Title

Contextualized, Metadata-Empowered, Coarse-to-Fine Weakly-Supervised Text Classification

Author

Mekala, Dheeraj

Publication Date

2021

UNIVERSITY OF CALIFORNIA SAN DIEGO

Contextualized, Metadata-Empowered, Coarse-to-Fine Weakly-Supervised Text Classification

A Thesis submitted in partial satisfaction of the
requirements for the degree of Master of Science

in

Computer Science

by

Dheeraj Mekala

Committee in charge:

Professor Jingbo Shang, Chair
Professor Julian John Mcauley
Professor Ndapandula Nakashole

2021

The Thesis of Dheeraj Mekala is approved and is acceptable in quality and form for publication on microfilm and electronically.

University of California San Diego

2021

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

I would like to acknowledge Professor Jingbo Shang for his support as the chair of my committee. His expertise in the domain and insightful discussions about the research were instrumental in completing this thesis. And also, I am immensely thankful to him for his support, flexibility and kindness particularly during the stressful times of the pandemic. I am deeply grateful to him for providing an enriching environment for the past two years which has helped me gain valuable research experience.

I would like to thank my parents, grand parents, and uncle for their incredible support throughout my life and always motivating me to aim high. Aditionally, I would like to express my gratitude to Professor Harish Karnick (CSE, IIT Kanpur) who kindled my interest in the fields of Machine learning and Natural Language Processing.

I would also like to thank Harsh, Varun, Chengyu, Zihan for the inspiring discussions that greatly polished this thesis work. Finally, I am grateful to my friends Sanjana, Palash, Winny, Amulya, Shreyas, Akshaya, Gautham, and Joseph for their friendship and emotional support for these two years, specifically during the difficult times of the pandemic.

Chapter 2, in full, is a reprint of the material as it appears in Mekala, Dheeraj; Shang, Jingbo. “Contextualized weak supervision for text classification,” in Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 323–333, 2020. The dissertation/thesis author was the primary investigator and author of this paper.

Chapter 3, in full, is a reprint of the material as it appears in Mekala, Dheeraj; Zhang, Xinyang; Shang, Jingbo. “Meta: Metadata-empowered weak supervision for text classification,” in Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 8351–8361, 2020. The dissertation/thesis author was the primary investigator and author of this paper.

Chapter 4, in part is currently being prepared for submission for publication of the material. Mekala, Dheeraj; Gangal, Varun; Shang, Jingbo. The dissertation/thesis author was the primary investigator and author of this material.

To my mom & dad for their love and support.

ABSTRACT OF THE THESIS

Contextualized, Metadata-Empowered, Coarse-to-Fine Weakly-Supervised Text Classification

by

Dheeraj Mekala

Master of Science in Computer Science

University of California San Diego, 2021

Professor Jingbo Shang, Chair

Text classification plays a fundamental role in transforming unstructured text data to structured knowledge. State-of-the-art text classification techniques rely on heavy domain-specific annotations to build massive machine(deep) learning models. Although these deep learning models exhibit superior performance, the lack of training data and expensive human effort in the manual annotation is a key bottleneck that forbids them from being adopted in many practical scenarios. To address this bottleneck, our research exploits the data and develops a family of data-driven text classification frameworks with minimal supervision, for e.g. class names, a few label-indicative seed words per class.

The massive volume of text data and complexity of natural language pose significant challenges to categorizing the text corpus without human annotations. For instance, the user-provided seed words can have multiple interpretations depending on the context, and their respective user-intended interpretation has to be identified for accurate classification. Moreover, metadata information like author, year, and location is widely available in addition to the text data, and it could serve as a strong, complementary source of supervision. However, leveraging metadata is challenging because (1) metadata is multi-typed, therefore it requires systematic modeling of different types and their combinations, (2) metadata is noisy, some metadata entities (e.g., authors, venues) are more compelling label indicators than others. And also, the label set is typically assumed to be fixed in traditional text classification problems. However, in many real-world applications, new classes especially more fine-grained ones will be introduced as the data volume increases. The goal of our research is to create general data-driven methods that transform real-world text data into structured categories of human knowledge with minimal human effort.

This thesis outlines a family of weakly supervised text classification approaches, which upon combining can automatically categorize huge text corpus into coarse and fine-grained classes, with just label hierarchy and a few label-indicative seed words as supervision. Specifically, it first leverages contextualized representations of word occurrences and seed word information to automatically differentiate multiple interpretations of a seed word, and thus resulting in contextualized weak supervision. Then, to leverage metadata, it organizes the text data and metadata together into a text-rich network and adopt network motifs to capture appropriate combinations of metadata. Finally, we introduce a new problem called coarse-to-fine grained classification, which aims to perform fine-grained classification on coarsely annotated data. Instead of asking for new fine-grained human annotations, we opt to leverage label surface names as the only human guidance and weave in rich pre-trained generative language models into the iterative weak supervision strategy. We have performed extensive experiments on real-world datasets from different domains. The results demonstrate significant advantages of using contextualized weak supervision and leveraging metadata, and superior performance over baselines.

Chapter 1

Introduction

Text classification is the activity of labeling natural language texts with relevant categories. It plays a fundamental role in transforming unstructured text data to structured knowledge. Classifying massive text articles into categories help users to easily search and navigate within categories. Over the last few years, it has been increasingly clear that deep learning models exhibit superior performance in this task. Despite their attractiveness and increasing popularity, the lack of training data and expensive human effort in the manual annotation is a key bottleneck that forbids them from being adopted in many practical scenarios.

To address this bottleneck, weak supervision in text classification has recently attracted much attention from researchers. One of the popular forms of weak supervision is a small set of user-provided label-indicative seed words for each class. There have been few studies in this direction. For instance, Doc2Cube [1] expands label keywords from label surface names and performs multi-dimensional document classification by learning dimension-aware embedding; WeSTClass [2] leverages seed information to generate pseudo documents and introduces a self-training module that bootstraps on real unlabeled data for model refining.

The massive volume of text data and complexity of natural language pose significant challenges to categorizing the text corpus without human annotations. For instance, the user-provided seed words can have multiple interpretations depending on the context, and their respective user-intended interpretation has to be identified for accurate classification. Existing methods mainly generate pseudo-labels in a context-free manner (e.g., string matching), and therefore, the ambiguous, context-dependent nature of human language has been long overlooked. Moreover, metadata information like author, year, and location is widely available in addition to the text data, and it could serve as a strong, complementary source of supervision. However, leveraging metadata is challenging because (1) metadata is multi-typed, therefore it requires systematic modeling of different types and their combinations, (2) metadata is noisy, some metadata entities (e.g., authors, venues) are more compelling label indicators than others. And also, the label set is typically assumed to be fixed in traditional text classification problems. However, in many real-world applications, new classes especially more fine-grained ones will be introduced as the data volume increases. One commonly used method is to extend the existing label set to a label hierarchy by expanding every original coarse-grained class into a few new, fine-grained ones, and then assign a fine-grained label to each document. The goal of our research is to create general data-driven methods that can be adopted in practical scenarios and transform real-world text data into structured categories of human knowledge with minimal human effort.

1.1 Overview and Contributions

This thesis outlines a family of weakly supervised text classification approaches, which upon combining can automatically categorize huge text corpus into coarse and fine-grained classes, with just label hierarchy and a few label-indicative seed words as supervision. Specifically, it first leverages contextualized representations of word occurrences and seed word information to automatically differentiate multiple interpretations of a word, and thus resulting in contextualized weak supervision. 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. Then, to leverage metadata, we organize the text and metadata together into a text-rich network and adopt network motifs to capture appropriate combinations of metadata. The network structure gives a holistic view of the corpus and enables to rank and select useful metadata entities. Finally, we introduce a new problem called coarse-to-fine grained classification, which aims to perform fine-grained classification on coarsely annotated data without any fine-grained human annotations. Instead of asking for new fine-grained human annotations, we opt to leverage label surface names as the only human guidance and weave in rich pre-trained generative language models into the iterative weak supervision strategy.

In summary, this thesis is a series of data-driven weakly supervised text classification frameworks, that take advantage of user-provided seed words, pre-trained language models, and signals from data and thus require no additional human curation or annotation. We will introduce how to contextualize weak supervision for text classification in Chapter 2, leverage metadata in Chapter 3, and our proposed coarse-to-fine grained classification framework in Chapter 4. Finally, we conclude the thesis by proposing potential future work in Chapter 5.

1.2 Open source tools

Our developed methods are made public and has attracted attention from the open-source community. The list of our open-sourced methods are as follows:

- ConWea: `https://github.com/dheeraj7596/ConWea`
- META: `https://github.com/dheeraj7596/META`
- C2F: `https://github.com/dheeraj7596/C2F`

1.3 Overall Impact

Our methods on contextualized weak supervision for text classification (ConWea [3]) and metadata-empowered weak supervision for text classification (META [4]) are being taught in graduate courses, e.g. *University of California San Diego* (CSE291-Advanced Data-Driven Text Mining).

Chapter 2

Contextualized Weak Supervision for Text Classification

In this chapter, we present our proposed method ConWea, that contextualizes weak supervision thereby resolving the interpretation of seed words and performs text classification.

2.1 Motivation & Overview

Weak supervision in text classification has recently attracted much attention from researchers, because it alleviates the burden of human experts on annotating massive documents. 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 [5, 6, 7, 1, 2].

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 2.1. 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,

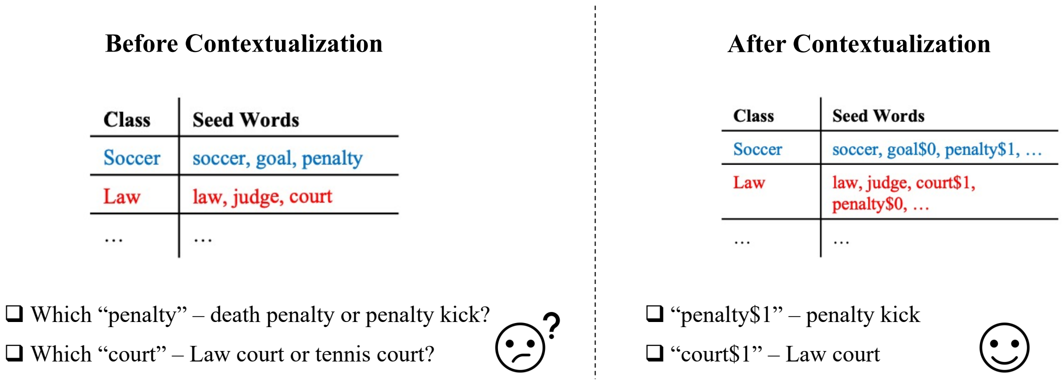

Before Contextualization
Class
Seed Words
Soccer
soccer, goal, penalty
Law
law, judge, court
…
…
❑ Which "penalty" – death penalty or penalty kick?
❑ Which "court" – Law court or tennis court?
?
After Contextualization
Class
Seed Words
Soccer
soccer, goal$0, penalty$1, …
Law
law, judge, court$1, penalty$0, …
…
…
❑ "penalty$1" – penalty kick
❑ "court$1" – Law court

Figure 2.1. Why contextualization?

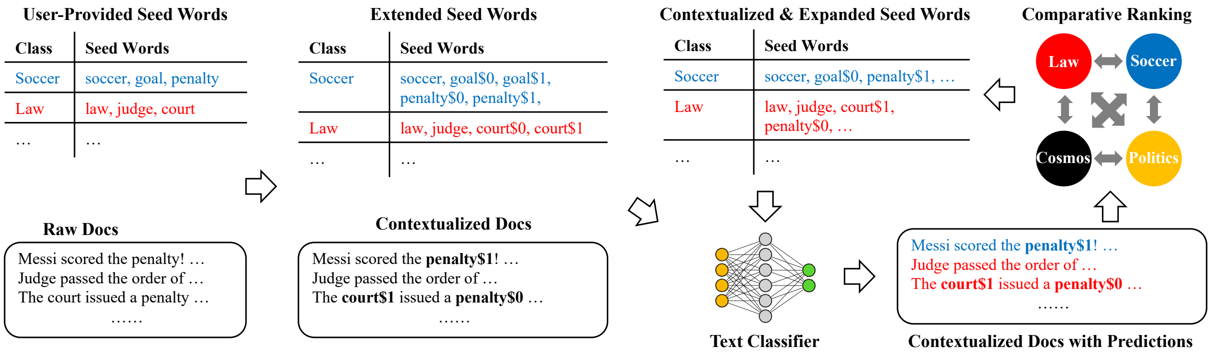

User-Provided Seed Words
Class
Seed Words
Soccer
soccer, goal, penalty
Law
law, judge, court
…
…
Raw Docs
Messi scored the penalty! …
Judge passed the order of …
The court issued a penalty …
……
Extended Seed Words
Class
Seed Words
Soccer
soccer, goal$0, goal$1, penalty$0, penalty$1,
Law
law, judge, court$0,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

Figure 2.2. 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.

thus being propagated and amplified over the iterations.

Bearing these challenges in mind, we propose ConWea, a **Con**textualized **Wea**kly supervised text classification framework. This framework introduces 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 performance of the final text classifier.

As illustrated in Figure 2.2, it leverages contextualized representation learning techniques, such as ELMo [8] and BERT [9], 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-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.

2.2 Related Work

In this section, 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.

2.2.1 Weak Supervision for Text Classification

Weak supervision has been studied for building document classifiers in various of forms, including hundreds of labeled training documents [10, 11, 12], class/category names [13, 1, 14], and user-provided seed words [2, 1]. Our method focuses on user-provided seed words as the source of weak supervision, Along this line, Doc2Cube [1] expands label keywords from label surface names and performs multi-dimensional document classification by learning dimension-aware embedding; PTE [10] utilizes both labeled and unlabeled documents to learn text embeddings specifically for a task, which are later fed to logistic regression classifiers for classification; WeSTClass [2] leverages 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 [15]. However, all these weak supervisions follow a context-free manner. Here, we propose to use contextualized weak supervision.

2.2.2 Contextualized Word Representations

Contextualized word representation is originated from machine translation (MT). CoVe [16] generates contextualized representations for a word based on pre-trained MT models, More recently, ELMo [8] leverages neural language models to replace MT models, 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 [17, 18, 9] emerge almost at the same time. Language models have also been extended to the character level [19, 20], 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.

2.2.3 Word Sense Disambiguation

Word Sense Disambiguation (WSD) is one of the challenging problems in natural language processing. Typical WSD models [21, 22, 23, 24, 25, 26] are trained for a general domain. Recent works [27, 28, 29] 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 [9], 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.

2.2.4 Document Classifier

Document classification problem has been long studied. In our implementation of the proposed framework, we used HAN [30], 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[31, 32, 33] 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.

2.3 Preliminaries

In this section, we introduce seed-driven weakly supervised text classification problem and provide an overview of our proposed framework.

2.3.1 Problem Formulation

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

Note that, all these words could be upgraded to phrases if phrase mining techniques [34, 35] were applied as pre-processing. In this chapter, we stick to the words.

2.3.2 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 [8, 9] to create a contextualized corpus. We choose BERT [9] 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 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 2.2. 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 this chapter self-contained, we will also brief the pseudo-label generation and document classifiers.

2.4 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 $w_1, \ldots, w_n$, where n is its total number of occurrences in the corpus.

2.4.1 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 [9] 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.

2.4.2 Choice of Clustering Methods

We model the word occurrence disambiguation problem as a clustering problem. Specifically, we propose to use the K-Means algorithm [36] 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.

