Learning Only from Relevant Keywords and Unlabeled Documents

Nontawat Charoenphakdee[1,3] **Jongyeong Lee**[1,3] **Yiping Jin**[2]
Dittaya Wanvarie[2] **Masashi Sugiyama**[3,4,1]
[1]Department of Computer Science, The University of Tokyo
[2]Department of Mathematics and Computer Science, Faculty of Science, Chulalongkorn University
[3]RIKEN Center for Advanced Intelligence Project
[4]Department of Complexity Science and Engineering, The University of Tokyo
{nontawat, lee}@ms.k.u-tokyo.ac.jp
jinyiping@knorex.com.
Dittaya.W@chula.ac.th
sugi@k.u-tokyo.ac.jp

Abstract

We consider a document classification problem where document labels are absent but only relevant keywords of a target class and unlabeled documents are given. Although heuristic methods based on pseudo-labeling have been considered, theoretical understanding of this problem has still been limited. Moreover, previous methods cannot easily incorporate well-developed techniques in supervised text classification. In this paper, we propose a theoretically guaranteed learning framework that is simple to implement and has flexible choices of models, e.g., linear models or neural networks. We demonstrate how to optimize the area under the receiver operating characteristic curve (AUC) effectively and also discuss how to adjust it to optimize other well-known evaluation metrics such as the accuracy and F$_1$-measure. Finally, we show the effectiveness of our framework using benchmark datasets.

1 Introduction

Supervised text classification is a traditional problem in natural language processing that has been studied extensively (Nigam et al., 2000; Sebastiani, 2002; Forman, 2003; Joulin et al., 2016). In this problem, we are given a set of labeled text documents and the goal is to construct a classifier that can classify an unseen document effectively. There are many useful applications for text classification, e.g., sentiment classification (Liu and Zhang, 2012; Medhat et al., 2014), biomedical text mining (Cohen and Hersh, 2005; Huang and Lu, 2015), and social media monitoring (Zeng et al., 2010; Hu and Liu, 2012).

In the real-world, it is impractical to expect that labeled data can always be obtained abundantly. For example, given big data of unlabeled texts, it can be very time-consuming and costly for the labeling task so that we can apply a supervised text classification method. Another example is when the label information is protected due to privacy concerns. These bottlenecks motivate researches on weakly-supervised learning (Zhou, 2017), which focuses on devising a machine learning method that can learn effectively although a lot of labeled data are not accessible.

An alternative is to provide keywords as a hint for classifying a document to a target class. Intuitively, this approach can be much cheaper when the number of documents is huge since the number of keywords does not necessarily grow linearly with the number of documents. *Dataless classification* is a text classification problem where we are given keywords for each class and unlabeled documents (Chang et al., 2008; Song and Roth, 2014; Chen et al., 2015; Li and Yang, 2018). Note that we do not have access to *any* labeled documents.

In this paper, we investigate dataless classification in the situation where only the keywords of the target class are given. We regard documents of a class that we have relevant keywords as positive and others as negative. Unlike dataless classification, we have access to only the keywords of the positive class. This problem setting can be considered more difficult than the traditional one since one can always reduce dataless classification to this problem by ignoring negative keywords. This scenario is also highly relevant when the information about negative classes cannot be explicitly described, e.g., in the information retrieval task, we may only have information about the target class of interest.

This problem setting has already been considered by Jin et al. (2017), where they called it *lightly-supervised one-class classification*. To the best of our knowledge, existing work in dataless classification and lightly-supervised one-class classification neither provide a theoretical guarantee, nor have flexible choices of models and opti-

Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, pages 3993–4002,
Hong Kong, China, November 3–7, 2019. ©2019 Association for Computational Linguistics

mization algorithms for this problem.

The goal of this paper is to formalize lightly-supervised one-class classification and develop a reliable and flexible framework to handle this problem effectively. To achieve this goal, we propose a framework that has a theoretical guarantee and allows practitioners to have flexible choices to maximize the performance, e.g., neural network architectures such as convolutional neural networks (Zhang et al., 2015) or recurrent neural networks (Lai et al., 2015). They can also pick an optimization method that is suitable for their model such as Adam (Kingma and Ba, 2014) or AMSGrad (Reddi et al., 2018). These advantages allow us to utilize well-developed supervised-learning text classification methods in our framework.

