Contextualized Weak Supervision for Text Classification

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

{dmekala, jshang}@ucsd.edu

Abstract

Weakly supervised text classification based on a few user-provided seed words has recently attracted much attention from researchers. Existing methods mainly generate pseudo-labels in a context-free manner (e.g., string matching), therefore, the ambiguous, context-dependent nature of human language has been long overlooked. In this paper, we propose a novel framework ConWea, providing contextualized weak supervision for text classification. Specifically, we leverage contextualized representations of word occurrences and seed word information to automatically differentiate multiple interpretations of the same word, and thus create a contextualized corpus. This contextualized corpus is further utilized to train the classifier and expand seed words in an iterative manner. This process not only adds new contextualized, highly label-indicative keywords but also disambiguates initial seed words, making our weak supervision fully contextualized. Extensive experiments and case studies on real-world datasets demonstrate the necessity and significant advantages of using contextualized weak supervision, especially when the class labels are fine-grained.

1 Introduction

Weak supervision in text classification has recently attracted much attention from researchers, because it alleviates the burden of human experts on annotating massive documents, especially in specific domains. One of the popular forms of weak supervision is a small set of user-provided seed words for each class. Typical seed-driven methods follow an iterative framework — generate pseudo-labels using some heuristics, learn the mapping between documents and classes, and expand the seed set (Agichtein and Gravano, 2000; Riloff et al., 2003; Kuipers et al., 2006; Tao et al., 2015; Meng et al., 2018).

Most of, if not all, existing methods generate pseudo-labels in a context-free manner, therefore, the ambiguous, context-dependent nature of human languages has been long overlooked. Suppose the user gives "penalty" as a seed word for the *sports* class, as shown in Figure 1. The word "penalty" has at least two different meanings: the penalty in *sports*-related documents and the fine or death penalty in *law*-related documents. If the pseudo-label of a document is decided based only on the frequency of seed words, some documents about *law* may be mislabelled as *sports*. More importantly, such errors will further introduce wrong seed words, thus being propagated and amplified over the iterations.

In this paper, we introduce contextualized weak supervision to train a text classifier based on user-provided seed words. The "contextualized" here is reflected in two places: the corpus and seed words. Every word occurrence in the corpus may be interpreted differently according to its context; Every seed word, if ambiguous, must be resolved according to its user-specified class. In this way, we aim to improve the accuracy of the final text classifier.

We propose a novel framework ConWea, as illustrated in Figure 1. It leverages contextualized representation learning techniques, such as ELMo (Peters et al., 2018) and BERT (Devlin et al., 2019), together with user-provided seed information to first create a *contextualized corpus*. This contextualized corpus is further utilized to train the classifier and expand seed words in an iterative manner. During this process, *contextualized seed words* are introduced by expanding and disambiguating the initial seed words. Specifically, for each word, we develop an unsupervised method to adaptively decide its number of interpretations, and accordingly, group all its occurrences based on their contextualized representations. We design a principled comparative ranking method to select highly label-

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 323–333
July 5 - 10, 2020. ©2020 Association for Computational Linguistics

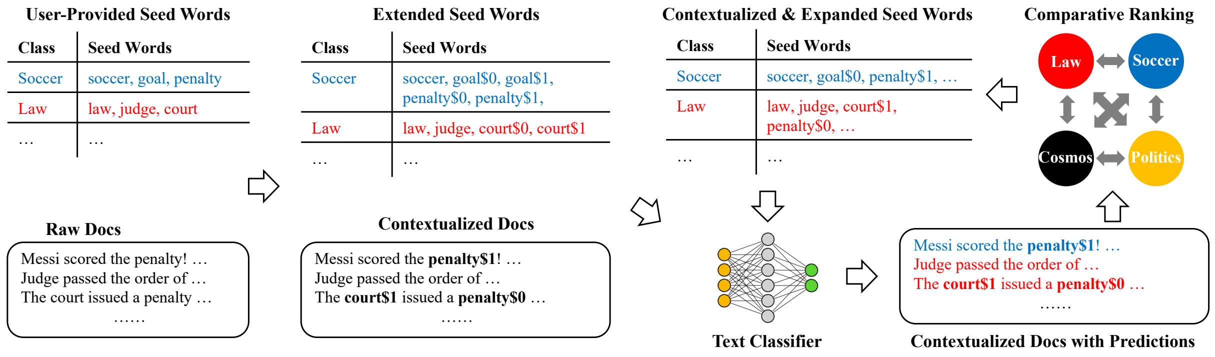

User-Provided Seed Words
Class | Seed Words
Soccer | soccer, goal, penalty
Law | law, judge, court
… | …
Extended Seed Words
Class | Seed Words
Soccer | soccer, goal$0, goal$1, penalty$0, penalty$1,
Law | law, judge, court$0, court$1
… | …
Contextualized & Expanded Seed Words
Class | Seed Words
Soccer | soccer, goal$0, penalty$1, …
Law | law, judge, court$1, penalty$0, …
… | …
Comparative Ranking
Law
Soccer
Cosmos
Politics
Raw Docs
Messi scored the penalty! …
Judge passed the order of …
The court issued a penalty …
……
Contextualized Docs
Messi scored the penalty$1! …
Judge passed the order of …
The court$1 issued a penalty$0 …
……
Text Classifier
Messi scored the penalty$1! …
Judge passed the order of …
The court$1 issued a penalty$0 …
……
Contextualized Docs with Predictions

Figure 1: Our proposed contextualized weakly supervised method leverages BERT to create a contextualized corpus. This contextualized corpus is further utilized to resolve interpretations of seed words, generate pseudo-labels, train a classifier and expand the seed set in an iterative fashion.

indicative keywords from the contextualized corpus, leading to contextualized seed words. We will repeat the iterative classification and seed word expansion process until the convergence.