2.4.3 Automated Parameter Setting

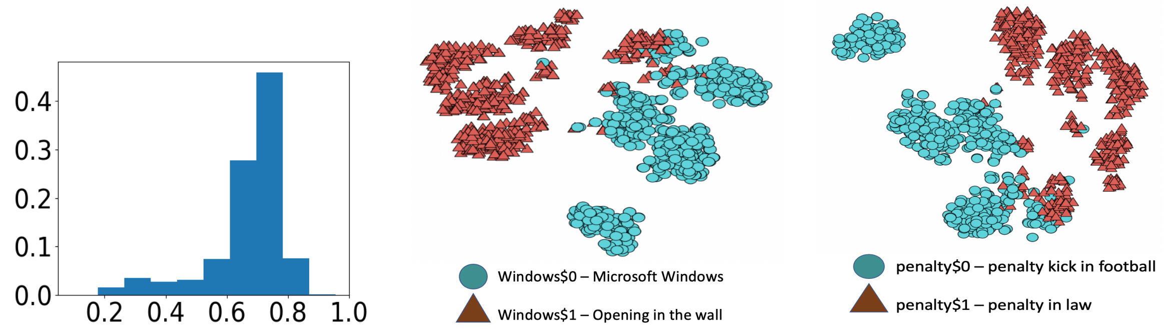
We choose the value of K purely based on a similarity threshold τ. τ is introduced to decide whether two clusters belong to the same interpretation by checking if the cosine similarity between two cluster center vectors is greater than τ. Intuitively, we should keep increasing K until there exist no 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.

$$K = \arg\max_K \{\cos(\mathbf{c}_i, \mathbf{c}_j) < \tau \; \forall \, i, j\} \tag{2.1}$$

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.

$$\tau(s) = \text{median}(\{\text{sim}(\mathbf{b}_{s_i}, \mathbf{b}_{s_j}) \mid \forall \, i, j\}) \tag{2.2}$$



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

Figure 2.3. 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.

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

$$\tau = \text{median}(\{\tau(s)|\forall s\}) \tag{2.3}$$

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 2.3(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 2.3. 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 [37] in Figure 2.3(b). Similar results can be observed for the word *penalty*, as shown in Figure 2.3(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.

Algorithm 1: Corpus Contextualization

```
Input: Word occurrences $w_1, w_2, \ldots, w_n$ of the word $w$, Seed words $s_1, s_2, \ldots, s_m$ and
  their occurrences $s_{i,j}$.
Output: Contextualized word occurrences $\hat{w}_1, \hat{w}_2, \ldots, \hat{w}_n$
Obtain $\mathbf{b}_{w_i}$ and $\mathbf{b}_{s_{i,j}}$ using BERT.
Compute $\tau$ follow Equation 2.3.
K ← 1
while True do
    Run K-Means on $\{b_{w_i}\}$ for (K+1) clusters.
    Obtain cluster centers $\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_{K+1}$.
    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 $\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_K$.
for each occurrence $w_i$ do
    Compute $\hat{w}_i$ following Equation 2.4.
Return $\hat{w}_i$.
```

2.4.4 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:

$$\hat{w}_i = \begin{cases} w & \text{if K = 1} \\ w\$j^* & \text{otherwise} \end{cases} \tag{2.4}$$

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 pseudocode for corpus contextualization is shown in Algorithm 1.

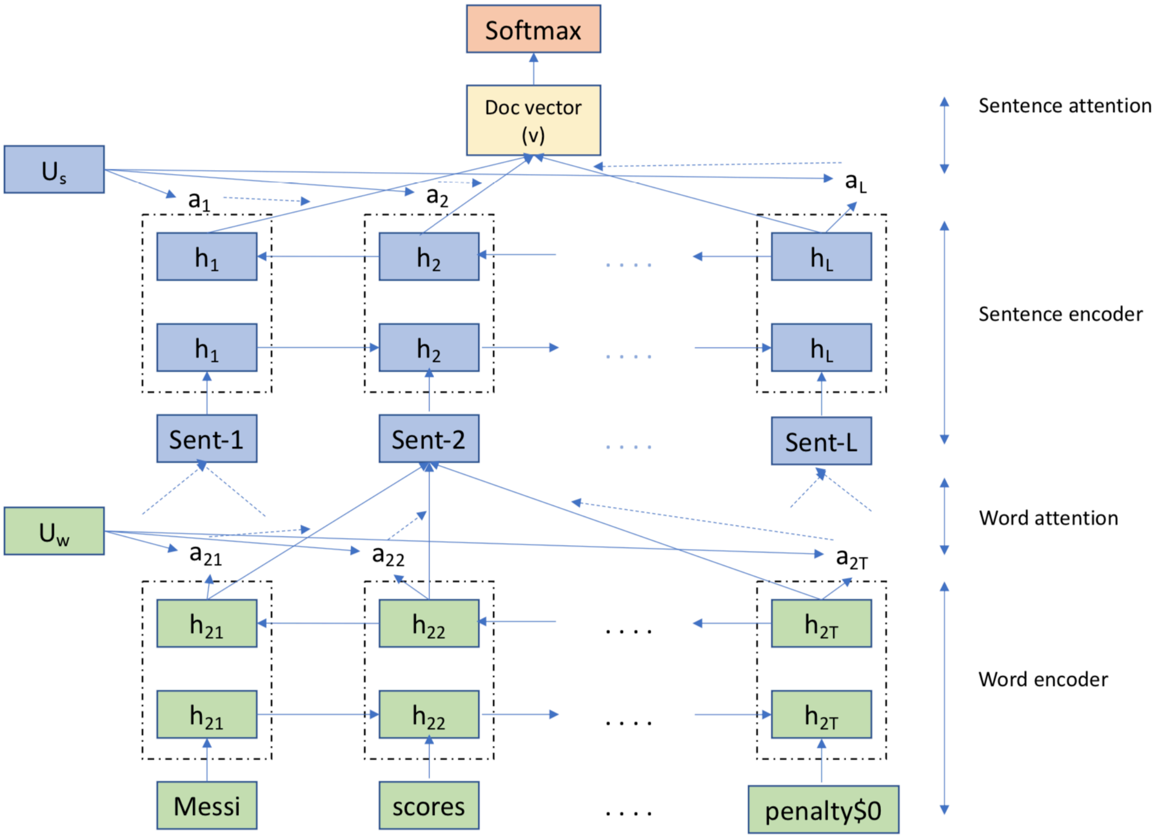


Figure 2.4. The HAN classifier used in our ConWea framework. It is trained on our contextualized corpus with the generated pseudo-labels.

2.5 Pseudo-Label and Text Classifier

In this section, we discuss pseudo-label generation and text classifier. These two parts are not the focus of the proposed method. We briefly introduce them to make the chapter self-contained.

We generate pseudo-labels for unlabeled contextualized documents and train a classifier based on these pseudo-labels, similar to many other weakly supervised methods [5, 6, 7, 1, 2].

2.5.1 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 $\mathrm{tf}(\hat{w}, d)$ denote term-frequency of a contextualized word w in the contextualized document d and $\mathscr{S}_c$ represents set of seed words of class c, the document d is assigned a label $l(d)$ as follows:

$$l(d) = \arg\max_{l}\{\sum_{i} tf(s_i, d) | \forall s_i \in \mathscr{S}_l\} \tag{2.5}$$

2.5.2 Document Classifier

Our framework is compatible with any text classification model. We use Hierarchical Attention Networks (HAN) [30] 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 2.4. We train a HAN model on contextualized corpus with the generated pseudo-labels. The predicted labels are used in seed expansion and disambiguation.

2.6 Seed Expansion and Disambiguation

2.6.1 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 $\mathscr{C}_j$ 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 $\mathscr{C}_j$ after observing the presence of word w (i.e., $P(\mathscr{C}_j|w)$) should be very close to 100%. Therefore, we use $P(\mathscr{C}_j|w)$ as our label-indicative measure:

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

  where $f_{\mathscr{C}_j}$ refers to the total number of documents that are predicted as class $\mathscr{C}_j$, and among them, $f_{\mathscr{C}_j,w}$ 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,

  $$\mathbf{F}(\mathscr{C}_j, w) = \tanh\left(\frac{f_{\mathscr{C}_j}(w)}{f_{C_j}}\right)$$

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

- **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 $\mathscr{D}$ and $f_{\mathscr{D},w}$ represents the document frequency of word w, the IDF of a word w is computed as follows:

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

Similar to previous work [1], we combine these three measures using the geometric mean, resulting in the ranking score $R(\mathscr{C}_j, w)$ of a word w for a class $\mathscr{C}_j$.

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

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

2.6.2 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.

2.7 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.

2.7.1 Datasets

Following previous work [1], [2], we use two news datasets in our experiments. The dataset statistics are provided in Table 2.1. 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,

Table 2.1. Dataset statistics.

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

sports) and 25 fine-grained categories (e.g., dance, music, hockey, basketball).

- **The 20 Newsgroups (20News):** The 20News dataset[1] 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.

2.7.2 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** [38] 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 [39] 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.

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

Table 2.2. Evaluation Results for All Methods on Fine-Grained and Coarse-Grained Labels. Both micro-F_1(mic-F_1) and macro-F_1(mac-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	Mic-F_1	Mac-F_1	Mic-F_1	Mac-F_1	Mic-F_1	Mac-F_1	Mic-F_1	Mac-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

- **Doc2Cube** [1] considers label surface names as seed set and performs multi-dimensional document classification by learning dimension-aware embedding.

- **WeSTClass** [2] 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 [21], 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.

2.7.3 Experiment Settings

We use pre-trained `BERT-base-uncased`[2] to obtain contextualized word representations. We follow [9] 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[3] and Dataless[4] with the hyper-parameters mentioned in their original papers.

2.7.4 Performance Comparison

We summarize the evaluation results of all methods in Table 2.2. 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:

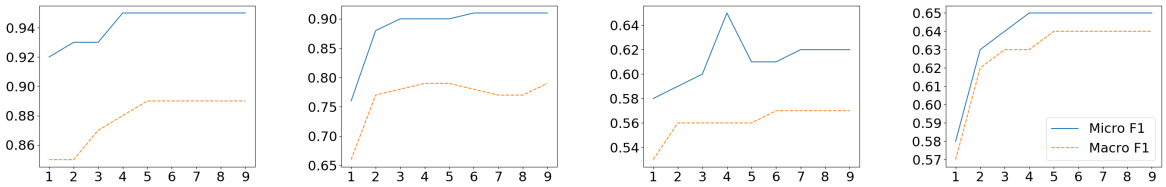
- Our proposed framework ConWea outperforms all the other methods with significant margins. By contextualizing the corpus and resolving the interpretation of seed words, ConWea achieves inspiring performance, demonstrating the necessity and the importance of using contextualized weak supervision.

- 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

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

[3] `https://github.com/yumeng5/WeSTClass`

[4] `https://cogcomp.org/page/software_view/Descartes`



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

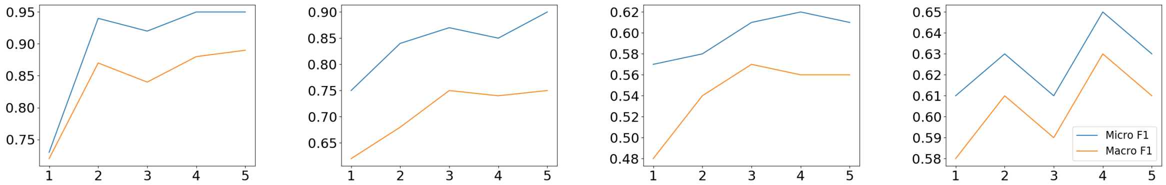
Figure 2.5. Micro- and Macro-F_1 scores w.r.t. the number of iterations.

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 [9], is adaptive to the input corpus, without requiring any additional human annotations. However, WSD methods(e.g., [21]) 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.



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

Figure 2.6. Micro- and Macro-F$_1$ scores w.r.t. the number of seed words.

2.7.5 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 2.5. 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.

2.7.6 Number of Seed Words

We vary the number of seed words per class and plot the F_1 score in Figure 2.6. 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 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.

2.7.7 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 2.3 shows a column-by-column comparison for

Table 2.3. 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

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 2.3, 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".

2.8 Summary

In this chapter, 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.

Chapter 2, in full, is a reprint of the material as it appears in Mekala, Dheeraj; Shang, Jingbo. "Contextualized weak supervision for text classification," in Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 323–333, 2020. The dissertation/thesis author was the primary investigator and author of this paper.

Chapter 3

META: Metadata-Empowered Weak Supervision for Text Classification

In this chapter, we present our proposed method META, that leverages metadata information as an additional source of weak supervision and incorporates it into the classification framework.

3.1 Motivation & Overview

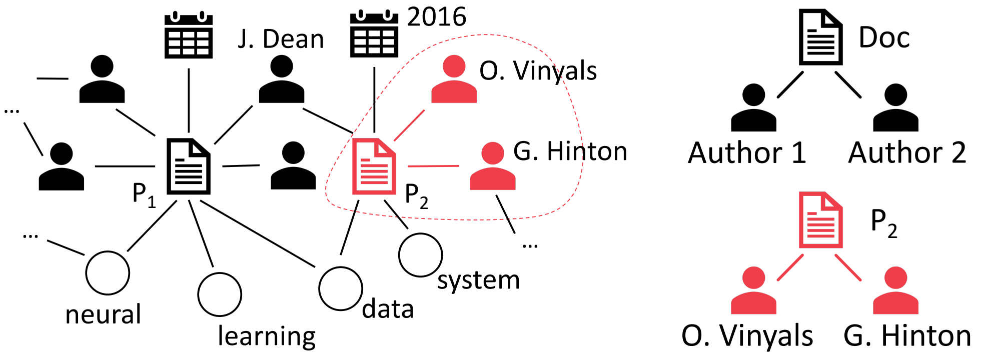
Weakly supervised text classification has recently gained much attention from the researchers because it reduces the burden of annotating the data. So far, the major source of weak supervision lies in text data itself [5, 7, 6, 1, 2, 3]. These methods typically require a few user-provided seed words for each class as weak supervision. They expand seed words with generated pseudo labels and improve their text classifier in an iterative fashion.

Metadata information (e.g., author, published year) in addition to textual information, is widely available across various domains (e.g., news articles, social media posts, and scientific papers) and it could serve as a strong, complementary weak supervision source. Take a look at the research papers in Figure 3.1(a) as an example. It shall be learned in a data-driven manner that *G. Hinton* is a highly-reputed machine learning researcher, thus his presence is a strong indicator of a paper belonging to the *Machine Learning* category.

Distilling effective metadata for weak supervision faces several major challenges. Meta-

Paper	**Authors**	**Year**	**Category**
P_1	G. Hinton, S. Osindero, YW. Teh	2006	ML
P_2	G. Hinton, O. Vinyals, J. Dean	2015	ML
P_3	J. Dean, S.Ghemawat	2008	Sys

(a) Examples of research papers with metadata.



(b) A text-rich network view of the papers.

(c) A motif pattern and a motif instance

Figure 3.1. Text corpus, text-rich network, and motif.

data is often multi-typed, each type and the type combinations could have very different semantics and may not be equally important. Moreover, even entities within a single metadata type could be noisy. Continuing our example in Figure 3.1(a), we shall notice that *year* is less helpful than an *author* to do classification. Among the authors, *J. Dean* might be an important figure but has research interests spanning across different domains. However, if we join the *author* with *year*, it carries more accurate semantics, and we may discover *J. Dean* has more interest in machine learning in recent years, thus becoming highly label-indicative.