We point out that the problem of lightly-supervised one-class classification is highly related to binary classification from corrupted labels. Intuitively, the common idea to solve this problem is to use keywords to pseudo-label given documents then performs classification (Jin et al., 2017). However, pseudo-positive and pseudo-negative labels are unreliable since they can be incorrectly labeled. Thus, the learning method should be robust against label corruption. In our framework, we also use relevant keywords to split unlabeled data into two sets. Then, the key idea is to employ a method based on an empirical risk minimization framework (Vapnik, 1998) with a loss function that benefits from a symmetric condition to maximize AUC (Charoenphakdee et al., 2019). We justify the soundness of our method by proving that our proposed framework gives a consistent estimation of the true expected AUC risk as long as the divided two sets have different proportions of positive data. We elucidate this theoretical result by providing an estimation error bound for AUC maximization using a Rademacher complexity measure. Furthermore, we discuss how to adjust our method to optimize other evaluation metrics, which are the accuracy and F$_1$-measure. We also illustrate that label corruption may cause a classifier to pick a wrong decision boundary and suggest to adjust the threshold on the basis of the proportion of positive data in unlabeled data.

2 Preliminaries

In this section, we introduce the notation, review AUC, and an empirical risk minimization framework for optimizing AUC.

Table 1: Examples of loss functions. A symmetric loss is a loss function such that $\ell(z) + \ell(-z) = K$, where K is a positive constant.

Loss name	$\ell(z)$	Symmetric
Logistic	$\log(1 + e^{-z})$	×
Squared	$(1 - z)^2$	×
Zero-one	$-\frac{1}{2}\text{sign}(z) + \frac{1}{2}$	✓
Sigmoid	$\frac{1}{1+\exp(z)}$	✓

2.1 Notation

Let $\boldsymbol{x} \in \mathcal{X}$ be a pattern in the input space $\mathcal{X}$ and $y \in \{-1, 1\}$ be a label. We denote $g : \mathcal{X} \to \mathbb{R}$ as a prediction function. Let $\mathbb{E}_{\mathrm{P}}[\cdot]$ and $\mathbb{E}_{\mathrm{N}}[\cdot]$ denote the expectation over the class-conditional probability $p(\boldsymbol{x}|y = 1)$ and $p(\boldsymbol{x}|y = -1)$, respectively. Moreover, let π denote a class prior $p(y = 1)$, i.e., the proportion of positive data.

Loss functions that we will discuss in this paper are in the family of margin losses, where a loss receives one argument $\ell : \mathbb{R} \to \mathbb{R}$. This loss family covers a lot of well-known losses (Bartlett et al., 2006), e.g., the logistic loss, squared loss, and hinge loss. We define the zero-one loss as $\ell_{0\text{-}1}(z) = -\frac{1}{2}\text{sign}(z) + \frac{1}{2}$, where $\text{sign}(g(\boldsymbol{x})) = 1$ if $g(\boldsymbol{x}) > 0$, -1 if $g(\boldsymbol{x}) < 0$, and 0 otherwise. Finally, we consider a symmetric loss as a margin loss ℓ_{sym} that satisfies the symmetric condition $\ell_{\text{sym}}(z) + \ell_{\text{sym}}(-z) = K$, where K is a positive constant. Examples of such losses are the zero-one loss and sigmoid loss (Ghosh et al., 2015). Table 1 provides examples of margin losses in binary classification.

2.2 Area under the Receiver Operating Characteristic Curve (AUC)

AUC is an evaluation metric for a bipartite ranking task (Cortes and Mohri, 2004; Menon and Williamson, 2016), where we want to find a function that outputs a higher value for positive data than negative data. Moreover, AUC is also a popular metric for a classifier under class imbalance (Menon et al., 2015). It is also known in the literature of statistics as the Wilcoxon-Mann-Whitney statistic (Mann and Whitney, 1947; Hanley and McNeil, 1982).

Let us consider the AUC risk, i.e., *bipartite ranking risk* (Narasimhan and Agarwal, 2013):

$$R^{\ell_{0\text{-}1}}_{\text{AUC}}(g) = \mathbb{E}_{\mathrm{P}}\left[\mathbb{E}_{\mathrm{N}}\left[\ell_{0\text{-}1}(g(\boldsymbol{x}^{\mathrm{P}}_i) - g(\boldsymbol{x}^{\mathrm{N}}_j))\right]\right]. \quad (1)$$