To the best of our knowledge, this is the first work on contextualized weak supervision for text classification. It is also worth mentioning that our proposed framework is compatible with almost any contextualized representation learning models and text classification models. Our contributions are summarized as follows:

- We propose a novel framework enabling contextualized weak supervision for text classification.
- We develop an unsupervised method to automatically group word occurrences of the same word into an adaptive number of interpretations based on contextualized representations and user-provided seed information.
- We design a principled ranking mechanism to identify words that are discriminative and highly label-indicative.
- We have performed experiments on real-world datasets for both coarse- and fine-grained text classification tasks. The results demonstrate the superiority of using contextualized weak supervision, especially when the labels are fine-grained.

Our code is made publicly available at GitHub[1].

2 Overview

Problem Formulation. The input of our problem contains (1) a collection of n text documents $\mathcal{D} = \{\mathcal{D}_1, \mathcal{D}_2, \dots, \mathcal{D}_n\}$ and (2) m target classes $\mathcal{C} = \{\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_m\}$ and their seed words $\mathcal{S} = \{\mathcal{S}_1, \mathcal{S}_2, \dots, \mathcal{S}_m\}$. We aim to build a high-quality document classifier from these inputs, assigning class label $\mathcal{C}_j \in \mathcal{C}$ to each document $\mathcal{D}_i \in \mathcal{D}$.

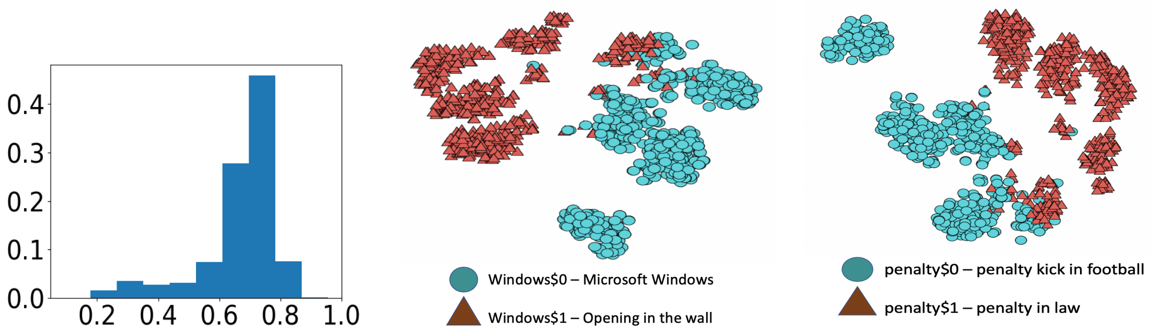
Note that, all these words could be upgraded to phrases if phrase mining techniques (Liu et al., 2015; Shang et al., 2018) were applied as preprocessing. In this paper, we stick to the words.

Framework Overview. We propose a framework, ConWea, enabling contextualized weak supervision. Here, "contextualized" is reflected in two places: the corpus and seed words. Therefore, we have developed two novel techniques accordingly to make both contextualizations happen.

First, we leverage contextualized representation learning techniques (Peters et al., 2018; Devlin et al., 2019) to create a contextualized corpus. We choose BERT (Devlin et al., 2019) as an example in our implementation to generate a contextualized vector of every word occurrence. We assume the user-provided seed words are of reasonable quality — the majority of the seed words are not ambiguous, and the majority of the occurrences of the seed words are about the semantics of the user-specified class. Based on these two assumptions, we are able to develop an unsupervised method to automatically group word occurrences of the same word into an adaptive number of interpretations, harvesting the contextualized corpus.

Second, we design a principled comparative ranking method to select highly label-indicative keywords from the contextualized corpus, leading to contextualized seed words. Specifically, we start with all possible interpretations of seed words and train a neural classifier. Based on the predictions, we compare and contrast the documents belonging to different classes, and rank contextualized words based on how label-indicative, frequent, and

[1] https://github.com/dheeraj7596/ConWea



(a) Similarity Distribution: Windows (b) Cluster Visualisation: Windows (c) Cluster Visualisation: Penalty

Figure 2: Document contextualization examples using word "windows" and "penalty". τ is decided based on the similarity distributions of all seed word occurrences. Two clusters are discovered for both words, respectively.

unusual these words are. During this process, we eliminate the wrong interpretations of initial seed words and also add more highly label-indicative contextualized words.

This entire process is visualized in Figure 1. We denote the number of iterations between classifier training and seed word expansion as T, which is the only hyper-parameter in our framework. We discuss these two novel techniques in detail in the following sections. To make our paper self-contained, we will also brief the pseudo-label generation and document classifiers.

3 Document Contextualization

We leverage contextualized representation techniques to create a contextualized corpus. The key objective of this contextualization is to disambiguate different occurrences of the same word into several interpretations. We treat every word separately, so in the rest of this section, we focus on a given word w. Specifically, given a word w, we denote all its occurrences as $w_1, \dots, w_n$, where n is its total number of occurrences in the corpus.