Bearing the challenges in mind, we propose META, a principled framework for metadata-empowered weakly-supervised text classification. As illustrated in Figure 3.1 and Figure 3.2, we first organize the text data and metadata together into a text-rich network. The network structure gives us a holistic view of the corpus and enables us to rank and select useful metadata entities. We leverage motif patterns [40, 41, 42] to model typed metadata as well as their combinations. A motif pattern is a subgraph pattern at the meta-level that captures higher-order connections and the semantics represented by these connections. It serves as a useful tool to model typed

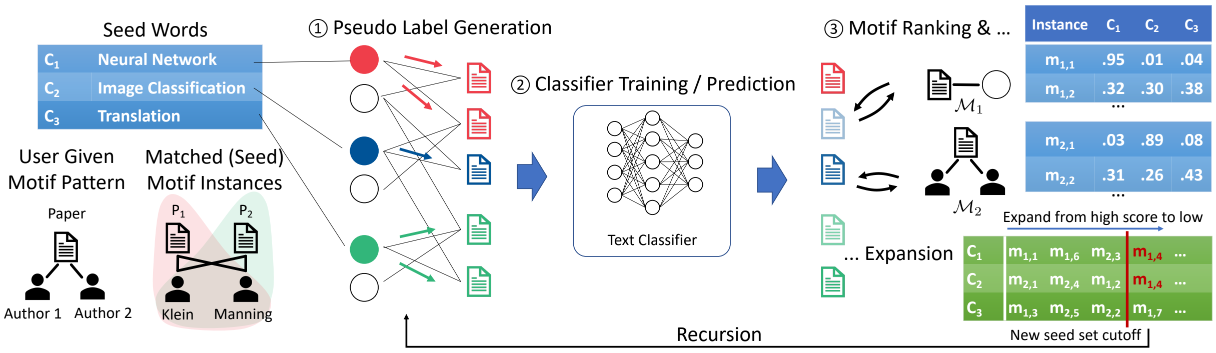

Seed Words
C_1 Neural Network
C_2 Image Classification
C_3 Translation
User Given Motif Pattern
Matched (Seed) Motif Instances
Paper
Author 1
Author 2
P_1
P_2
Klein
Manning
① Pseudo Label Generation
② Classifier Training / Prediction
Text Classifier
③ Motif Ranking & ...
... Expansion
$\mathcal{M}_1$
$\mathcal{M}_2$
Instance C_1 C_2 C_3
$m_{1,1}$ .95 .01 .04
$m_{1,2}$ .32 .30 .38
$m_{2,1}$ .03 .89 .08
$m_{2,2}$ .31 .26 .43
Expand from high score to low
C_1 $m_{1,1}$ $m_{1,6}$ $m_{2,3}$ $m_{1,4}$ ...
C_2 $m_{2,1}$ $m_{2,4}$ $m_{1,2}$ $m_{1,4}$ ...
C_3 $m_{1,3}$ $m_{2,5}$ $m_{2,2}$ $m_{1,7}$ ...
New seed set cutoff
Recursion

Figure 3.2. Our META framework. In each iteration, we generate pseudo labels for documents, train the text classifier, and rank all words and motif instances in a unified ranking framework. We then expand seed sets until an automatic cutoff is reached.

edges, typed paths (a.k.a. meta-paths) [43], and higher-order structures in the network. With little effort, users can specify a few possibly useful motif patterns as input to our model. We develop a unified, principled ranking mechanism to select label-indicative motif instances and words, forming expanded weak supervision. Note that, such instance-level selection process also implicitly refines the motif patterns, ensuring the robust performance of META even when irrelevant motif patterns exist in input. It is worth a mention that META is compatible with any text classifiers.

Our contributions are summarized as follows:

- We explore to incorporate metadata information as an additional source of weak supervision for text classification along with seed words.
- We propose a novel framework META, which introduces motif patterns to capture the high-order combinations among different types of metadata and conducts a unified ranking and selection of label-indicative motif instances and words.
- We conduct experiments on two real-world datasets. The results and case studies demonstrate the superiority of incorporating metadata as parts of weak supervision and verify the effectiveness of META.

3.2 Related Work

In this section, we review the literature about (1) weakly supervised text classification methods, (2) text classification with metadata, and (3) document classifiers.

3.2.1 Weakly Supervised Text Classification

Due to the training data bottleneck in supervised classification, weakly supervised classification has recently attracted much attention from researchers. The majority of weakly supervised classification techniques require seeds in various forms, including label surface names [14, 13, 1], label-indicative words [38, 2, 1, 3], and labeled-documents [10, 12, 11, 2].

Dataless [13] considers label surface names as seeds and classifies documents by embedding both labels and documents in a semantic space and computing semantic similarity between a document and a potential label; Along similar lines, Doc2Cube [1] expands label-indicative words using label surface names and performs multi-dimensional document classification by learning dimension-aware embedding; WeSTClass [2] considers both word-level and document level supervision sources. It first generates bag-of-words pseudo documents for neural model pre-training, then bootstraps the model on unlabeled data. This method is later extended to a hierarchical setting with a pre-defined hierarchy [15]; ConWea [3] leverages contextualized representation techniques to provide contextualized weak supervision for text classification.

However, all these techniques consider only the text data and don't leverage metadata information for classification. In this chapter, we focus on user-provided seed words and mine label-indicative words and metadata in an iterative manner.

3.2.2 Text Classification with Metadata

Previous studies try to incorporate metadata information to improve the performance of the classifier. [44] and [45] consider the user and product information as metadata for document-level sentiment classification; [46] use author information for paper classification; [47] employ user biography data for tweet localization. However, all these frameworks are in a supervised setting and use fixed metadata types for each task whereas our method is generalized for different metadata types and multiple metadata combinations.

Another way to leverage metadata for text understanding is to organize the corpus into a heterogeneous information network. A straightforward approach is to obtain document representations using their respective meta-path guided node embeddings [48, 49] and train a classifier. However, higher-order connectivity cannot be captured by meta-paths and this approach can't handle new documents directly without re-training the embeddings. Recently, [50] proposed a minimally supervised framework to categorize text with metadata. However, they require labeled documents as supervision and they only consider typed edges in the model. Network motifs [41] can capture higher-order connectivity and have been proved fundamental in complex real-word networks across various domains [40]. [42] leveraged motifs for topic taxonomy construction in an unsupervised setting. Our proposed method mines highly label-indicative metadata information with a unified motif and word ranking framework, and effectively expands weak supervision to improve document classification.

3.2.3 Document classifier

Document classification has been a long-studied problem in Natural Language Processing. CNN-based classifiers [31, 51, 33], RNN-based classifiers [52] achieve competitive performance. [30] proposed Hierarchical Attention Network (HAN) for document classification that performs attention first on the sentences in the document, and on the words in the sentence to find the most important sentences and words in a document. Though our framework uses HAN as the document classifier, it is also compatible with all the above-mentioned text classifiers. We choose HAN for the demonstration purpose.

3.3 Preliminaries

In this section, we briefly discuss the concepts that are essential to understand our framework such as Text-rich network, motif pattern and motif instances. In the end, we formally formulate the problem and provide a brief description of our proposed framework.

3.3.1 Documents as Text-rich Network

Given a collection of n text documents $\mathscr{D} = \{\mathscr{D}_1, \mathscr{D}_2, \ldots, \mathscr{D}_n\}$, and their corresponding metadata, we propose to organize them into a *text-rich network*, as illustrated in Figure 3.1(b). A text-rich network is a heterogeneous network with documents, words, different types of metadata as nodes, and their associations as edges. For example, our text-rich network for research papers has papers, words, authors, and publication years as nodes. Each paper is connected to its associated words and metadata nodes. Such a network provides a holistic and structured representation of the input.

3.3.2 Seed Words and Motif Patterns

Users are asked to provide a few seed words $\mathscr{S} = \{\mathscr{S}_1^w, \mathscr{S}_2^w, \ldots, \mathscr{S}_l^w\}$ for each of l classes (i.e., $\mathscr{C}_1, \mathscr{C}_2, \ldots, \mathscr{C}_l$) in our classification problem, as well as k motif patterns $\{\mathscr{M}_1, \mathscr{M}_2, \ldots, \mathscr{M}_k\}$. Motif patterns are sub-graph patterns at the meta-level (i.e., every node is abstracted by its type). They are able to capture semantics and higher-order inter-connections among nodes. A motif instance is a sub-graph instance in the graph that follows a motif pattern. Figure 3.1(c) presents an example of a motif pattern that captures co-authorship and a motif instance following this motif pattern. In this framework, we discover seed motif instances for each class label, denoted as $\{\mathscr{S}_1^m, \mathscr{S}_2^m, \ldots, \mathscr{S}_l^m\}$.

3.3.3 Problem Formulation

Given the text-rich network and user-provided seed words and motif patterns as input, we aim to build a document classifier, assigning a class label $\mathscr{C}_j$ to each document $\mathscr{D}_i$.

3.3.4 Framework Overview

As shown in Figure 3.2, META is an iterative framework, generating pseudo labels and training the text classifier alternatively, similar to many other weakly supervised text classification methods [7, 1, 2]. One iteration in META consists of the following steps:

- Generate pseudo labels based on the seeds.
- Train a text classifier based on pseudo labels.
- Rank and select words and motif instances to expand the seeds.

We repeat these steps iteratively. We denote the number of iterations as T, which is the only hyper-parameter in our framework.

The novelty of META mainly lies in integrating two sources of weak supervisions, seed motif instances, and seed words. Given each motif instance m or each word w, for each label l, we estimate a *ranking score* $\mathscr{R}_{m,l}$ or $\mathscr{R}_{w,l}$ ranging between 0 and 1, measuring how label-indicative it is to the particular label l. Such ranking scores are utilized to select new seed motif instances and seed words. Note that, while this selection is conducted at the instance level, it also selects motif patterns implicitly and therefore ensures robust performance when users provide some irrelevant motif patterns.

3.4 Pseudo Labels and Text Classifier

In this section, we discuss pseudo-label generation and text classifier. Based on seed words, seed motif instances, and their respective ranking scores for each class, we generate pseudo labels for unlabeled text documents and train a classifier based on these pseudo labels. In the first iteration, we have no seed motif instances and the ranking score is 1 for all seed words.

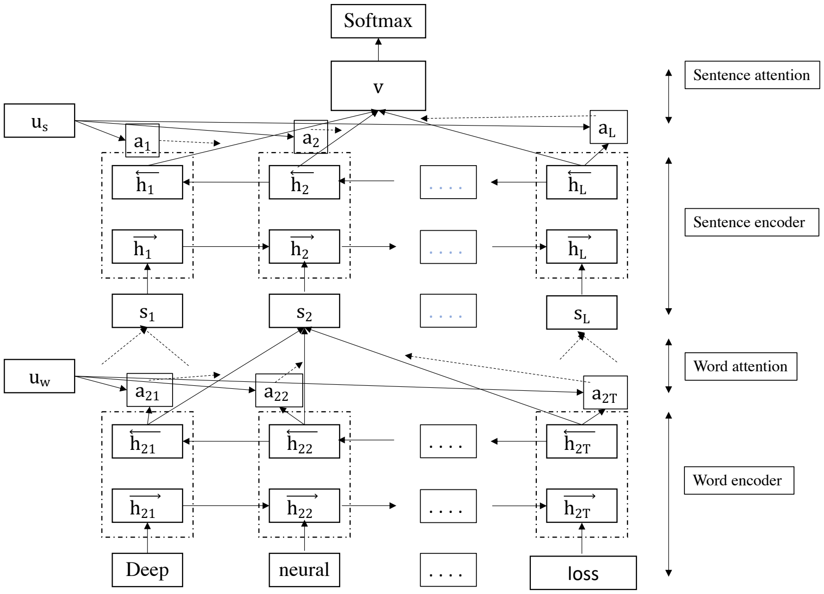


Figure 3.3. HAN Classifier used in our META.

3.4.1 Pseudo-Label Generation

Suppose we have seed word sets $\mathscr{S}^{w}_{1..l}$ and seed motif instance sets $\mathscr{S}^{m}_{1..l}$ for all l labels, we generate pseudo labels using a simple yet effective count-based technique. Specifically, given a document $\mathscr{D}_i$, the probability that it belongs to the class l is proportional to the aggregated ranking scores of its respective seed words and seed motif instances.

$$P(l|\mathscr{D}_i) \propto \sum_{w\in\mathscr{D}_i\cap\mathscr{S}^w_l} f_{\mathscr{D}_i,w}\cdot\mathscr{R}_{w,l} + \sum_{m\in\mathscr{D}_i\cap\mathscr{S}^m_l} \mathscr{R}_{m,l}$$

where $f_{\mathscr{D}_i,w}$ is the term frequency of the word w in document $\mathscr{D}_i$. The pseudo label of document $\mathscr{D}_i$ is then assigned as follows:

$$l(\mathscr{D}_i) = \arg\max_{l} P(l|\mathscr{D}_i)$$

3.4.2 Document Classifier

Our framework is compatible with any text classification model as a classifier. We use Hierarchical Attention Networks (HAN) [30] as the classifier. HAN is designed to capture the

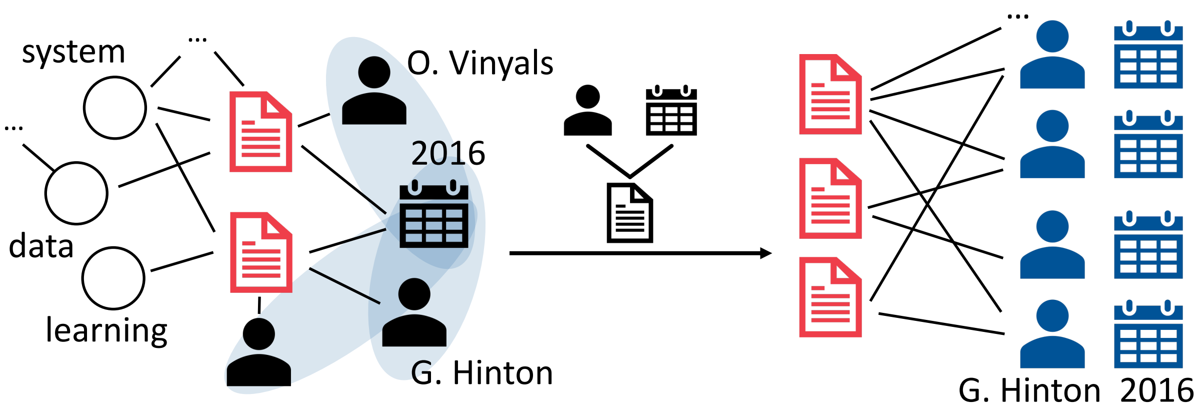


Figure 3.4. Using motif patterns, we construct bipartite graphs from the text-rich network linking documents to their respective motif instances.

hierarchical document structure i.e. words – sentences – documents. As illustrated in Figure 3.3, HAN performs attention first on the sentences in the document to find the important sentence in a document and on the words in the sentence to identify important words in a sentence. We train a HAN model on unlabeled documents with the generated pseudo-labels. For the document $\mathscr{D}_i$, it estimates the probability $\hat{Y}_{i,l}$ for each class l. Such predicted distributions are used in the expansion of seed words and motifs.

3.5 Unified Seed Ranking and Expansion

In this section, we describe the ranking score and seed expansion technique. Once the text classifier is trained, we rank words and motif instances together for each class. Then, we expand the seed sets by adding top-ranked words and motif instances. This improves the quality of the weak supervision over iterations, thereby improving the text classifier. We present our design of the unified ranking and expansion as follows.

3.5.1 Ranking Score Design

An ideal seed word or motif instance for a particular class should be highly relevant and highly exclusive to this class. So an effective ranking score must quantify relevance and exclusiveness. Such a ranking score for words alone has been explored by previous studies [1, 3], typically based on similarity and frequency-based metrics. In this framework design, we have motif instances in addition to words, therefore, we build upon the text-rich network to unify the ranking process.

