarse-grained performance on this dataset. The results in Table 5 show that X-Class-Hier performs the best, and it is a better solution than X-Class-End. We conjecture that this is because the fine-grained classes' similarities are drastically different (a pair of fine-grained classes can much similar than another pair). Overall, we show that we can apply our method to a hierarchy of classes.

6 Related Work

We discuss related work from two angles.

Weakly supervised text classification. Weakly supervised text classification has attracted much attention from researchers (Tao et al., 2018; Meng et al., 2020a; Mekala and Shang, 2020; Meng et al., 2020b). The general pipeline is to generate a set of document-class pairs to train a supervised model above them. Most previous work utilizes keywords

to find such pseudo data for training, which requires an expert that understands the corpus well. In this paper, we show that it is possible to reach a similar, and often better, performance on various datasets without such guidance from experts.

A recent work (Meng et al., 2020b) also studied the same topic — extremely weak supervision on text classification. It follows a similar idea of (Meng et al., 2020a) and further utilizes BERT to query replacements for class names to find keywords for classes, identifying potential classes for documents via string matching. Compared with LoTClass, our X-Class has a less strict requirement of class names being existent in the corpus, and can work well even when there is only one occurrence (refer to Section 4.7).

BERT for topic clustering. Aharoni and Goldberg (2020) showed that document representations obtained by an average of token representations from BERT preserve domain information well. We borrow this idea to improve our document representations through clustering. Our work differs from theirs in that our document representations are guided by the given class names.

7 Conclusions and Future Work

We propose our method X-Class for extremely weak supervision on text classification, which is to classify text with only class names as supervision. X-Class leverages BERT representations to generate class-oriented document presentations, which we then cluster to form document-class pairs, and in the end, fed to a supervised model to train on. We further set up benchmark datasets for this task that covers different data (news and reviews) and various class types (topics, locations, and sentiments). Through extensive experiments, we show the strong performance and stability of our method.

There are two directions that are possible to explore. First, focusing on the extremely weak supervision setting, we can extend to many other natural language tasks to eliminate human effort, such as Named Entity Recognition and Entity Linking. Second, based on the results on extremely weak supervision, we can expect an unsupervised version of text classification, where machines suggest class names and classify documents automatically.

8 Acknowledgements

We thank all reviewers for their constructive comments; Yu Meng for valuable discussions and comments. Our work is supported in part by NSF Convergence Accelerator under award OIA-2040727. Any opinions, findings, and conclusions or recommendations expressed herein are those of the authors and should not be interpreted as necessarily representing the views, either expressed or implied, of the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for government purposes notwithstanding any copyright annotation hereon.

9 Ethical Considerations

We do not anticipate any significant ethical concerns; Text Classification is a fundamental problem in Natural Language Processing. The intended use of this work would be to classify documents, such as news articles, efficiently. A minor consideration is the potential for certain types of hidden biases to be introduced into our results, such as a biased selection of class names or language model pre-trained on biased data. We did not observe this kind of issue in our experiments, and indeed these considerations seem low-risk for the specific datasets studied here.

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, ACL 2020, Online, July 5-10, 2020*, pages 7747–7763. Association for Computational Linguistics.

Dana Angluin and Philip D. Laird. 1987. Learning from noisy examples. *Mach. Learn.*, 2(4):343–370.

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 and Peter E. Hart. 1973. *Pattern classification and scene analysis*. A Wiley-Interscience publication. Wiley.

Kawin Ethayarajh. 2019. How contextual are contextualized word representations? comparing the geometry of bert, elmo, and GPT-2 embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural*

Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019, pages 55–65. Association for Computational Linguistics.

Jacob Goldberger and Ehud Ben-Reuven. 2017. Training deep neural-networks using a noise adaptation layer. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Hwiyeol Jo and Ceyda Cinarel. 2019. Delta-training: Simple semi-supervised text classification using pretrained word embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 3456–3461. Association for Computational Linguistics.

Ken Lang. 1995. Newsweeder: Learning to filter netnews. In *Machine Learning, Proceedings of the Twelfth International Conference on Machine Learning, Tahoe City, California, USA, July 9-12, 1995*, pages 331–339. Morgan Kaufmann.

Stuart P. Lloyd. 1982. Least squares quantization in PCM. *IEEE Trans. Inf. Theory*, 28(2):129–136.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 1412–1421. The Association for Computational Linguistics.

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, ACL 2020, Online, July 5-10, 2020*, pages 323–333. 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*. Association for Computational Linguistics.

Yu Meng, Jiaxin Huang, Guangyuan Wang, Zihan Wang, Chao Zhang, Yu Zhang, and Jiawei Han. 2020a. Discriminative topic mining via category-name guided text embedding. In *WWW '20: The Web Conference 2020, Taipei, Taiwan, April 20-24, 2020*, pages 2121–2132. ACM / IW3C2.

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, Jiaming Shen, Chao Zhang, and Jiawei Han. 2019. Weakly-supervised hierarchical text classification. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 6826–6833. AAAI Press.

Yu Meng, Yunyi Zhang, Jiaxin Huang, Chenyan Xiong, Heng Ji, Chao Zhang, and Jiawei Han. 2020b. 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.

David M. W. Powers. 1998. Applications and explanations of zipf's law. In *Proceedings of the Joint Conference on New Methods in Language Processing and Computational Natural Language Learning, NeMLaP/CoNLL 1998, Macquarie University, Sydney, NSW, Australia, January 11-17, 1998*, pages 151–160. ACL.

Paulo E. Rauber, Alexandre X. Falcão, and Alexandru C. Telea. 2016. Visualizing time-dependent data using dynamic t-sne. In *18th Eurographics Conference on Visualization, EuroVis 2016 - Short Papers, Groningen, The Netherlands, June 6-10, 2016*, pages 73–77. Eurographics Association.

Fangbo Tao, Chao Zhang, Xiusi Chen, Meng Jiang, Tim Hanratty, Lance M. Kaplan, and Jiawei Han. 2018. Doc2cube: Allocating documents to text cube without labeled data. In *IEEE International Conference on Data Mining, ICDM 2018, Singapore, November 17-20, 2018*, pages 1260–1265. IEEE Computer Society.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *CoRR*, abs/1910.03771.

Wenpeng Yin, Jamaal Hay, and Dan Roth. 2019. Benchmarking zero-shot text classification: Datasets, evaluation and entailment approach. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 3912–3921. Association for Computational Linguistics.

Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural*

Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada, pages 649–657.