Contextualized Representation. First, we obtain a contextualized vector representation $\mathbf{b}_{w_i}$ for each w_i. Our proposed method is compatible with almost any contextualized representation learning model. We choose BERT (Devlin et al., 2019) as an example in our implementation to generate a contextualized vector for each word occurrence. In this contextualized vector space, we use the cosine similarity to measure the similarity between two vectors. Two word occurrences w_i and w_j of the same interpretation are expected to have a high cosine similarity between their vectors $\mathbf{b}_{w_i}$ and $\mathbf{b}_{w_j}$. For the ease of computation, we normalize all contextualized representations into unit vectors.

Choice of Clustering Methods. We model the word occurrence disambiguation problem as a clustering problem. Specifically, we propose to use the K-Means algorithm (Jain and Dubes, 1988) to cluster all contextualized representations $\mathbf{b}_{w_i}$ into K clusters, where K is the number of interpretations. We prefer K-Means because (1) the cosine similarity and Euclidean distance are equivalent for unit vectors and (2) it is fast and we are clustering a significant number of times.

Automated Parameter Setting. We choose the value of K purely based on a similarity threshold τ. τ is introduced to decide whether two clusters belong to the same interpretation by checking if the cosine similarity between two cluster center vectors is greater than τ. Intuitively, we should keep increasing K until there exist no two clusters with the same interpretation. Therefore, we choose K to be the largest number such that the similarity between any two cluster centers is no more than τ.

$$K = \arg\max_{K}\{\cos(\mathbf{c}_i, \mathbf{c}_j) < \tau \forall i, j\} \qquad (1)$$

where $\mathbf{c}_i$ refers to the i-th cluster center vector after clustering all contextualized representations into K clusters. In practice, K is usually no more than 10. So we increase K gradually until the constraint is violated.

We pick τ based on user-provided seed information instead of hand-tuning, As mentioned, we make two "majority" assumptions: (1) For any seed word, the majority of its occurrences follow the intended interpretation by the user; and (2) The majority of the seed words are not ambiguous — they only have one interpretation. Therefore, for each seed word s, we take the median of pairwise cosine similarities between its occurrences.

$$\tau(s) = \text{median}(\{\text{sim}(\mathbf{b}_{s_i}, \mathbf{b}_{s_j}) | \forall i, j\}) \qquad (2)$$

```
Algorithm 1: Corpus Contextualization
Input: Word occurrences $w_1, w_2, \dots, w_n$ of
  the word $w$, Seed words $s_1, s_2, \dots, s_m$ and
  their occurrences $s_{i,j}$.
Output: Contextualized word occurrences
  $\hat{w_1}, \hat{w_2}, \dots, \hat{w_n}$
Obtain $\mathbf{b}_{w_i}$ and $\mathbf{b}_{s_{i,j}}$ using BERT.
Compute $\tau$ follow Equation 3.
K ← 1
while True do
    Run K-Means on $\{b_{w_i}\}$ for (K+1) clusters.
    Obtain cluster centers $\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_{K+1}$.
    if $\max_{i,j} \cos(\mathbf{c}_i, \mathbf{c_j}) > \tau$ then
        Break
    K ← K + 1
Run K-Means on $\{b_{w_i}\}$ for K clusters.
Obtain cluster centers $\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_K$.
for each occurrence $w_i$ do
    Compute $\hat{w_i}$ following Equation 4.
Return $\hat{w_i}$.
```

Then, we take the median of these medians over all seed words as τ. Mathematically,

$$\tau = \text{median}(\{\tau(s) | \forall s\}) \tag{3}$$

The nested median solution makes the choice of τ safe and robust to outliers. For example, consider the word "windows" in the 20Newsgroup corpus. In fact, the word *windows* has two interpretations in the 20Newsgroup corpus — one represents an opening in the wall and the other is an operating system. We first compute the pairwise similarities between all its occurrences and plot the histogram as shown in Figure 2(a). From this plot, we can see that its median value is about 0.7. We apply the same for all seed words and obtain τ following Equation 3. τ is calculated to be 0.82. Based on this value, we gradually increase K for "windows" and it ends up with $K = 2$. We visualize its K-Means clustering results using t-SNE (Maaten and Hinton, 2008) in Figure 2(b). Similar results can be observed for the word *penalty*, as shown in Figure 2(c). These examples demonstrate how our document contextualization works for each word.

In practice, to make it more efficient, one can subsample the occurrences instead of enumerating all pairs in a brute-force manner.

Contextualized Corpus. The interpretation of each occurrence of w is decided by the cluster-ID to which its contextualized representation belongs. Specifically, given each occurrence w_i, the word w is replaced by $\hat{w_i}$ in the corpus as follows:

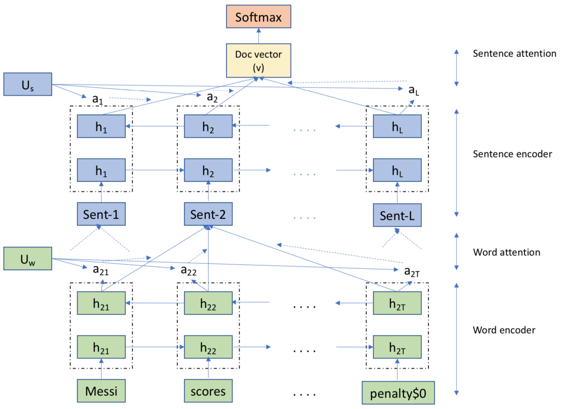


Figure 3: The HAN classifier used in our ConWea framework. It is trained on our contextualized corpus with the generated pseudo-labels.

$$\hat{w_i} = \begin{cases} w & \text{if K = 1} \\ w\$j^* & \text{otherwise} \end{cases} \tag{4}$$

where

$$j^* = \arg \max_{j=1}^{K} \cos(\mathbf{b}_{w_i}, \mathbf{c}_j)$$

By applying this to all words and their occurrences, the corpus is contextualized. The pseudo-code for corpus contextualization is shown in Algorithm 1.