Given k user-provided motif patterns $\mathscr{M}_1$, …, $\mathscr{M}_k$ and the text-rich network $\mathscr{G}$, we construct k bipartite graphs $\mathscr{G}_1^B, \ldots, \mathscr{G}_k^B$, one for each motif pattern (see Figure 3.4). In the i-th bipartite graph $\mathscr{G}_i^B$, the node set contains two parts: (1) all documents and (2) all motif instances following the motif pattern $\mathscr{M}_i$ in the text-rich network $\mathscr{G}$; The edges in the graph $\mathscr{G}_i^B$ connect the documents to the motif instances which are subsets of the metadata associated with the documents.

For the sake of simplicity, we introduce one more motif pattern, document–word. It makes words a special case of motif instances, and one can easily construct a similar bipartite graph for words. Therefore, in the rest of this section, we use motif instances to explain our ranking score design.

For each motif pattern $\mathscr{M}$, we conduct one personalized random walk on its corresponding bipartite graph $\mathscr{G}^B$ for each label l. Specifically, we normalize each column of the adjacency matrix of the bipartite graph $\mathscr{G}^B$ by the degree of its respective node, resulting in the transition matrix $\mathbf{W}$. Suppose $\mathbf{p}_{l,u}$ represents the personalized PageRank (PPR) score of each node u for each label l, we initialize the PPR score of each document node to $\hat{Y}_{i,l}$ and PPR score of each motif instance node to 0. This initialization ensures that a random walk starts from a document node and since $\mathscr{G}^B$ is bipartite, it ends at a motif instance node. We iteratively update the PPR scores as follows:

$$\mathbf{p}_l^{(t+1)} \leftarrow \mathbf{W}\mathbf{p}_l^{(t)}$$

Since each document node is initialized with probabilities corresponding to l and the random walk starts from a document node and ends at a motif instance node, this can be viewed as a label propagation problem. Based on the previous work in label propagation [53], similar nodes are more likely to form edges and the PPR score is used to measure the similarity. Therefore, we believe that $\mathbf{p}_{l,m}$ reflects the *relevance* of a motif instance m to the particular class label l.

Though the absolute values of PPR scores are quite small, their relative magnitude conveys their affinity towards a label. Therefore, we normalize these PPR scores into a distribution, resulting in the ranking scores. Mathematically, for a label l, the ranking score of a motif instance m is:

$$\mathscr{R}_{m,l} = \frac{\mathbf{p}_{l,m}}{\sum_{l' \in \mathscr{C}} \mathbf{p}_{l',m}}$$

If a motif instance has similar relevance to multiple labels, the ranking score distribution becomes flat irrespective of the magnitude of its respective PPR scores. From this, we realize that our ranking score also quantifies *exclusiveness*, which is an essential characteristic of a highly label-indicative term.

Based on this ranking score, we rank words and motif instances in a unified manner and expand the seed word set and seed motifs set.

3.5.2 Expansion

Given the ranking scores of all words and motif instances for every label, we expand the seed words and seed motifs simultaneously for all labels. Intuitively, a highly label-indicative motif instance would not belong to the seed sets of multiple labels. Therefore, when any motif instance is expanded to seed sets of multiple classes, we stop the expansion of motif instances of the corresponding motif pattern. Also, we set a hard threshold of $\frac{1}{|\mathscr{C}|}$, where $|\mathscr{C}|$ is the number of classes, on ranking scores for those added motif instances. In this way, the number of new seed words and seed motif instances is decided by the method automatically. It is worth mentioning that our expansion here is adaptive and every label may have a different number of seeds. Note that, in the first iteration, pseudo labels are generated using only seed words but ranking scores are obtained for all words and motif instances. The highly ranked motif instances and words are used as seeds in further iterations.

After expanding the seed sets for every label, we generate pseudo labels and train the classifier. This process is repeated iteratively for T iterations.

Table 3.1. Dataset statistics.

Dataset	# Docs	# Classes	Avg Doc Len
DBLP	38,128	9	893
Book Graph	33,594	8	620

3.6 Experiments

In this section, we evaluate META and compare it with existing techniques on two real-world datasets in a weakly supervised classification setting.

3.6.1 Datasets

We conduct experiments on the DBLP dataset [54] and the Book Graph dataset [55, 56]. The dataset statistics are shown in Table 3.1. The details of the datasets are mentioned below.

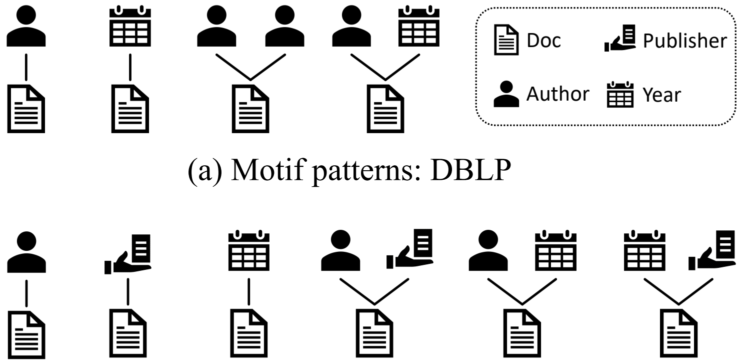
- **DBLP dataset:** The DBLP dataset contains a comprehensive set of research papers in computer science. We select $38,128$ papers published in flagship venues. In addition to text data, it has information about authors, published year, and venue for each paper. There are $9,300$ distinct authors and 42 distinct years. For each paper, we annotate its research area largely based on its venue as the classification objective[1]. Therefore, in our experiments, we drop the venue information to ensure a fair comparison.

- **Book Graph dataset:** The Book Graph dataset is a collection of the description of books, user-book interactions, and users' book reviews collected from a popular online book review website named Goodreads[2]. We select books belonging to eight popular genres[3]. The genre of a book is viewed as the label to be predicted. The total number of books selected is $33,594$. We use the title and description of a book as text data and author,

[1] Classes in DBLP: (1) computer vision, (2) computational linguistics, (3) biomedical engineering, (4) software engineering, (5) graphics, (6) data mining, (7) security and cryptography, (8) signal processing, (9) robotics, and (10) theory.

[2] `https://www.goodreads.com/`

[3] Classes in Book Graph: (1) children, (2) graphic comics, (3) paranormal fantasy, (4) history & biography, (5) crime, mystery thriller, (6) poetry, (7) romance, and (8) young adult.



(a) Motif patterns: DBLP

(b) Motif patterns: Book Graph

Figure 3.5. Motif Patterns used in Experiments.

publisher, and year as metadata. In total, there are 22,145 distinct authors, 5,186 distinct publishers, and 136 distinct years.

3.6.2 Motif Patterns

The motif patterns we used as metadata information for DBLP and Book Graph datasets are shown in Figure 3.5.

3.6.3 Seed Words

The seed words are obtained as follows: we asked 5 human experts to recommend 5 seed words for each class and selected the final seed words based on majority voting i.e. (>3 recommendations).

3.6.4 Evaluation Metrics

Both datasets are imbalanced with respect to the label distribution. Being aware of this fact, we adopt micro- and macro-F_1 scores as evaluation metrics.

3.6.5 Implementation Details

To make the model robust to multi-word phrases as supervision, we extract phrases using Autophrase [34, 35]. We set the word vector dimension to be 100 for all the methods that use word embeddings. We set the number of iterations parameter for META to 9.

3.6.6 Compared Methods

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

- **IR-TF-IDF** treats seed words as a query. It computes the relevance of a document to a class by aggregating the TF-IDF values of its seed words. Each document is assigned the label which is the most relevant to this document.

- **Word2Vec** learns word vector representations [57] for all words in the corpus. It computes label representations by aggregating the word vectors of all its seed words. Each document is assigned the label whose cosine similarity with this document is maximum.

- **Doc2Cube** [1] considers label surface names as seed set and performs multi-dimensional document classification by learning dimension-aware embedding.

- **WeSTClass** [2] leverages seed words to generate bag-of-words pseudo documents for neural model pre-training and then bootstraps the model on unlabeled data. Specifically, we compare with WeSTClass-CNN which is the best configuration under our setting. We use the public implementations of WeSTClass[4] with the hyperparameters mentioned in the paper.

- **Metapath2Vec** [48] learns node representations in the text-rich network using meta-path-guided random walks by capturing the structural and semantic correlations of differently typed nodes. We use the first two motif patterns in Figure 3.5(a) and the first three motif patterns in Figure 3.5(b) as meta-paths because the rest cannot be represented as meta-paths. We generate pseudo-labels using the seed words and train a logistic regression classifier with document nodes representations as input to predict the labels.

We denote our framework with HAN classifier as **META**, with CNN classifier as **META-CNN**, and with BERT(bert-base-uncased) classifier as **META-BERT**. We also compare with their

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

Table 3.2. Evaluation Results on Two Datasets. **++** represents that the input is metadata-augmented.

	DBLP		Books Graph	
Methods	Mi-F_1	Ma-F_1	Mi-F_1	Ma-F_1
IR-TF-IDF	0.19	0.20	0.24	0.29
Word2Vec	0.23	0.22	0.28	0.26
Doc2Cube	0.37	0.36	0.33	0.31
WeSTClass	0.58	0.53	0.42	0.41
Metapath2Vec	0.64	0.61	0.47	0.48
IR-TF-IDF++	0.19	0.20	0.24	0.29
Word2Vec++	0.24	0.21	0.26	0.25
Doc2Cube++	0.40	0.38	0.36	0.33
WeSTClass++	0.60	0.55	0.47	0.43
META	**0.66**	**0.63**	0.62	**0.63**
META-CNN	0.61	0.58	0.54	0.55
META-BERT	0.64	0.61	**0.63**	**0.63**
META-NoMeta	0.61	0.58	0.58	0.58
META-CNN-NoMeta	0.56	0.53	0.53	0.53
META-BERT-NoMeta	0.58	0.57	0.60	0.60
HAN-Sup	0.75	0.72	0.77	0.76
HAN-Sup++	0.79	0.77	0.81	0.81

respective ablated versions **META-NoMeta**, **META-CNN-NoMeta**, **META-BERT-NoMeta** where metadata information is not expanded and not considered while generating pseudo labels.

For a fair comparison, we also present results of all the baselines on the **metadata-augmented** datasets, where a token for every relevant motif instance is appended to the text data of a document. This is denoted by **++** in Table 3.2, e.g., WeSTClass++ represents the performance of WeSTClass on metadata-augmented datasets.

We also present the performance of HAN in a supervised setting which is denoted as **HAN-Sup**. The results of HAN-Sup reported are on the test set which follows an 80-10-10 train-dev-test split.

3.6.7 Performance Comparison

The evaluation results of all methods are summarized in Table 3.2. We can observe that our proposed framework outperforms all the compared weakly supervised methods. We discuss the effectiveness of our proposed META as follows:

- META achieves the best performance among all the compared weakly supervised methods with significant margins. By extracting the highly label-indicative motif instances along with words and using them together in pseudo label generation, META successfully leverages metadata information and achieves superior performance.

- We observe that the performance of META is better than all the compared weakly supervised models on metadata-augmented datasets. By comparing those ++ methods with their text-only counterparts, one can easily observe that adding metadata in text classification is indeed helpful. However, META does not restrict to single metadata types and goes beyond by employing motif patterns to capture the metadata information. It is successful in identifying the appropriate label-indicative metadata combinations and therefore achieves even better performance.

- The comparison between META and Metapath2Vec demonstrates the advantages of motif patterns over the meta-paths. For example, on the Book Graph dataset, the last three motif patterns in Figure 3.5(b) cannot be represented through meta-paths and this significantly affects the performance. It's also worth mentioning that Metapath2Vec cannot handle new documents directly without re-training the embedding whereas our framework can directly predict without any additional effort.

- The comparison between META and the ablation method META-NoMeta demonstrates the effectiveness of our motif instance expansion. For example, on the Book Graph dataset, the motif instance expansion improves the micro-F1 score from 0.58 to 0.62 and macro-F1 score from 0.58 to 0.63, which are quite significant.

- The comparison between META-CNN, META-BERT, and their respective ablated versions META-CNN-NoMeta, META-BERT-NoMeta demonstrate that our proposed approach provides significant additive gains to different classifiers and thereby showing the effectiveness

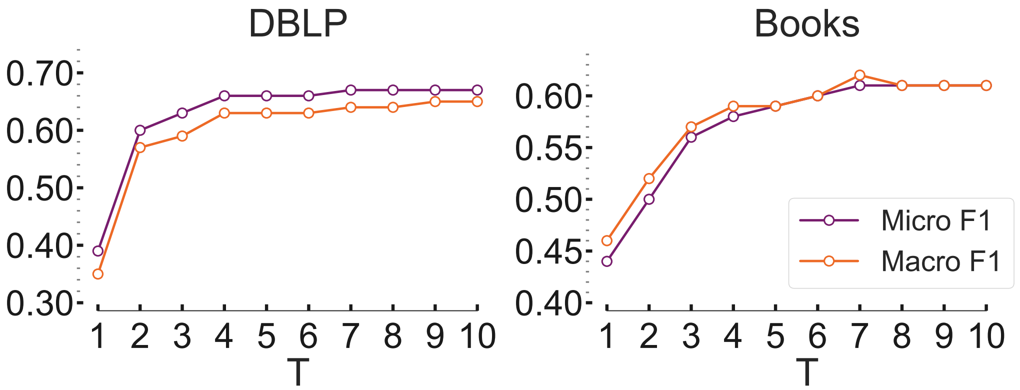


Figure 3.6. Micro- and Macro-F_1 scores w.r.t. the number of iterations.

of leveraging metadata information as an additional source of weak supervision.

- The comparison between META and HAN-Sup demonstrates that META is effective in decreasing the gap between the performance of the weakly supervised and supervised settings.

3.6.8 Parameter Study

The only hyper-parameter in our framework META is T, the number of iterations. We experiment on both datasets to study the effect of the number of iterations on the performance. The plots of micro-F1 score and macro-F1 score with respect to the number of iterations are shown in Figure 3.6. We observe that the performance increases initially and gets gradually converged by 6 or 7 iterations. We also observe that the expanded seed words and seed motifs have become almost unchanged. While there is some fluctuation, a reasonably large T, such as $T = 9$ or $T = 10$, is recommended.

3.6.9 Number of Seed Words

We vary the number of seed words per class and plot the performance in Figure 3.7. We observe that the performance increases as the number of seed words increase, which is generally intuitive. For reasonable performance, we observe that three seed words are sufficient.

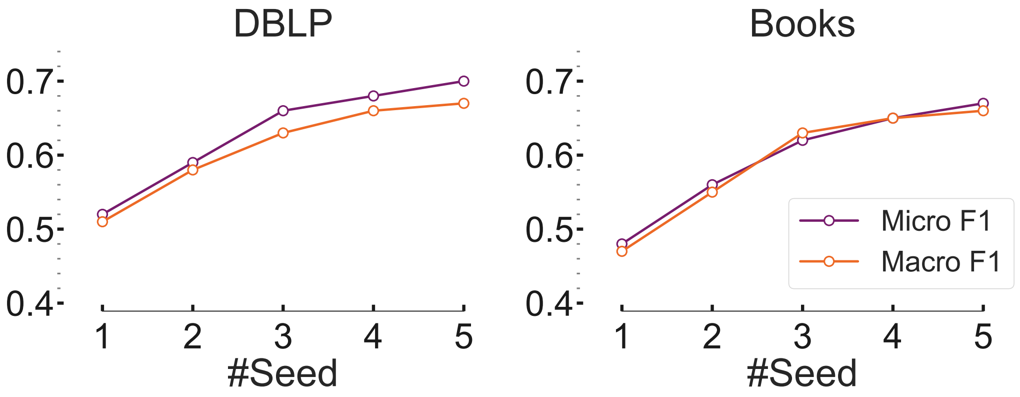


Figure 3.7. Micro- and Macro-F_1 scores w.r.t. the number of seed words.

Table 3.3. Case Study: Expanded motif instances.

Expanded motif instances of Book Graph dataset					
Class	**Author**	**Publisher**	**Author-Publisher**	**Year**	**Author-Year**
children	Z. Fraillon, K. Argent	Brighter Child, HarperCollins Children's Books	(**N. Gaiman**, Bloomsbury UK) (M. Fox, Penguin Australia)	N/A	(**N. Gaiman**, 2004) (S. Blackall, 2010)
comics	F. Teran, B. Kane	Marvel, Titan Books Ltd	(**N. Gaiman**, Marvel) (T. McFarlane, Marvel Comics)	N/A	(T. Hairsine, 2013) (A. Sinclair, 2009)
fantasy	J. Barne, S. Dubbin	DAW Books, Inc., Edge Publishing	(W. King, Titan Books Ltd) (G.J. Grant, Prime Books)	N/A	(G.J. Grant, 2012) (M. Lingen, 2012)
poetry	B. Guest, E. Dickinson	Shearsman Books, Souvenir Press	(**N. Gaiman**, MagicPress) (R. Browning, Wordsworth Editions)	1692, 1914	(E. Dickinson, 1959) (J. McCrae, 1929)

3.6.10 Case Studies

We present case studies to showcase the effectiveness of our framework in addressing the challenges of leveraging metadata.

Leveraging Metadata Combinations

Table 3.3 shows a few samples of expanded motif instances. First, let's take a look at motif instances related to authors and publishers. We can observe that strong label-indicative authors and publishers are mined accurately. For example, *Marvel*, a widely known comics publisher, is present in the expanded publishers for *comics* genre; A classic American poet *E. Dickinson* is successfully identified as label-indicative for *poetry* genre.

Note that, the author *N. Gaiman* (in blue) who has written books in multiple genres including comic books, graphic novels, etc., is not a label-indicative author for any of these categories, because he is not exclusive to any one category, which is accurately captured by

Table 3.4. Case Study: Percentage of motif instances expanded for Book Graph dataset. **A** stands for author, **P** for publisher and **Y** for year.

	Percentage of motif instances expanded					
Label	**A**	**P**	**Y**	**A-P**	**P-Y**	**A-Y**
children	5.12	9.42	0	9.21	12.73	9.68
comics	4.91	1.33	0	9.52	1.48	14.11
fantasy	6.2	2.8	0	13.1	2.95	10.97
history	4.31	10.5	6.12	8.1	11.8	7.94
mystery	4.11	8.6	3.67	9.8	11.04	9.59
poetry	6.8	9.2	15.4	10	8.17	9.11
romance	5.6	13.5	1.47	9.6	12.28	9.19
y. adult	3.52	13.7	2.2	9.1	15.04	9.32

our framework. However, his works in various genres together with their respective publisher information form a unique label-indicative pattern which is reflected by the "Author-Publisher" motif pattern.

Now, adding *year* metadata into the loop, although "Year-Document" is a user-provided motif pattern, META identifies that *year* information alone is not much helpful in classification. This demonstrates the robustness of our framework when users provide some irrelevant motif patterns. However, if we combine author information with year, it then carries more accurate semantics, and we may discover that *N.Gaiman* had authored more children's books in early 2000, thus becoming highly label-indicative.

Eliminating Noise in Metadata

Table 3.4 presents the percentage of motif instances expanded out of the total motif instances following a motif pattern, for every label. One can observe that META actually prunes out many motif instances, as the final selection ratio is far less than 100%.

For the "**Y**ear-Document" motif pattern, we observe that its motif instances are only expanded for a few genres, which is generally intuitive. For example, one can see that a significant percentage of "Year-Document" motif instances expanded for *history* and *poetry*. After a closer

Table 3.5. Expanded seed words of *comics*, *history*, and *mystery* classes in Books dataset.

Expanded seed words	
Label	**Seed words**
comics	batman, superman, marvel, mary-jane, general zod
history	history, world war, world war ii, political science
mystery	serial killer, sherlock holmes, inspector lestrade

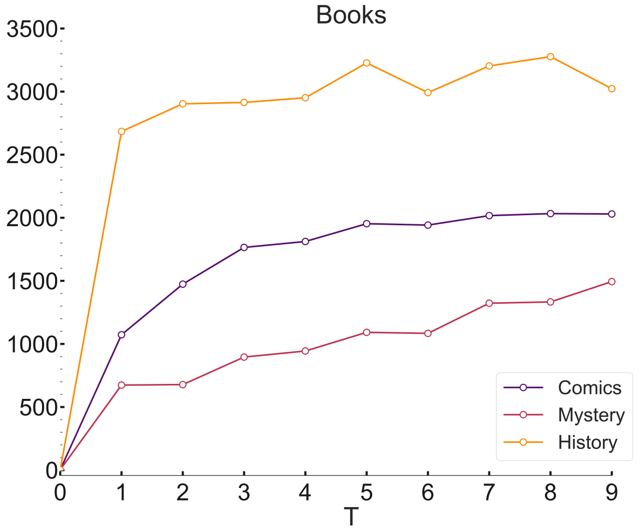


Figure 3.8. Number of seed words w.r.t. the number of iterations

inspection, we find that the expanded years were concentrated between the late 1800 and early 1900, thus developing an affinity for this time period.

One can also observe that the percentage of motif instances following the "Publisher-Document" motif pattern expanded varies for different labels, ranging from 1 to 13.5. This illustrates that our expansion is adaptive.

Seed words Expansion

Figure 3.8 shows the number of seed words expanded after each iteration for *comics*, *hystory*, and *mystery* classes in Books dataset. We observe that the number varies for each label because of our data-driven, adaptive thresholds, which is different for each label.

One can also observe that the the number increases over iterations and gets almost stagnated at the end, indicating that the seed sets are getting refined and converged. A few examples of expanded seed words are shown in Table 3.5.

3.7 Summary

In this chapter, we propose META, a novel framework that leverages metadata information as an additional source of weak supervision and incorporates it into the classification framework. Our method organizes the text data and metadata together into a text-rich network and employs motif patterns to capture appropriate metadata combinations. Using the initial user-provided seed words and motif patterns, our method generates pseudo labels, trains classifier, and ranks and filters highly label-indicative words, motifs in a unified manner and adds them to their respective seed set. Experimental results and case studies demonstrate that our model outperforms previous methods significantly, thereby signifying the advantages of leveraging metadata as weak supervision.

In the future, we are interested in effectively integrating different forms of supervision including annotated documents. Also, we only consider positively label-indicative metadata combinations currently. There should be negatively label-indicative combinations as well which can eliminate some classes from potential labels. This is another potential direction for the extension of our method.

Chapter 3, in full, is a reprint of the material as it appears in Mekala, Dheeraj; Zhang, Xinyang; Shang, Jingbo. "Meta: Metadata-empowered weak supervision for text classification," in Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 8351–8361, 2020. The dissertation/thesis author was the primary investigator and author of this paper.

Chapter 4

Coarse2Fine: Fine-grained Text Classification on Coarsely-grained Annotated Data

In this chapter, we introduce the task of coarse-to-fine grained classification and will lay out its significance. And then, we present our proposed method C2F, that leverages fine-grained label surface names as the only human guidance and utilize rich pre-trained generative language models as data generators to perform fine-grained text classification on coarsely-grained annotated data without any fine-grained annotations.

4.1 Motivation & Overview

In traditional text classification problems, the label set is typically assumed to be fixed. However, in many real-world applications, new classes especially more fine-grained ones will be introduced as the data volume increases. One commonly used method is to extend the existing label set to a label hierarchy by expanding every original coarse-grained class into a few new, fine-grained ones, and then assign a fine-grained label to each document. Using the directory structure for a set of files in computer as an example (see in Figure 4.1), people usually start organizing the files in a coarse-grained fashion like “Music” and “Academics”. Once the number of files in each of these coarse-grained directories increases, the categorization serves little purpose. Therefore, we would like to create new fine-grained sub-directories inside coarse-grained directories like {“rap”, “rock”, “oldies”} for “music” and similarly for “academics”.

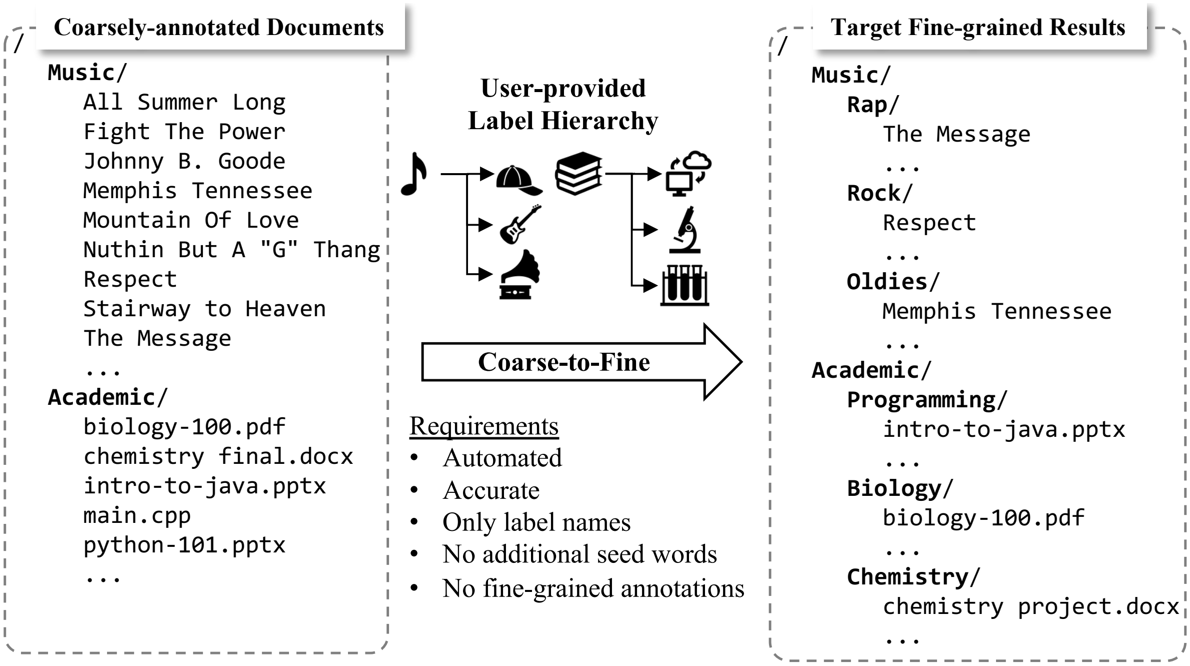


Figure 4.1. A visualization of our coarse-to-fine problem.

However, the process of assigning these files into fine-grained sub-directories typically begins with almost no supervision for fine-grained labels.

To accommodate such requirement, we introduce a new, important problem called coarse-to-fine grained classification, which aims to perform fine-grained classification on coarsely annotated data without any fine-grained human annotations. Note that, coarse-to-fine setting differs from generic zero-shot text classification in terms of having additional coarse supervision and a pre-conceived label hierarchy, though the final label set is available in either case. Therefore, we want to capture the coarse-grained supervision and label hierarchy available to perform fine-grained classification.

In this chapter, we propose a novel framework C2F as illustrated in Figure 4.2. In the absence of fine-grained human annotations, it uses fine-grained label surface names as weak-supervision signals and leverages pre-trained language models as data generators. Similar to previous work, we first generate weak supervision from the whole corpus by assuming label surface names as their respective strong label-indicators. For two iterations, C2F fine-tunes

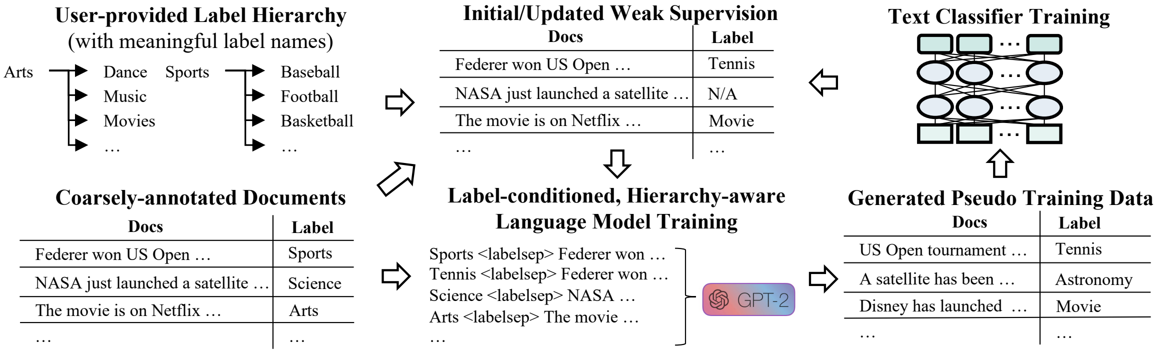

User-provided Label Hierarchy
(with meaningful label names)
Arts
Dance
Music
Movies
…
Sports
Baseball
Football
Basketball
…
Coarsely-annotated Documents
Docs
Label
Federer won US Open …
Sports
NASA just launched a satellite …
Science
The movie is on Netflix …
Arts
…
…
Initial/Updated Weak Supervision
Docs
Label
Federer won US Open …
Tennis
NASA just launched a satellite …
N/A
The movie is on Netflix …
Movie
…
…
Label-conditioned, Hierarchy-aware
Language Model Training
Sports <labelsep> Federer won …
Tennis <labelsep> Federer won …
Science <labelsep> NASA …
Arts <labelsep> The movie …
…
GPT-2
Text Classifier Training
Generated Pseudo Training Data
Docs
Label
US Open tournament …
Tennis
A satellite has been …
Astronomy
Disney has launched …
Movie
…
…

Figure 4.2. A visualization of our C2F framework.

language models based on weak supervision and trains a classifier based on generated pseudo-training data to refine weak supervision. We observe that raw weak supervision usually has a highly-skewed label distribution, especially at the beginning because the popularity of the label names varies. Since we have no prior knowledge about the underlying label distribution, to avoid significant deviations from that distribution, we opt to draw a balanced, weakly annotated subset through a stratified sampling before any model training. We propose to fine-tune language models in a label-conditioned, hierarchy-aware manner. Specifically, we inform the language models with label information by adding the label surface names at the beginning of each document. We further incorporate a regularization objective into the fine-tuning process that captures the constraints derived from the label hierarchy. Facilitated by this fine-tuning process, we then generate pseudo-training data for each fine-grained label and train a classifier. Next, using this fine-grained classifier's predictions over the coarsely annotated data, we select the samples with a high predicted probability for each respective fine-grained label. These sets of samples serve as weak supervision for the next iteration.

We conduct experiments on two real-world datasets containing both coarse and fine-grained labels. The results demonstrate the effectiveness of our framework in leveraging label hierarchy and rich pre-trained language model to perform fine-grained text classification with no supervision. Via thorough ablation, we isolate separate benefits accrued, initially just from using the label-conditioned, fine-tuned language model in the weak supervision pipeline, and the later incremental benefit once we incorporate our proposed regularization objective into the language model fine-tuning.

To the best of our knowledge, we are the first to work on the coarse-to-fine text grained classification, which aims to perform fine-grained classification on coarsely annotated data without any fine-grained annotations. It is also worth mentioning that C2F is compatible with almost any generative language model and text classification model. Our contributions are summarized as follows:

- We develop a label-conditioned fine-tuning formulation for language model to facilitate conditional corpus generation.
- We devise a regularization objective based on the coarse-fine label constraints derived from the label hierarchy to be consistent with the pre-conceived label hierarchy.
- We conduct extensive experiments demonstrate the superiority of C2F.

4.2 Related Work

In this section, we review the literature about different weakly supervised text classification methods.