4 Pseudo-Label and Text Classifier

We generate pseudo-labels for unlabeled contextualized documents and train a classifier based on these pseudo-labels, similar to many other weakly supervised methods (Agichtein and Gravano, 2000; Riloff et al., 2003; Kuipers et al., 2006; Tao et al., 2015; Meng et al., 2018). These two parts are not the focus of this paper. We briefly introduce them to make the paper self-contained.

Pseudo-Label Generation. There are several ways to generate pseudo-labels from seed words. As proof-of-concept, we employ a simple but effective method based on counting. Each document is assigned a label whose aggregated term frequency of seed words is maximum. Let tf$(\hat{w}, d)$ denote term-frequency of a contextualized word w in the contextualized document d and $\mathcal{S}_c$ represents set of seed words of class c, the document d is assigned a label $l(d)$ as follows:

$$l(d) = \arg \max_{l} \{\sum_{i} tf(s_i, d) | \forall s_i \in \mathcal{S}_l\} \tag{5}$$

Document Classifier. Our framework is compatible with any text classification model. We use Hierarchical Attention Networks (HAN) (Yang et al., 2016) as an example in our implementation. HAN considers the hierarchical structure of documents (document – sentences – words) and includes an attention mechanism that finds the most important words and sentences in a document while taking the context into consideration. There are two levels of attention: word-level attention identifies the important words in a sentence and sentence level attention identifies the important sentences in a document. The overall architecture of HAN is shown in Figure 3. We train a HAN model on contextualized corpus with the generated pseudo-labels. The predicted labels are used in seed expansion and disambiguation.

5 Seed Expansion and Disambiguation

Seed Expansion. Given contextualized documents and their predicted class labels, we propose to rank contextualized words and add the top few words into the seed word sets. The core element of this process is the ranking function. An ideal seed word s of label l, is an unusual word that appears only in the documents belonging to label l with significant frequency. Hence, for a given class $\mathcal{C}_j$ and a word w, we measure its ranking score based on the following three aspects:

- **Label-Indicative.** Since our pseudo-label generation follows the presence of seed words in the document, ideally, the posterior probability of a document belonging to the class $\mathcal{C}_j$ after observing the presence of word w (i.e., $P(\mathcal{C}_j|w)$) should be very close to 100%. Therefore, we use $P(\mathcal{C}_j|w)$ as our label-indicative measure:

$$\mathbf{LI}(\mathcal{C}_j, w) = P(\mathcal{C}_j|w) = \frac{f_{\mathcal{C}_j,w}}{f_{\mathcal{C}_j}}$$

  where $f_{\mathcal{C}_j}$ refers to the total number of documents that are predicted as class $\mathcal{C}_j$, and among them, $f_{\mathcal{C}_j,w}$ documents contain the word w. All these counts are based on the prediction results on the input unlabeled documents.

- **Frequent.** Ideally, a seed word s of label l appears in the documents belonging to label l with significant frequency. To measure the frequency score, we first compute the average frequency of seed word s in all the documents belonging to label l. Since average frequency is unbounded, we apply $tanh$ function to scale it, resulting in the frequency score,

$$\mathbf{F}(\mathcal{C}_j, w) = \tanh(\frac{f_{\mathcal{C}_j}(w)}{f_{\mathcal{C}_j}})$$

  Here, different from $f_{\mathcal{C}_j,w}$ defined earlier, $f_{\mathcal{C}_j}(w)$ is the frequency of word w in documents that are predicted as class $\mathcal{C}_j$.

- **Unusual:** We want our highly label-indicative and frequent words to be unusual. To incorporate this, we consider inverse document frequency (IDF). Let n be the number of documents in the corpus $\mathcal{D}$ and $f_{\mathcal{D},w}$ represents the document frequency of word w, the IDF of a word w is computed as follows:

$$\mathbf{IDF}(w) = \log(\frac{n}{f_{\mathcal{D},w}})$$

Similar to previous work (Tao et al., 2015), we combine these three measures using the geometric mean, resulting in the ranking score $R(\mathcal{C}_j, w)$ of a word w for a class $\mathcal{C}_j$.

$$R(\mathcal{C}_j, w) = \big(\mathbf{LI}(\mathcal{C}_j, w) \times \mathbf{F}(\mathcal{C}_j, w) \times \mathbf{IDF}(w)\big)^{1/3}$$

Based on this aggregated score, we add top words to expand the seed word set of the class $\mathcal{C}_j$.

Seed Disambiguation. While the majority of user-provided seed words are nice and clean, some of them may have multiple interpretations in the given corpus. We propose to disambiguate them based on the ranking. We first consider all possible interpretations of an initial seed word, generate the pseudo-labels, and train a classifier. Using the classified documents and the ranking function, we rank all possible interpretations of the same initial seed word. Because the majority occurrences of a seed word are assumed to belong to the user-specified class, the intended interpretation shall be ranked the highest. Therefore, we retain only the top-ranked interpretation of this seed word.