There are three main sources of weak supervision: (1) a set of representative keywords for each class [2, 3, 4], (2) a few labeled documents [10, 11, 12, 2], (3) label surface names [1, 58, 59]. Typically, weakly supervised text classification frameworks obtain pseudo-labeled data, train a classifier, and improve the classifier by bootstrapping over unlabeled data. Seed-driven frameworks obtain pseudo-labeled data from user-provided seed words. When a few labeled documents are provided as weak supervision, the above-mentioned pipeline similarly starts with these as pseudo-labeled data. In this chapter, we focus on label surface names as the source of weak supervision. Along this line, Doc2Cube [60] expands label keywords from label surface names and performs multidimensional document classification by learning dimension-aware embedding; [58] identifies keywords for classes by querying replacements for class names using BERT and pseudo-labels the documents by string matching with the selected keywords. [59] proposed an adaptive representation learning method for obtaining label and document embedding and these document embeddings are further clustered to pseudo-label the corpus. However, all these methods perform flat text classification. Although our method performs text classification using only fine-grained label surface names as supervision, we have coarse-grained annotated data and leverage it to improve fine-grained classification. There are a few methods that perform weakly supervised hierarchical classification [15, 61]. However, our problem statement is different from hierarchical classification. We have coarse-grained annotated data and our framework utilizes it and label hierarchy to perform fine-grained text classification. Recently, [62] introduced coarse-to-fine weakly-supervised multi-label learning problem. However, they assume a few fine-grained labeled documents as supervision whereas our framework requires only label surface names. Additionally, our framework is generative in nature i.e. instead of pseudo-labeling the given corpus, we generate training data and train the classifier.

4.3 Preliminaries

In this section, we formulate the problem and provide a brief description of our proposed framework.

4.3.1 Problem Formulation

The input of our problem contains: (1) A tree-structured label hierarchy $\mathscr{T}$ with coarse-grained labels $\mathscr{C}$ at the first level and fine-grained labels $\mathscr{F}$ as their children. The m coarse-grained classes are named $\{\mathscr{C}_1, \mathscr{C}_2, \ldots, \mathscr{C}_m\}$, and k fine-grained classes are named as $\{\mathscr{F}_1, \mathscr{F}_2, \ldots, \mathscr{F}_k\}$. All these class names are in natural language (e.g., words or phrases) and assumed to be informative; and (2) a collection of n text documents $\mathscr{D} = \{\mathscr{D}_1, \mathscr{D}_2, \ldots, \mathscr{D}_n\}$ and their corresponding coarse-grained labels $\{c_1, c_2, \ldots, c_n\}$.

We record the mapping from each fine-grained class to its corresponding coarse-grained parent class as $f^{\uparrow}$: $\mathscr{F} \rightarrow \mathscr{C}$. The fine-grained classes in a coarse-grained label are represented by the coarse-to-fine mapping $f_{\downarrow}$: $\mathscr{C} \rightarrow \mathscr{P}(\mathscr{F})$, where $\mathscr{P}(\cdot)$ is the powerset operator, which generates the set of all subsets. In this problem, each coarse class maps to a non-empty subset of fine classes, and all these subsets of fine-classes taken together are mutually non-overlapping and exhaustive.

We aim to build a high-quality document classifier from these inputs, assigning a fine-grained class label $\mathscr{F}_j \in f_{\downarrow}(c_i)$ to each document $\mathscr{D}_i \in \mathscr{D}$.

4.3.2 Framework Overview

As visualized in Figure 4.2, our C2F aims to build a text classifier that can assign fine-grained labels to a set of coarsely-annotated documents based on only the label surface names and their hierarchical relations. In the absence of fine-grained human annotations, it uses fine-grained label surface names as weak-supervision signals and leverages a pre-trained language model as data generators. Following an iterative process, C2F fine-tunes a language model based on weak supervision. This fine-tuned language model is used to generate pseudo training data to train a fine-grained text classifier. Based on the classifier's predictions, we select highly probable samples for each fine-grained class and repeat this process for one more iteration by replacing weak supervision with these samples. This bootstrapping increases the quality of weak supervision by eliminating the mislabeled samples and improves the performance of text classifier as we show later in our case studies.

Our major innovations lie in how to better incorporate the label names and their hierarchical relations into the language model and therefore generate more high-quality psuedo training data. Our framework is compatible with any generative language model and we choose GPT-2 [63] in our implementation. We feed label names to the language model through a label-conditioned formulation. We further incorporate a regularization objective into the fine-tuning process that captures the constraints derived from the label hierarchy. The key components of C2F are discussed in detail in the following sections.

4.4 Initial Fine-grained Weak Supervision

We assume that user-provided label surface names are of high quality and are strong indicators for their respective classes, following the state-of-the-art weakly supervised text classification methods that only rely on label surface names [59, 58]. This assumption is intuitive and valid because we don't have any guidance other than class names from the user and we expect those to be of high quality and indicative of the categories.

Ideally, the posterior probability of a document belonging to a class after observing the presence of strong indicators should be close to 1. Therefore, we consider samples that *exclusively* contain the label surface name as its respective weak supervision. Mathematically, let $W(\mathscr{F}_j)$ denote weak supervision of fine-grained class $\mathscr{F}_j$:

$$W(\mathscr{F}_j) = \{\mathscr{D}_i | \mathscr{D}_i \cap f_{\downarrow}(c_i) = \{\mathscr{F}_j\}\}$$

where $\mathscr{D}_i \cap f_{\downarrow}(c_i)$ returns a set of fine-grained label names under the coarse-grained class c_i that appear in the docuemnt $\mathscr{D}_i$. When this set only contains $\mathscr{F}_j$, it means "exclusive" to other fine-grained labels. This "exclusiveness" could help us improve the precision of the initial weak supervision. Note that, it is implied that $\mathscr{F}_j \in f_{\downarrow}(c_i)$.

We observe that the initial weak supervision obtained usually has a highly-skewed label distribution, because the popularity of the label names varies. This difference in distribution could bias the generative language model towards majority label and might affect the quality of generated samples, which in turn, would affect the performance of text classifier. To address this problem, as there is no other prior knowledge, we opt to draw a balanced, weakly annotated subset through a stratified sampling before any model training. In other words, we make the size of weak supervision uniform for all samples equal to the minority label.

4.5 Tailored Language Model Training

In this section, we describe our label-conditioned, hierarchy-aware language model training formulations that facilitate conditional corpus generation. Specifically, we continuously train a pre-trained language model to capture the distribution $P(D|l)$, where D is a document and l is a (coarse or fine) label surface name. Thus, this model can generate pseudo-training documents for fine-grained labels, when we plug in fine-grained label surface names.

4.5.1 Label-Conditioned Generation

Before we describe our formulation, we briefly introduce GPT-2 and its pre-training objective.

GPT-2

GPT-2 is a large, pre-trained left-to-right language model which exhibits strong performance with minimal in-task fine-tuning on many generation tasks, such as dialog [64] and story generation [65]. Its strong zero-shot ability across tasks stems from its pre-training on the vast and diverse WebText corpus ($\approx$8M documents), besides the good inductive bias of its transformer-based architecture. GPT-2 is trained on standard language modeling objective to maximize the likelihood of a document D as follows:

$$\mathscr{L}(D) = \sum_i \log P(w_i | w_{i-1}, \ldots, w_1; \Theta)$$

where $P(\cdot)$ is modeled with a transformer-based architecture with parameters Θ.

To continuously train GPT-2 in a label-conditioned way, one has to maximize $P(D|l)$ instead of $P(D)$. We designate the label surface names as the special token sequences and append them in the beginning to their respective documents with another special token `<labelsep>` separating the label sequence and document. For example, a sample document "*Messi plays for FC Barcelona*" belonging to "*soccer*" is modified to "*soccer* `<labelsep>` *Messi plays for FC*

Barcelona". Therefore, our objective is to maximize $\mathscr{L}(D|l)$ defined below:

$$\mathscr{L}(D|l) = \sum_i \log P(w_i|w_{i-1},\ldots,w_1;l;\Theta)$$

Note that, the l here could be the label surface name of a coarse-grained or fine-grained class. One can view our formulation as asking the label token sequence to play the role of prompt and the document D to be the continuation, thus facilitating conditional corpus generation.

During the continuous training process, we have access to both the gold, coarse-grained labels and weak, fine-grained labels. Examples included in weak supervision give rise to two label-conditioned documents — one by prefixing with the coarse-grained, gold label and the other with the weak, fine-grained one (due to the "exclusiveness" in the initial weak supervision). Those not in the weakly supervised set only give rise to the first kind. Since there is no conflict between these two labels, we simply treat a document as belonging independently to either of them.

4.5.2 Hierarchy-Aware Regularization

Our label-conditioned generation treats both fine-grained and coarse-grained labels as prompts and does not use any information from the hierarchy. Therefore, we propose to add a regularization to the language model with constraints derived from hierarchy.

Intuitively, fine-grained labels are more specific to coarse-grained labels, and therefore, when generating the same document conditioned on its gold fine-grained label, it should have a higher probability than that conditioned on its coarse-grained label. We believe the same intuition is applicable to the high-quality weak supervision. Therefore, we seek to enforce the constraint while continuously training on weak supervision. Specifically, a document should be more likely given its fine-grained (weak) label rather than its coarse-grained label. Mathematically,

$$P(\mathscr{D}_i|\mathscr{F}_j) > P(\mathscr{D}_i|c_i), \forall \mathscr{D}_i \in W(\mathscr{F}_j)$$

where $\mathscr{W}(\mathscr{F}_j)$ is the weak supervision for fine-grained label $\mathscr{F}_j$. Note that, it is implied from $W(\mathscr{F}_j)$ that $\mathscr{F}_j \in f_{\downarrow}(c_i)$.

This inequality can be expressed in the form of a margin between $P(\mathscr{D}_i|\mathscr{F}_j)$ and $P(\mathscr{D}_i|c_i)$, which can be implemented in practice through an additional Hinge loss term:

$$HL(\mathscr{D}_i, \mathscr{F}_j) = \max(0, \log P(\mathscr{D}_i|c_i) - \log P(\mathscr{D}_i|\mathscr{F}_j) + \varepsilon)$$

where ε is a positive constant.

We incorporate this hierarchy-aware regularization into the final objective function as follows:

$$\begin{aligned}\mathscr{O} = &\sum_{\mathscr{D}_i \in \mathscr{D}} \mathscr{L}(D_i|c_i) \\ &+ \sum_{\mathscr{F}_j} \sum_{\mathscr{D}_i \in W(\mathscr{F}_j)} \mathscr{L}(\mathscr{D}_i|\mathscr{F}_j) - \lambda HL(\mathscr{D}_i, \mathscr{F}_j)\end{aligned}$$

The final optimization aims to maximize $\mathscr{O}$.

4.6 Pseudo Training Data Generation, Text Classifier, & Weak Supervision Update

After continuously training the language model in a label-conditioned way, we generate the data for each fine-grained category. Specifically, we send the corresponding label surface name as the prompt to our language model, and it then generates samples for that respective class. Since we don't know the label distribution beforehand, we assume it's a balanced distribution and thus, avoiding inducing potential bias in the classifier. We generate twice the required documents divided equally among fine-grained labels. Specifically, for a fine-grained label $\mathscr{F}_j \in f_{\downarrow}(c)$, we generate $2\frac{N_c}{|f_{\downarrow}(c)|}$ documents, where N_c is the number of documents that belong to coarse-grained label c.

We train a text classifier over these generated documents and their corresponding fine-grained labels. Our framework is compatible with any text classifier and we use BERT

Table 4.1. Dataset statistics

Dataset	$\|\mathscr{D}\|$	$\|\mathscr{C}\|$	$\|\mathscr{F}\|$	Coarse labels	Fine labels
NYT	11,744	5	26	arts, business, politics, science, sports	dance, music, movies, television, economy, energy companies, international business, stocks & bonds, abortion, federal budget, gay rights, gun control, immigration, law enforcement, military, surveillance, the affordable care act, cosmos, environment, baseball, basketball, football, golf, hockey, soccer, tennis
20News	16,468	5	17	computer, politics, recreation, religion, science	graphics, windows, ibm, mac, x window, mideast, guns, autos, motorcycles, baseball, hockey, christian, atheism, encryption, electronics, medicine, space

(`bert-base-uncased`) [66] classifier in our experiments.

After training the text classifier, we obtain fine-grained predictions and probability scores for all coarsely annotated documents $\mathscr{D}$. Finally, we bootstrap it on unlabelled data by *replacing* weak supervision $W(\mathscr{F}_j)$ by top-k predictions where $k = |W(\mathscr{F}_j)|$ in every fine-grained label $\mathscr{F}_j$ and repeat this process *one more time*. In our experiments, we observe that these top-$|W(\mathscr{F}_j)|$ predictions are of significantly higher quality than the initial weak supervision, thus improving the text classifier.

4.7 Experiments

In this section, we start with introducing datasets, compared methods, and experimental settings. And then, we present quantitative evaluation results of C2F together with all compared methods. In the end, we will show qualitative studies to analyze different aspects of our C2F framework.

4.7.1 Datasets

We evaluate our framework on two hierarchical datasets where each document has one coarse-grained label and one fine-grained label. The dataset statistics are provided in Table 4.1. The details of these datasets are as follows:

- **The New York Times (NYT):** Following the previous work [2, 3, 59] we experiment on the NYT dataset. It is a collection of news articles written and published by The New York Times. Each news article is classified into one of 5 coarse-grained genres (e.g., arts, sports) and 25 fine-grained categories (e.g., movies, music, baseball, football).

- **The 20 Newsgroups (20News):** The 20News dataset[1] 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). There are three miscellaneous labels (i.e., "misc.forsale", "talk.politics.misc", "talk.religion.misc"). As one can notice, their label names are about 'miscellaneous' and contain information of various types. Since these labels and label surface names have no focused meaning, we drop the documents annotated as these labels in our experiments.

4.7.2 Compared Methods

We compare our framework with a wide range of zero-shot and weakly supervised text classification methods described below:

- **Word2Vec** learns word vector representations [39] for all words in the corpus and consider the word vectors of label surface name vectors as their respective label representations. Each document is labeled with the most similar label based on cosine similarity.

- **WeSTClass** [2] assumes words and documents share a joint semantic space and model each class as a high-dimensional spherical distribution. Words are sampled from this learned distribution to create pseudo-training data over which a classification model is trained. This model is refined through self-training on unlabeled documents.

- **ConWea** [3] is a seed-driven contextualized weak supervision framework. They leverage pre-trained language models to resolve interpretation of seed words and make the weak supervision contextualized.

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

- **LOTClass** [58] uses pre-trained language model like BERT [66] to query replacements for class names and constructs a category vocabulary for each class. This is further used to fine-tune the language model on a word-level category prediction task and identifies potential classes for documents via string matching. A classifier is trained on this pseudo-labeled data with further self-training.

- **X-Class** [59] learns class-oriented document representations that make it adaptive to the user-specified classes. These document representations are aligned to classes through PCA + GMM, harvesting pseudo labels for a supervised classifier training.

We also compare **C2F** with its ablated variants. **C2F-NoHier** uses label-conditioned generation alone without the hierarchy-aware regularization. **C2F-Ind** and **C2F-Ind-NoHier** are run individually on each coarse-grained label c to assign a fine-grained label $\mathscr{F}_j \in f_{\downarrow}(c)$ and the predictions are accumulated at the end to compute aggregated results. However, **C2F-Ind** uses both label-conditioned generation with the hierarchy-aware regularization whereas **C2F-Ind-NoHier** uses label-conditioned generation alone.

For a fair comparison, we make coarse-grained annotated data available for all baselines and run them individually on each coarse-grained label c to assign a fine-grained label $\mathscr{F}_j \in f_{\downarrow}(c)$ and the predictions are accumulated at the end to compute aggregated results. We provide label surface names as seed words for seed-word-driven baselines like ConWea and WeSTClass.