After this step, we have fully contextualized our weak supervision, including the initial user-provided seeds.

6 Experiments

In this section, we evaluate our framework and many compared methods on coarse- and fine-grained text classification tasks under the weakly supervised setting.

Table 1: Dataset statistics.

Dataset	# Docs	# Coarse	# Fine	Avg Doc Len
NYT	13,081	5	25	778
20News	18,846	6	20	400

6.1 Datasets

Following previous work (Tao et al., 2015), (Meng et al., 2018), we use two news datasets in our experiments. The dataset statistics are provided in Table 1. Here are some details.

- **The New York Times (NYT):** The NYT dataset contains news articles written and published by The New York Times. These articles are classified into 5 wide genres (e.g., arts, sports) and 25 fine-grained categories (e.g., dance, music, hockey, basketball).
- **The 20 Newsgroups (20News):** The 20News dataset[2] is a collection of newsgroup documents partitioned widely into 6 groups (e.g., recreation, computers) and 20 fine-grained classes (e.g., graphics, windows, baseball, hockey).

We perform coarse- and fine-grained classifications on the NYT and 20News datasets. NYT dataset is imbalanced in both fine-grained and coarse-grained classifications. 20News is nearly balanced in fine-grained classification but imbalanced in coarse-grained classification. Being aware of these facts, we adopt micro- and macro-F_1 scores as evaluation metrics.

6.2 Compared Methods

We compare our framework with a wide range of methods described below:

- **IR-TF-IDF** treats the seed word set for each class as a query. The relevance of a document to a label is computed by aggregated TF-IDF values of its respective seed words. The label with the highest relevance is assigned to each document.
- **Dataless** (Chang et al., 2008) uses only label surface names as supervision and leverages Wikipedia to derive vector representations of labels and documents. Each document is labeled based on the document-label similarity.
- **Word2Vec** first learns word vector representations (Mikolov et al., 2013) for all terms in the corpus and derive label representations by aggregating the vectors of its respective seed words. Finally, each document is labeled with the most similar label based on cosine similarity.
- **Doc2Cube** (Tao et al., 2015) considers label surface names as seed set and performs multi-dimensional document classification by learning dimension-aware embedding.
- **WeSTClass** (Meng et al., 2018) leverages seed information to generate pseudo documents and refines the model through a self-training module that bootstraps on real unlabeled documents.

We denote our framework as **ConWea**, which includes contextualizing corpus, disambiguating seed words, and iterative classification & key words expansion. Besides, we have three ablated versions. **ConWea-NoCon** refers to the variant of ConWea trained without the contextualization of corpus. **ConWea-NoSeedExp** is the variant of ConWea without the seed expansion module. **ConWea-WSD** refers to the variant of ConWea, with the contextualization module replaced by Lesk algorithm (Lesk, 1986), a classic Word-sense disambiguation algorithm (WSD).

We also present the results of **HAN-Supervised** under the supervised setting for reference. We use 80-10-10 for train-validation-test splitting and report the test set results for it. All weakly supervised methods are evaluated on the entire datasets.

6.3 Experiment Settings

We use pre-trained `BERT-base-uncased`[3] to obtain contextualized word representations. We follow Devlin et al. (2019) and concatenate the averaged word-piece vectors of the last four layers.

The seed words are obtained as follows: we asked 5 human experts to nominate 5 seed words per class, and then considered the majority words (i.e., > 3 nominations) as our final set of seed words. For every class, we mainly use the label surface name as seed words. For some multi-word class labels (e.g., “international business”), we have multiple seed words, but never exceeds four per each class. The same seed words are utilized for all compared methods for fair comparisons.

For ConWea, we set $T = 10$. For any method using word embedding, we set its dimension to be 100. We use the public implementations of WeSTClass[4] and Dataless[5] with the hyper-parameters mentioned in their original papers.

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

[3] https://github.com/google-research/bert

[4] https://github.com/yumeng5/WeSTClass

[5] https://cogcomp.org/page/software_view/Descartes

Table 2: Evaluation Results for All Methods on Fine-Grained and Coarse-Grained Labels. Both micro-F$_1$ and macro-F$_1$ scores are presented. Ablation and supervised results are also included.

	NYT				**20 Newsgroup**			
	5-Class (Coarse)		**25-Class** (Fine)		**6-Class** (Coarse)		**20-Class** (Fine)	
Methods	Micro-F$_1$	Macro-F$_1$	Micro-F$_1$	Macro-F$_1$	Micro-F$_1$	Macro-F$_1$	Micro-F$_1$	Macro-F$_1$
IR-TF-IDF	0.65	0.58	0.56	0.54	0.49	0.48	0.53	0.52
Dataless	0.71	0.48	0.59	0.37	0.50	0.47	0.61	0.53
Word2Vec	0.92	0.83	0.69	0.47	0.51	0.45	0.33	0.33
Doc2Cube	0.71	0.38	0.67	0.34	0.40	0.35	0.23	0.23
WeSTClass	0.91	0.84	0.50	0.36	0.53	0.43	0.49	0.46
ConWea	**0.95**	**0.89**	**0.91**	**0.79**	**0.62**	**0.57**	**0.65**	**0.64**
ConWea-NoCon	0.91	0.83	0.89	0.74	0.53	0.50	0.58	0.57
ConWea-NoExpan	0.92	0.85	0.76	0.66	0.58	0.53	0.58	0.57
ConWea-WSD	0.83	0.78	0.72	0.64	0.52	0.46	0.49	0.47
HAN-Supervised	0.96	0.92	0.94	0.82	0.90	0.88	0.83	0.83

6.4 Performance Comparison

We summarize the evaluation results of all methods in Table 2. As one can observe that our proposed framework achieves the best performance among all the compared weakly supervised methods. We discuss the effectiveness of ConWea as follows:

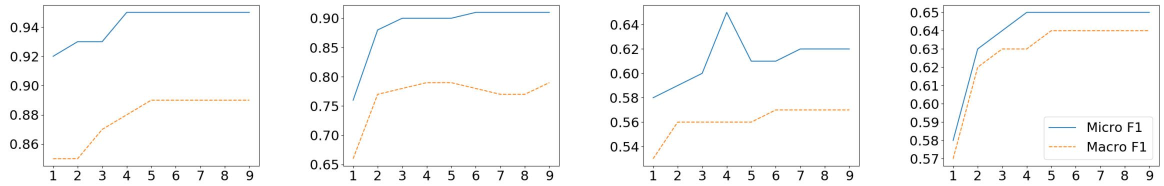
- Our proposed framework ConWea outperforms all the other methods with significant margins. By contextualizing the corpus and resolving the interpretation of seed words, ConWea achieves inspiring performance, demonstrating the necessity and the importance of using contextualized weak supervision.
- We observe that in the fine-grained classification, the advantages of ConWea over other methods are even more significant. This can be attributed to the contextualization of corpus and seed words. Once the corpus is contextualized properly, the subtle ambiguity between words is a drawback to other methods, whereas ConWea can distinguish them and predict them correctly.
- The comparison between ConWea and the ablation method ConWea-NoExpan demonstrates the effectiveness of our Seed Expansion. For example, for fine-grained labels on the 20News dataset, the seed expansion improves the micro-F1 score from 0.58 to 0.65.
- The comparison between ConWea and the two ablation methods ConWea-NoCon and ConWea-WSD demonstrates the effectiveness of our Contextualization. Our contextualization, building upon (Devlin et al., 2019), is adaptive to the input corpus, without requiring any additional human annotations. However, WSD methods(e.g., (Lesk, 1986)) are typically trained for a general domain. If one wants to apply WSD to some specific corpus, additional annotated training data might be required to meet the similar performance as ours, which defeats the purpose of a weakly supervised setting. Therefore, we believe that our contextualization module has its unique advantages. Our experimental results further confirm the above reasoning empirically. For example, for coarse-grained labels on the 20News dataset, the contextualization improves the micro-F1 score from 0.53 to 0.62.
- We observe that ConWea performs quite close to supervised methods, for example, on the NYT dataset. This demonstrates that ConWea is quite effective in closing the performance gap between the weakly supervised and supervised settings.

6.5 Parameter Study

The only hyper-parameter in our algorithm is T, the number of iterations of iterative expansion & classification. We conduct experiments to study the effect of the number of iterations on the performance. The plot of performance w.r.t. the number of iterations is shown in Figure 4. We observe that the performance increases initially and gradually converges after 4 or 5 iterations. We observe that after the convergence point, the expanded seed words have become almost unchanged. While there is some fluctuation, a reasonably large T, such as $T = 10$, is a good choice.

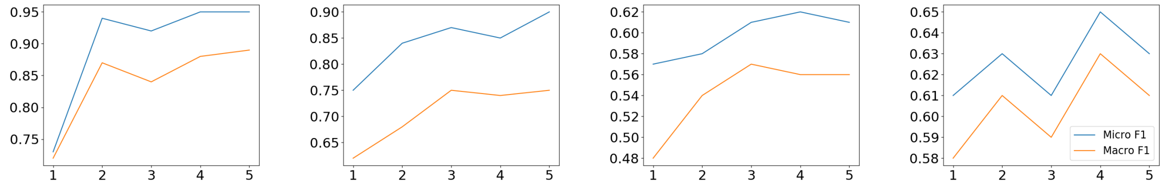
6.6 Number of Seed Words

We vary the number of seed words per class and plot the F_1 score in Figure 5. One can observe that in general, the performance increases as the number of seed words increase. There is a slightly different pattern on the 20News dataset when the labels are fine-grained. We conjecture that it is caused by the



(a) NYT Coarse (b) NYT Fine (c) 20News Coarse (d) 20News Fine

Figure 4: Micro- and Macro-F_1 scores w.r.t. the number of iterations.



(a) NYT Coarse (b) NYT Fine (c) 20News Coarse (d) 20News Fine

Figure 5: Micro- and Macro-F_1 scores w.r.t. the number of seed words.

subtlety of seed words in fine-grained cases – additional seed words may bring some noise. Overall, three seed words per class are enough for reasonable performance.