We also present the performance of BERT in a supervised setting which is denoted as **BERT-Sup**. The results of BERT-Sup reported are on the test set which follows an 80-10-10 train-dev-test split.

4.7.3 Experimental Settings

While fine-tuning GPT-2, we experiment with learning rates $\alpha \in \{5e^{-5}, 5e^{-4}, 5e^{-6}\}$, with $\alpha = 5e^{-4}$ being found optimal, and continue the label-conditioned language model training for up to 5 epochs.

Table 4.2. Micro and Macro f1 scores and their respective standard deviations on two datasets are presented.

	NYT		**20 Newsgroup**	
Methods	Mi-F_1	Ma-F_1	Mi-F_1	Ma-F_1
Word2Vec	32.50 (2.50)	17.50 (1.50)	11.03 (1.30)	11.03 (0.90)
ConWea	76.23 (0.97)	69.82 (0.54)	56.14 (0.76)	56.21 (0.32)
WeSTClass	73.96 (0.49)	65.03 (0.31)	55.46 (0.19)	55.53 (0.38)
LOTClass	15.00 (1.20)	20.21 (0.76)	34.18 (0.64)	33.63 (0.71)
X-Class	91.16 (0.56)	81.09 (0.39)	73.15 (0.23)	73.06 (0.12)
C2F	**92.62 (0.54)**	**87.01 (0.72)**	**77.50 (0.96)**	**77.57 (0.89)**
C2FAblation	90.44 (0.91)	85.50 (0.82)	76.27 (0.85)	76.13 (0.78)
C2F-Ind	91.60 (0.45)	86.82 (0.44)	74.62(0.96)	74.50 (0.97)
C2F-Ind-NoHier	90.95 (0.59)	85.75 (0.17)	74.59 (0.63)	74.48 (0.57)
BERT-Sup	98.00 (0.27)	94.00 (0.57)	96.39 (0.43)	96.36 (0.72)

Generation from the model is done via nucleus sampling [67], with a budget of $p = 0.9$ and a length limit of 200 subwords. The prompt given for generation is simply the tag sequence corresponding to the intended fine-grained label of the sample to be generated. Since fine-grained class ratios are apriori unknown, an equal number of examples are sampled for each fine-grained class within the same coarse-grained class.

For the hierarchy-aware regularization, we set the hinge loss margin $\varepsilon = \log 5$ and $\lambda = 0.01$. For hyperparameter selection of ε, we sweep over the sequence of values in $\{logn\}_{n=1}^{n=10}$. Further searching is done through two levels of binary search. The decision to initially sweep over values in logarithmic fashion is taken based on two intuitions: i) Larger jumps were found to skip over the domain of variation of epsilon too quickly ii) ε is essentially a margin on logarithmic probabilities.

4.7.4 Quantitative Results

We evaluate our framework using Micro-f1(Mi-F_1) and Macro-f1(Ma-F_1) as performance metrics. The evaluation results of all methods run on three random seeds are summarized in Table 4.2 along with their respective standard deviations. We can observe that our proposed framework achieves superior performance compared to all other baselines. We discuss the effectiveness of C2F as follows:

- Our proposed framework demonstrates the best performance among all compared baselines. By utilizing the generative language model through label-conditioned fine-tuning and regularizing it with hierarchical hinge loss to leverage the hierarchy, it is able to generate good quality pseudo training data, which further helped in achieving the best performance.
- C2F outperforms X-Class, a zero shot classification framework with a significant margin. X-Class doesn't take advantage of label hierarchy and requires class names to be one word whereas our framework has no such limitation and leverages rich language models to understand informative label surface names.
- We have to note the significantly low performance of LOTClass. LOTClass queries replacements of label surface names and consider those to be indicative of the label. This is a valid assumption for the coarse-grained classification but when the classes become more and more fine-grained, the replacements may not be indicative of its respective class. For example, consider the sentence "I won a baseball game today". If "baseball" is replaced by "tennis", it is still a valid and meaningful statement but "tennis" is not indicative of "baseball". Therefore, LOTClass performs low in the fine-grained text classification task. Our framework separates the weak supervision for each label initially and fine-tunes the language model in a label conditioned way. Therefore, our framework is able to distinguish between fine-grained labels as well.
- The comparison between C2F and C2FAblation shows that the hierarchy hinge loss helped in leveraging the constraints from hierarchy to improve the language model.
- The comparison between C2F and C2F-Ind shows that the fine-grained classification benefits from the hierarchical structure and joint training with other coarse-grained classes.
- We observe that the performance of C2F is quite close to supervised method BERT-Sup, for example, on the NYT dataset. This demonstrates that C2F is quite effective in closing

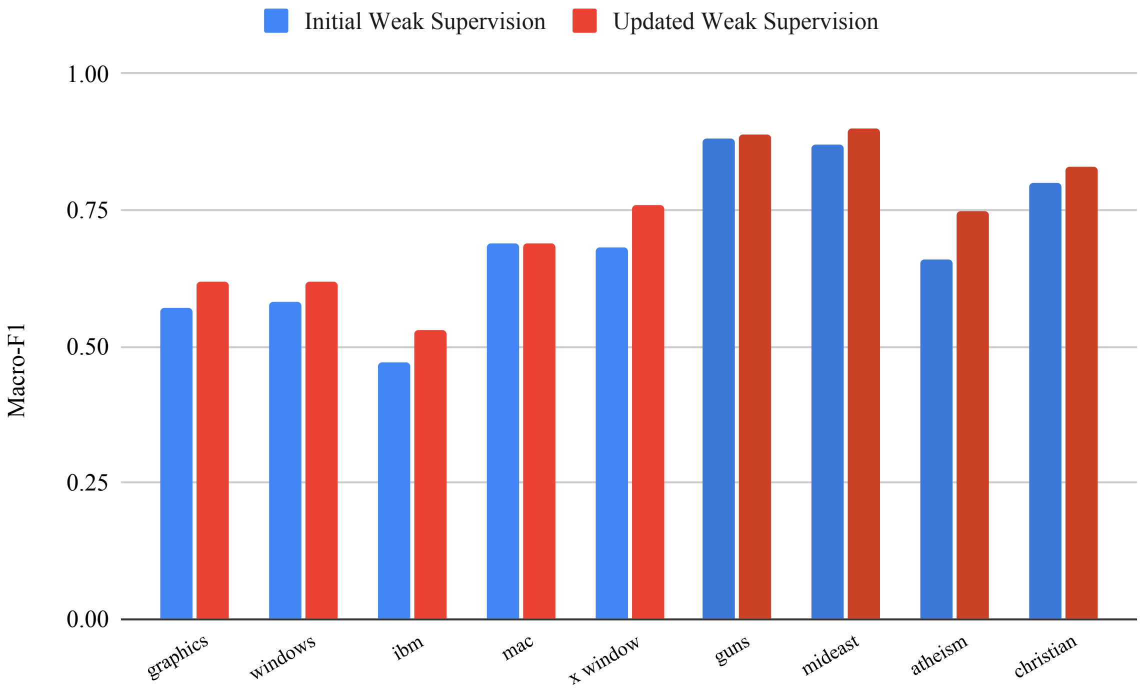


Figure 4.3. Performance increase in fine-grained text classifier from initial to updated weak supervision.

the performance gap between the weakly supervised and supervised setting with just label surface names as supervision.

4.7.5 Performance increase with bootstrapping

The f1-scores of fine-grained labels in three coarse-grained labels “computer”, “politics”, “religion” across iteration-0 and iteration-1 are plotted in Fig 4.3. We can observe that the performance has increased significantly from iteration-0 (in blue) to iteration-1 (in red). We attribute this increase to our bootstrapping where we select samples with high predicted probability and use it as weak supervision for the next iteration.

4.7.6 Sensitivity to ε

A potential concern with the experimental setup can be overtly high sensitivity of C2F to the hinge loss margin parameter, i.e ε. However, from the plot in Figure 4.4, we clearly see

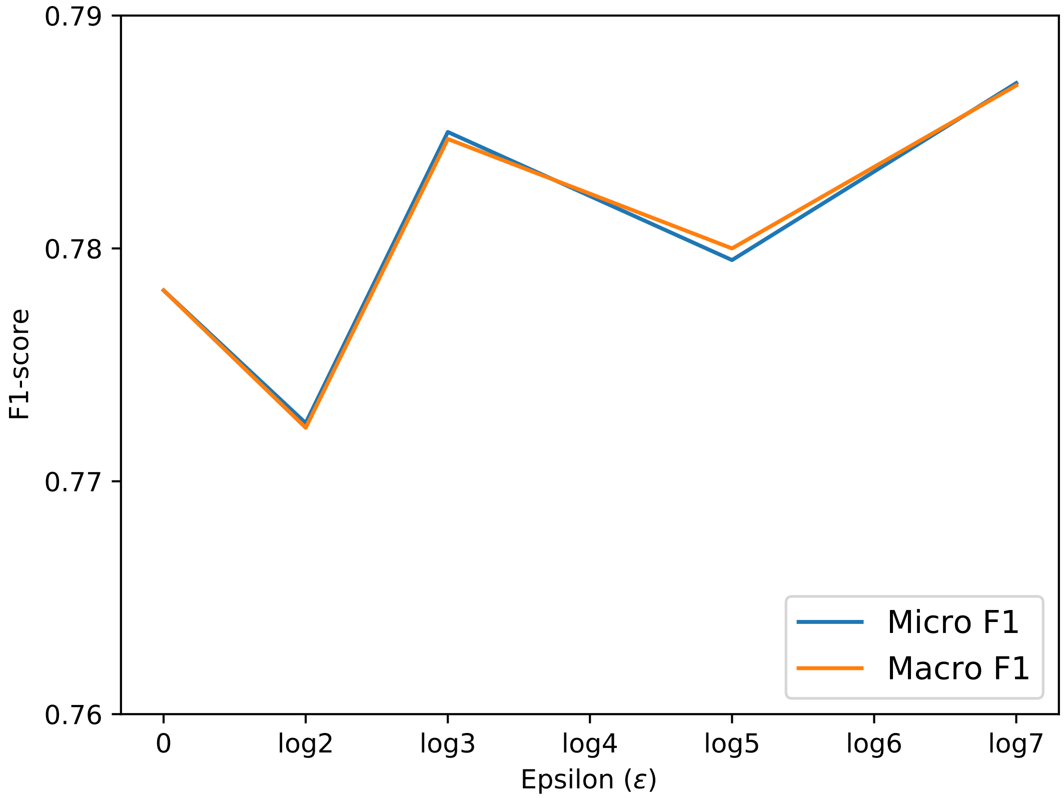


Figure 4.4. Micro and Macro-F1 scores vs ε on 20News.

that F1 scores aren't drastically sensitive to epsilon - with standard deviations of 0.00515 and 0.00517 for Macro and Micro-F1 scores respectively.

4.7.7 Stastical Significance Results

We perform a t-test between C2F and each of the other baselines on both datasets and the results are showed in Table 4.3. From these p-values, we can conclude that the performance improvement over baselines is significant.

4.7.8 Qualitative Analysis

Given a particular coarse label (say `sports`) and its data subsets $X = \{\mathscr{D}_i | c_i = \text{``}sports\text{''}\}$ and $X_f = \{\mathscr{D}_i | c_i = \text{``}sports\text{''}, f_i = f \in f_{\downarrow}(\text{``}sports\text{''})\}$, as a matter of post-hoc analysis, we can

Table 4.3. Statistical significance results.

Baseline	**p-value NYT**	**p-value 20News**
ConWea	8.34×10^{-131}	2.49×10^{-165}
WeSTClass	5.18×10^{-146}	1.97×10^{-166}
X-Class	6.45×10^{-71}	1.63×10^{-92}
C2F-NoHier	1.80×10^{-25}	2.33×10^{-55}
C2F-Ind	1.36×10^{-18}	7.92×10^{-110}
C2F-Ind-NoHier	3.46×10^{-24}	3.17×10^{-114}

compare three distinct "supervised" splits a classifier could've been trained on:

1. **Gold**: Data along with gold fine-grained labels, which is not actually available in our setting.

2. **C2F-Init**: This is the subset of X for which the initial weak supervision strategy assigns fine labels based on label surface names.

3. **C2F-Gen**: This is the data sampled from our trained language model as the generator for each of the respective fine-grained labels.

Which supervision is more apt from the purview of training? To answer this, we examine the entropy $H()$ of the word frequency distribution of the three datasets. Specifically, we examine reduction in value from entropy of the overall set $H(X)$ to the mean entropy on partitioning further by fine label, i.e $\bar{H}(X_f)$. Larger this drop, more internally coherent are the label partitions.

As we can see from Figure 4.5, the drops $H(X) - \bar{H}(X_f)$ are greater for C2F-GEN compared to both C2F-INIT and GOLD, indicating that it produces more mutually discriminative examples than both of them. At the same time, we observe that overall entropy of C2F-GEN, i.e $H(Gen) = 6.631$ does not drastically differ in value from, though it is significantly lesser than, that of GOLD, $H(Gold) = 6.924$, being 4.21% smaller. In summary, we see that C2F-GEN provides a more discriminative training signal without reducing example diversity.

Table 4.4 shows a few samples of generated documents for fine-grained labels in both NYT and 20Newsgroup datasets.

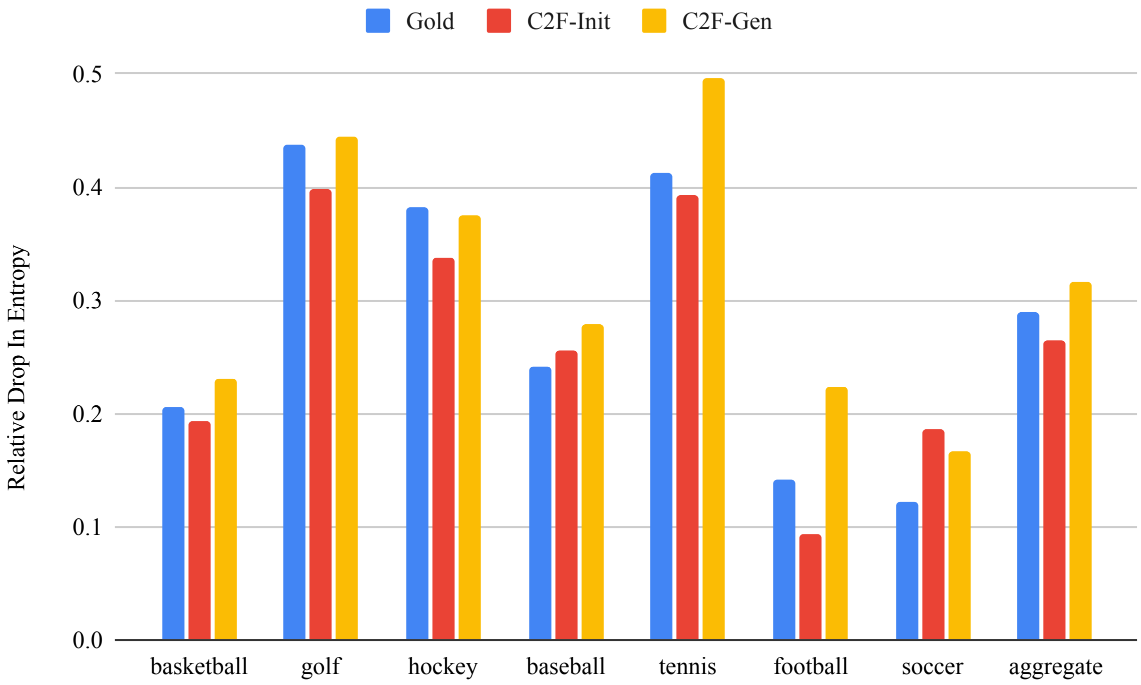


Figure 4.5. Relative drops in entropy $H(X) - H(X_{l_f})$ on splitting by fine label l_f, $\forall l_f \in f_{\downarrow}(\text{SPORTS})$, along with their aggregate. *C2F-Gen, C2F-Init, Gold* stands for the generator sampled, initial weakly supervised subset and the entire ground truth datasets respectively.