6.7 Case Study

We present a case study to showcase the power of contextualized weak supervision. Specifically, we investigate the differences between the expanded seed words in the plain corpus and contextualized corpus over iterations. Table 3 shows a column-by-column comparison for the class *For Sale* on the 20News dataset. The class *For Sale* refers to documents advertising goods for sale. Starting with the same seed sets in both types of corpora, from Table 3, in the second iteration, we observe that "space" becomes a part of expanded seed set in the plain corpus. Here "space" has two interpretations, one stands for the physical universe beyond the Earth and the other is for an area of land. This error gets propagated and amplified over the iterations, further introducing wrong seed words like "nasa", "shuttle" and "moon", related to its first interpretation. The seed set for contextualized corpus addresses this problem and adds only the words with appropriate interpretations. Also, one can see that the initial seed word "offer" has been disambiguated as "offer$0".

7 Related Work

We review the literature about (1) weak supervision for text classification methods, (2) contextualized representation learning techniques, (3) document classifiers, and (4) word sense disambiguation.

7.1 Weak Supervision for Text Classification

Weak supervision has been studied for building document classifiers in various of forms, including hundreds of labeled training documents (Tang et al., 2015; Miyato et al., 2016; Xu et al., 2017), class/category names (Song and Roth, 2014; Tao et al., 2015; Li et al., 2018), and user-provided seed words (Meng et al., 2018; Tao et al., 2015). In this paper, we focus on user-provided seed words as the source of weak supervision, Along this line, Doc2Cube (Tao et al., 2015) expands label keywords from label surface names and performs multi-dimensional document classification by learning dimension-aware embedding; PTE (Tang et al., 2015) utilizes both labeled and unlabeled documents to learn text embeddings specifically for a task, which are later fed to logistic regression classifiers for classification; Meng et al. (2018) leverage seed information to generate pseudo documents and introduces a self-training module that bootstraps on real unlabeled data for model refining. This method is later extended to handle hierarchical classifications based on a pre-defined label taxonomy (Meng et al., 2019). However, all these weak supervisions follow a context-free manner. Here, we propose to use contextualized weak supervision.

7.2 Contextualized Word Representations

Contextualized word representation is originated from machine translation (MT). CoVe (McCann et al., 2017) generates contextualized representations for a word based on pre-trained MT models, More recently, ELMo (Peters et al., 2018) leverages neural language models to replace MT models,

Table 3: Case Study: Seed word expansion of the *For Sale* class in context-free and contextualized corpora. The *For Sale* class contains documents advertising goods for sale. Blue bold words are potentially wrong seeds.

	Seed Words for *For Sale* class	
Iter	**Plain Corpus**	**Contextualized Corpus**
1	sale, offer, forsale	sale, offer, forsale
2	**space**, price, shipping, sale, offer	shipping, forsale, offer\$0, condition\$0, sale
3	**space**, price, shipping, sale, **nasa**, offer, package, email	price, shipping, sale, forsale, condition\$0, offer\$0, package, email
4	**space**, price, **moon**, shipping, sale, **nasa**, offer, **shuttle**, package, email	price, shipping, sale, forsale, condition\$0, offer\$0, package, email, offers\$0, obo\$0

which removes the dependency on massive parallel texts and takes advantages of nearly unlimited raw corpora. Many models leveraging language modeling to build sentence representations (Howard and Ruder, 2018; Radford et al., 2018; Devlin et al., 2019) emerge almost at the same time. Language models have also been extended to the character level (Liu et al., 2018; Akbik et al., 2018), which can generate contextualized representations for character spans.

Our proposed framework is compatible with all the above contextualized representation techniques. In our implementation, we choose to use BERT to demonstrate the power of using contextualized supervision.

7.3 Word Sense Disambiguation

Word Sense Disambiguation (WSD) is one of the challenging problems in natural language processing. Typical WSD models (Lesk, 1986; Zhong and Ng, 2010; Yuan et al., 2016; Raganato et al., 2017; Le et al., 2018; Tripodi and Navigli, 2019) are trained for a general domain. Recent works (Li and Jurafsky, 2015; Mekala et al., 2016; Gupta et al., 2019) also showed that machine-interpretable representations of words considering its senses, improve document classification. However, if one wants to apply WSD to some specific corpus, additional annotated training data might be required to meet the similar performance as ours, which defeats the purpose of a weakly supervised setting.

In contrast, our contextualization, building upon (Devlin et al., 2019), is adaptive to the input corpus, without requiring any additional human annotations. Therefore, our framework is more suitable than WSD under the weakly supervised setting.. Our experimental results have verified this reasoning and showed the superiority of our contextualization module over WSD in weakly supervised document classification tasks.

7.4 Document Classifier

Document classification problem has been long studied. In our implementation of the proposed ConWea framework, we used HAN (Yang et al., 2016), which considers the hierarchical structure of documents and includes attention mechanisms to find the most important words and sentences in a document. CNN-based text classifiers(Kim, 2014; Zhang et al., 2015; Lai et al., 2015) are also popular and can achieve inspiring performance.

Our framework is compatible with all the above text classifiers. We choose HAN just for a demonstration purpose.

8 Conclusions and Future Work

In this paper, we proposed ConWea, a novel contextualized weakly supervised classification framework. Our method leverages contextualized representation techniques and initial user-provided seed words to contextualize the corpus. This contextualized corpus is further used to resolve the interpretation of seed words through iterative seed word expansion and document classifier training. Experimental results demonstrate that our model outperforms previous methods significantly, thereby signifying the superiority of contextualized weak supervision, especially when labels are fine-grained.

In the future, we are interested in generalizing contextualized weak supervision to hierarchical text classification problems. Currently, we perform coarse- and fine-grained classifications separately. There should be more useful information embedded in the tree-structure of the label hierarchy. Also, extending our method for other types of textual data, such as short texts, multi-lingual data, and code-switched data is a potential direction.

Acknowledgements

We thank Palash Chauhan and Harsh Jhamtani for valuable discussions.

References

Eugene Agichtein and Luis Gravano. 2000. Snowball: Extracting relations from large plain-text collections. In *Proceedings of the fifth ACM conference on Digital libraries*, pages 85–94. ACM.

Alan Akbik, Duncan Blythe, and Roland Vollgraf. 2018. Contextual string embeddings for sequence labeling. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1638–1649.

Ming-Wei Chang, Lev-Arie Ratinov, Dan Roth, and Vivek Srikumar. 2008. Importance of semantic representation: Dataless classification. In *Aaai*, volume 2, pages 830–835.

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, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Vivek Gupta, Ankit Saw, Pegah Nokhiz, Harshit Gupta, and Partha Talukdar. 2019. Improving document classification with multi-sense embeddings. *arXiv preprint arXiv:1911.07918*.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 328–339.

Anil K Jain and Richard C Dubes. 1988. Algorithms for clustering data. *Englewood Cliffs: Prentice Hall, 1988*.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.

Benjamin J Kuipers, Patrick Beeson, Joseph Modayil, and Jefferson Provost. 2006. Bootstrap learning of foundational representations. *Connection Science*, 18(2):145–158.

Siwei Lai, Liheng Xu, Kang Liu, and Jun Zhao. 2015. Recurrent convolutional neural networks for text classification. In *Twenty-ninth AAAI conference on artificial intelligence*.

Minh Le, Marten Postma, Jacopo Urbani, and Piek Vossen. 2018. A deep dive into word sense disambiguation with lstm. In *Proceedings of the 27th international conference on computational linguistics*, pages 354–365.

Michael Lesk. 1986. Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. In *Proceedings of the 5th annual international conference on Systems documentation*, pages 24–26.

Jiwei Li and Dan Jurafsky. 2015. Do multi-sense embeddings improve natural language understanding? *arXiv preprint arXiv:1506.01070*.

Keqian Li, Hanwen Zha, Yu Su, and Xifeng Yan. 2018. Unsupervised neural categorization for scientific publications. In *SIAM Data Mining*, pages 37–45. SIAM.

Jialu Liu, Jingbo Shang, Chi Wang, Xiang Ren, and Jiawei Han. 2015. Mining quality phrases from massive text corpora. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, pages 1729–1744.

Liyuan Liu, Jingbo Shang, Xiang Ren, Frank Fangzheng Xu, Huan Gui, Jian Peng, and Jiawei Han. 2018. Empower sequence labeling with task-aware neural language model. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605.

Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. 2017. Learned in translation: Contextualized word vectors. In *Advances in Neural Information Processing Systems*, pages 6294–6305.

Dheeraj Mekala, Vivek Gupta, Bhargavi Paranjape, and Harish Karnick. 2016. Scdv: Sparse composite document vectors using soft clustering over distributional representations. *arXiv preprint arXiv:1612.06778*.

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*, pages 983–992. ACM.

Yu Meng, Jiaming Shen, Chao Zhang, and Jiawei Han. 2019. Weakly-supervised hierarchical text classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6826–6833.

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

Takeru Miyato, Andrew M Dai, and Ian Goodfellow. 2016. Adversarial training methods for semi-supervised text classification. *arXiv:1605.07725*.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of NAACL-HLT*, pages 2227–2237.

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

Alessandro Raganato, Jose Camacho-Collados, and Roberto Navigli. 2017. Word sense disambiguation: A unified evaluation framework and empirical comparison. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 99–110.

Ellen Riloff, Janyce Wiebe, and Theresa Wilson. 2003. Learning subjective nouns using extraction pattern bootstrapping. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*, pages 25–32. Association for Computational Linguistics.

Jingbo Shang, Jialu Liu, Meng Jiang, Xiang Ren, Clare R Voss, and Jiawei Han. 2018. Automated phrase mining from massive text corpora. *IEEE Transactions on Knowledge and Data Engineering*, 30(10):1825–1837.

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

Jian Tang, Meng Qu, and Qiaozhu Mei. 2015. Pte: Predictive text embedding through large-scale heterogeneous text networks. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1165–1174. ACM.

Fangbo Tao, Chao Zhang, Xiusi Chen, Meng Jiang, Tim Hanratty, Lance Kaplan, and Jiawei Han. 2015. Doc2cube: Automated document allocation to text cube via dimension-aware joint embedding. *Dimension*, 2016:2017.

Rocco Tripodi and Roberto Navigli. 2019. Game theory meets embeddings: a unified framework for word sense disambiguation. 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)*, pages 88–99.

Weidi Xu, Haoze Sun, Chao Deng, and Ying Tan. 2017. Variational autoencoder for semi-supervised text classification. In *AAAI*.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 1480–1489.

Dayu Yuan, Julian Richardson, Ryan Doherty, Colin Evans, and Eric Altendorf. 2016. Semi-supervised word sense disambiguation with neural models. *arXiv preprint arXiv:1603.07012*.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in neural information processing systems*, pages 649–657.

Zhi Zhong and Hwee Tou Ng. 2010. It makes sense: A wide-coverage word sense disambiguation system for free text. In *Proceedings of the ACL 2010 system demonstrations*, pages 78–83.