4.8 Summary

In this chapter, we introduced the task of coarse-to-fine grained classification and laid out its significance. Next, we showed the promise of incorporating large pre-trained LMs like GPT-2 into a weak supervision strategy which starts out with just label surface names. Finally, we showed a way to attune these models for our task even better, through explicit regularization based on coarse-fine label constraints which fall naturally out of our task definition. We outperform multiple SOTA zero-shot baselines on NYT and 20News, underscoring utility both of incorporating pre-trained LMs as well as task constraints. We believe exploring newer ways of exploiting task agnostic knowledge sources and injecting task constraints into the weakly supervised learning process are promising avenues for future work.

Table 4.4. Example generated samples for fine-grained labels *hockey*, *basketball*, *cosmos*, and *economy* in NYT dataset and *autos*, *atheism*, *windows* in 20Newsgroup dataset.

Label	Sentence
hockey	the rangers' two injury-riddled offense and seven-game effort to defeat the florida panthers on saturday have prompted coach alain vigneault to call on the veteran right wing to start against the capitals. his message to the rookie goalie thomas vanek was that "he was fine today" and the rangers did not have to worry about his status in a game against another talented team.
basketball	the nets entered the n.b.a. season with such wide open space that even their most ardent supporters of the franchise might admit their disappointment. the season ended sunday with the los angeles lakers selecting the first-round draft pick, with the eventual no. 1 pick jason kidd, a dynamic penetrator who could challenge for a starting job, at least in a year.
cosmos	washington — the outermost reaches of space are barely perceptible from the surface, making it difficult to see the red planet, but a spacecraft based on this data could find something as yet elusive: an elusive galaxy that has just vanished.on thursday, kepler spacecraft, which is carrying back-to-back samples of solar starlight, will begin a journey that will lead the space agencyś curiosity.
economy	washington — the number of americans seeking unemployment benefits fell less than expected last week, the latest evidence that steady job growth is holding steady. jobless claims rose,000 to a seasonally adjusted,000, the labor department said on thursday. that is 146 more than previously reported.last week's drop in jobless claims came two months after steady increases in the previous week, when claims rose.
autos	re manual shift bigots in article (james bruder) writes another question that hasn't been addressed yet is how come the auto mated to the lever controlling selector is not mounted to the transmission? i would think that the mated to the shifter would be mounted in the passenger compartment. is this a problem with the manual transmission? and if so, is it a problem with the shifter's mounting point?
atheism	re why do people become atheists? in article, (kent sandvik) writes in article, (robert beauchaine) writes and i suppose i would have better evidence if i could. why would it be any different, for one thing? iḿ fairly new to this group, so perhaps this sort of question has already been asked, and answered before. but iv́e just started to think about it.
windows	re dos 6.0 in article 1qh61o, (russ sharp) wrote it's absolutely ludicrous for me to try and run dos 6.0 without the bloody help of at least 8 people. i've tried compiling it on several systems, and i've run it six times without a problem. dos 6.0 didn't mention a config.sys or anything else. there were a couple other windows' manuals which did mention about config.sys.

Chapter 4, in part is currently being prepared for submission for publication of the material. Mekala, Dheeraj; Gangal, Varun; Shang, Jingbo. The dissertation/thesis author was the primary investigator and author of this material.

Chapter 5

Future Work

In this thesis, we studied a family of weakly supervised text classification approaches to minimize the human effort in manual annotations. We first discussed contextualizing weak supervision to identify the interpretation of user-provided seed words. Then, we described leveraging widely-available metadata information for text classification. Finally, we introduced a new problem called coarse-to-fine grained classification that aims to perform fine-grained classification on coarsely annotated data without any fine-grained annotations.

In the future, it will be interesting to expand our study to multi-lingual and code-switched data. And also, these frameworks can be extended to integrate different forms of supervision including annotated documents. Finally, we believe that our study can inspire many real life applications such as automatic file destination recommender system that suggests a destination path for every file that is downloaded from web using its metadata and file hierarchy.

Bibliography

[1] F. Tao, C. Zhang, X. Chen, M. Jiang, T. Hanratty, L. Kaplan, and J. Han, "Doc2cube: Automated document allocation to text cube via dimension-aware joint embedding," *Dimension*, vol. 2016, p. 2017, 2015.

[2] Y. Meng, J. Shen, C. Zhang, and J. Han, "Weakly-supervised neural text classification," in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pp. 983–992, ACM, 2018.

[3] D. Mekala and J. Shang, "Contextualized weak supervision for text classification," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 323–333, 2020.

[4] D. Mekala, X. Zhang, and J. Shang, "Meta: Metadata-empowered weak supervision for text classification," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 8351–8361, 2020.

[5] E. Agichtein and L. Gravano, "Snowball: Extracting relations from large plain-text collections," in *Proceedings of the fifth ACM conference on Digital libraries*, pp. 85–94, ACM, 2000.

[6] E. Riloff, J. Wiebe, and T. Wilson, "Learning subjective nouns using extraction pattern bootstrapping," in *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*, pp. 25–32, Association for Computational Linguistics, 2003.

[7] B. J. Kuipers, P. Beeson, J. Modayil, and J. Provost, "Bootstrap learning of foundational representations," *Connection Science*, vol. 18, no. 2, pp. 145–158, 2006.

[8] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," in *Proceedings of NAACL-HLT*, pp. 2227–2237, 2018.

[9] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "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)*, pp. 4171–4186, 2019.

[10] J. Tang, M. Qu, and Q. Mei, “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*, pp. 1165–1174, ACM, 2015.

[11] T. Miyato, A. M. Dai, and I. Goodfellow, “Adversarial training methods for semi-supervised text classification,” *arXiv:1605.07725*, 2016.

[12] W. Xu, H. Sun, C. Deng, and Y. Tan, “Variational autoencoder for semi-supervised text classification,” in *AAAI*, 2017.

[13] Y. Song and D. Roth, “On dataless hierarchical text classification,” in *AAAI*, 2014.

[14] K. Li, H. Zha, Y. Su, and X. Yan, “Unsupervised neural categorization for scientific publications,” in *SIAM Data Mining*, pp. 37–45, SIAM, 2018.

[15] Y. Meng, J. Shen, C. Zhang, and J. Han, “Weakly-supervised hierarchical text classification,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 6826–6833, 2019.

[16] B. McCann, J. Bradbury, C. Xiong, and R. Socher, “Learned in translation: Contextualized word vectors,” in *Advances in Neural Information Processing Systems*, pp. 6294–6305, 2017.

[17] J. Howard and S. Ruder, “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)*, vol. 1, pp. 328–339, 2018.

[18] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, “Improving language understanding by generative pre-training,” 2018.

[19] L. Liu, J. Shang, X. Ren, F. F. Xu, H. Gui, J. Peng, and J. Han, “Empower sequence labeling with task-aware neural language model,” in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[20] A. Akbik, D. Blythe, and R. Vollgraf, “Contextual string embeddings for sequence labeling,” in *Proceedings of the 27th International Conference on Computational Linguistics*, pp. 1638–1649, 2018.

[21] M. Lesk, “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*, pp. 24–26, 1986.

[22] Z. Zhong and H. T. Ng, “It makes sense: A wide-coverage word sense disambiguation system for free text,” in *Proceedings of the ACL 2010 system demonstrations*, pp. 78–83, 2010.

[23] D. Yuan, J. Richardson, R. Doherty, C. Evans, and E. Altendorf, “Semi-supervised word sense disambiguation with neural models,” *arXiv preprint arXiv:1603.07012*, 2016.

[24] A. Raganato, J. Camacho-Collados, and R. Navigli, "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*, pp. 99–110, 2017.

[25] M. Le, M. Postma, J. Urbani, and P. Vossen, "A deep dive into word sense disambiguation with lstm," in *Proceedings of the 27th international conference on computational linguistics*, pp. 354–365, 2018.

[26] R. Tripodi and R. Navigli, "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)*, pp. 88–99, 2019.

[27] J. Li and D. Jurafsky, "Do multi-sense embeddings improve natural language understanding?," *arXiv preprint arXiv:1506.01070*, 2015.

[28] D. Mekala, V. Gupta, B. Paranjape, and H. Karnick, "SCDV : Sparse composite document vectors using soft clustering over distributional representations," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, (Copenhagen, Denmark), pp. 659–669, Association for Computational Linguistics, Sept. 2017.

[29] V. Gupta, A. Saw, P. Nokhiz, H. Gupta, and P. Talukdar, "Improving document classification with multi-sense embeddings," *arXiv preprint arXiv:1911.07918*, 2019.

[30] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy, "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*, pp. 1480–1489, 2016.

[31] Y. Kim, "Convolutional neural networks for sentence classification," *arXiv preprint arXiv:1408.5882*, 2014.

[32] X. Zhang, J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," in *Advances in neural information processing systems*, pp. 649–657, 2015.

[33] S. Lai, L. Xu, K. Liu, and J. Zhao, "Recurrent convolutional neural networks for text classification," in *Twenty-ninth AAAI conference on artificial intelligence*, 2015.

[34] J. Liu, J. Shang, C. Wang, X. Ren, and J. Han, "Mining quality phrases from massive text corpora," in *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, pp. 1729–1744, 2015.

[35] J. Shang, J. Liu, M. Jiang, X. Ren, C. R. Voss, and J. Han, "Automated phrase mining from massive text corpora," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 10, pp. 1825–1837, 2018.

[36] A. K. Jain and R. C. Dubes, “Algorithms for clustering data,” *Englewood Cliffs: Prentice Hall, 1988*, 1988.

[37] L. v. d. Maaten and G. Hinton, “Visualizing data using t-sne,” *Journal of machine learning research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

[38] M.-W. Chang, L.-A. Ratinov, D. Roth, and V. Srikumar, “Importance of semantic representation: Dataless classification.,” in *Aaai*, vol. 2, pp. 830–835, 2008.

[39] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” *arXiv preprint arXiv:1301.3781*, 2013.

[40] A. R. Benson, D. F. Gleich, and J. Leskovec, “Higher-order organization of complex networks,” *Science*, vol. 353, no. 6295, pp. 163–166, 2016.

[41] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon, “Network motifs: simple building blocks of complex networks,” *Science*, vol. 298, no. 5594, pp. 824–827, 2002.

[42] J. Shang, X. Zhang, L. Liu, S. Li, and J. Han, “Nettaxo: Automated topic taxonomy construction from text-rich network,” in *Proceedings of The Web Conference 2020*, 2020.

[43] Y. Sun, J. Han, X. Yan, P. S. Yu, and T. Wu, “Pathsim: Meta path-based top-k similarity search in heterogeneous information networks,” *Proceedings of the VLDB Endowment*, vol. 4, no. 11, pp. 992–1003, 2011.

[44] D. Tang, B. Qin, and T. Liu, “Learning semantic representations of users and products for document level sentiment classification,” in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 1014–1023, 2015.

[45] H. Chen, M. Sun, C. Tu, Y. Lin, and Z. Liu, “Neural sentiment classification with user and product attention,” in *Proceedings of the 2016 conference on empirical methods in natural language processing*, pp. 1650–1659, 2016.

[46] M. Rosen-Zvi, T. Griffiths, M. Steyvers, and P. Smyth, “The author-topic model for authors and documents,” *arXiv preprint arXiv:1207.4169*, 2012.

[47] Y. Zhang, W. Wei, B. Huang, K. M. Carley, and Y. Zhang, “Rate: Overcoming noise and sparsity of textual features in real-time location estimation,” in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pp. 2423–2426, 2017.

[48] Y. Dong, N. V. Chawla, and A. Swami, “metapath2vec: Scalable representation learning for heterogeneous networks,” in *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 135–144, 2017.

[49] J. Shang, M. Qu, J. Liu, L. M. Kaplan, J. Han, and J. Peng, “Meta-path guided embedding for similarity search in large-scale heterogeneous information networks,” *arXiv preprint arXiv:1610.09769*, 2016.

[50] Y. Zhang, Y. Meng, J. Huang, F. F. Xu, X. Wang, and J. Han, "Minimally supervised categorization of text with metadata," *arXiv preprint arXiv:2005.00624*, 2020.

[51] R. Johnson and T. Zhang, "Effective use of word order for text categorization with convolutional neural networks," *arXiv preprint arXiv:1412.1058*, 2014.

[52] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Y. Ng, and C. Potts, "Recursive deep models for semantic compositionality over a sentiment treebank," in *Proceedings of the 2013 conference on empirical methods in natural language processing*, pp. 1631–1642, 2013.

[53] A. Hensley, A. Doboli, R. Mangoubi, and S. Doboli, "Generalized label propagation," in *2015 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, 2015.

[54] J. Tang, J. Zhang, L. Yao, J. Li, L. Zhang, and Z. Su, "Arnetminer: Extraction and mining of academic social networks," in *KDD'08*, pp. 990–998, 2008.

[55] M. Wan and J. J. McAuley, "Item recommendation on monotonic behavior chains," in *Proceedings of the 12th ACM Conference on Recommender Systems, RecSys 2018, Vancouver, BC, Canada, October 2-7, 2018* (S. Pera, M. D. Ekstrand, X. Amatriain, and J. O'Donovan, eds.), pp. 86–94, ACM, 2018.

[56] M. Wan, R. Misra, N. Nakashole, and J. J. McAuley, "Fine-grained spoiler detection from large-scale review corpora," in *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers* (A. Korhonen, D. R. Traum, and L. Màrquez, eds.), pp. 2605–2610, Association for Computational Linguistics, 2019.

[57] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, pp. 3111–3119, 2013.

[58] Y. Meng, Y. Zhang, J. Huang, C. Xiong, H. Ji, C. Zhang, and J. Han, "Text classification using label names only: A language model self-training approach," *arXiv preprint arXiv:2010.07245*, 2020.

[59] Z. Wang, D. Mekala, and J. Shang, "X-class: Text classification with extremely weak supervision," *arXiv preprint arXiv:2010.12794*, 2020.

[60] F. Tao, C. Zhang, X. Chen, M. Jiang, T. Hanratty, L. Kaplan, and J. Han, "Doc2cube: Allocating documents to text cube without labeled data," in *2018 IEEE International Conference on Data Mining (ICDM)*, pp. 1260–1265, IEEE, 2018.

[61] Y. Zhang, X. Chen, Y. Meng, and J. Han, "Hierarchical metadata-aware document categorization under weak supervision," in *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, pp. 770–778, 2021.

[62] C.-Y. Hsieh, M. Xu, G. Niu, H.-T. Lin, and M. Sugiyama, “A pseudo-label method for coarse-to-fine multi-label learning with limited supervision,” 2019.

[63] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, “Language models are unsupervised multitask learners,” *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.

[64] Y. Zhang, S. Sun, M. Galley, Y.-C. Chen, C. Brockett, X. Gao, J. Gao, J. Liu, and B. Dolan, “Dialogpt: Large-scale generative pre-training for conversational response generation,” *arXiv preprint arXiv:1911.00536*, 2019.

[65] A. See, A. Pappu, R. Saxena, A. Yerukola, and C. D. Manning, “Do massively pretrained language models make better storytellers?,” in *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pp. 843–861, 2019.

[66] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” *arXiv preprint arXiv:1810.04805*, 2018.

[67] A. Holtzman, J. Buys, L. Du, M. Forbes, and Y. Choi, “The curious case of neural text degeneration,” *arXiv preprint arXiv:1904.09751*, 2019.