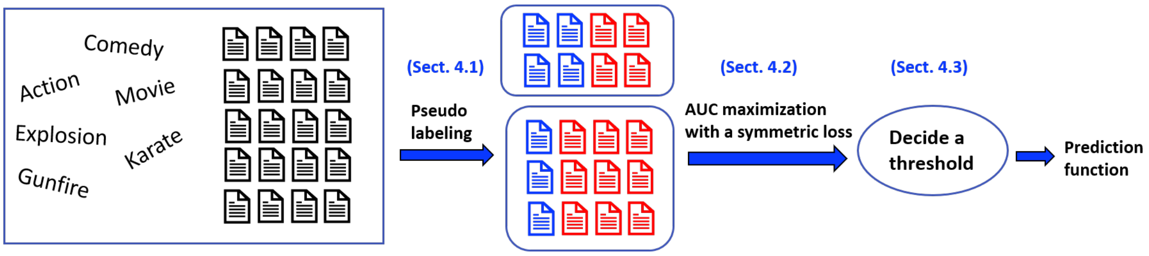

Figure 1: An overview of the framework. Blue documents indicates clean positive data and red documents denote clean negative data in the two sets of data divided by the pseudo-labeling algorithm. Note that labels are not observed by the framework. Sect. denotes a section that describes the procedure for each step.

Then, AUC is defined as $\mathrm{AUC}(g) = 1 - R^{\ell_{0\text{-}1}}_{\mathrm{AUC}}(g)$. It is important to note that unlike many other evaluation metrics, we can evaluate AUC without deciding the threshold for a function g. In this paper, we will show that optimizing AUC in our problem setting has a great advantage when no information about the threshold is given since we only have keywords and unlabeled documents.

2.3 Empirical Risk Minimization (ERM) for AUC Maximization

Here, we review a widely used framework in machine learning called the ERM framework (Vapnik, 1998) for AUC maximization.

The goal of AUC maximization is to minimize the AUC risk in (1) with respect to the zero-one loss $R^{\ell_{0\text{-}1}}_{\mathrm{AUC}}(g)$ (Menon and Williamson, 2016).

Although the goal is to minimize the bipartite ranking risk with respect to the zero-one loss $\ell_{0\text{-}1}$, a surrogate loss ℓ, e.g., the logistic loss or the squared loss, is instead applied in practice since minimizing $\ell_{0\text{-}1}$ is computationally infeasible (Ben-David et al., 2003; Zhang, 2004; Bartlett et al., 2006; Feldman et al., 2012). Note that we cannot directly minimize the AUC risk in our setting since we do not have access to the positive and negative distributions.

To explain the ERM framework for bipartite ranking, in this section, let us assume that we are given positive data $\{\boldsymbol{x}^{\mathrm{P}}_i\}_{i=1}^{n_{\mathrm{P}}}$ and negative data $\{\boldsymbol{x}^{\mathrm{N}}_j\}_{j=1}^{n_{\mathrm{N}}}$ drawn from the class-conditional probability densities $p(\boldsymbol{x}|y=1)$ and $p(\boldsymbol{x}|y=-1)$, respectively. Having access to positive and negative data, the ERM framework suggests us to minimize the following empirical risk (Yan et al., 2003; Cortes and Mohri, 2004):

$$\hat{R}^{\ell}_{\mathrm{AUC}}(g) = \frac{1}{n_{\mathrm{P}} n_{\mathrm{N}}} \sum_{i=1}^{n_{\mathrm{P}}} \sum_{j=1}^{n_{\mathrm{N}}} \ell(g(\boldsymbol{x}^{\mathrm{P}}_i) - g(\boldsymbol{x}^{\mathrm{N}}_j)). \tag{2}$$

With the given training data and a surrogate loss ℓ, we can minimize $\hat{R}^{\ell}_{\mathrm{AUC}}(g)$. This risk estimator $\hat{R}^{\ell}_{\mathrm{AUC}}(g)$ is an unbiased and consistent estimator of the true AUC risk (1), which means it converges to the true AUC risk as the number of data increases (Yan et al., 2003; Herschtal and Raskutti, 2004; Gao and Zhou, 2015).

Note that in practice, it is common to apply a regularization method for controlling the bias-variance trade-off of the estimator to avoid overfitting. In this paper, we adopt the ERM framework for minimizing the AUC risk. Note that one challenge of our problem is that we cannot apply the standard AUC risk directly since we have no access to any labeled data.

3 Problem Formulation

In this section, we formulate the problem of learning from keywords and unlabeled data. We are given a set of relevant keywords $W := \{w_j\}_{j=1}^{n_{\mathrm{W}}}$, and unlabeled documents drawn from the following distribution:

$$X_{\mathrm{U}} := \{\boldsymbol{x}^{\mathrm{U}}_i\}_{i=1}^{n_{\mathrm{U}}} \overset{\text{i.i.d.}}{\sim} p_{\pi}(\boldsymbol{x}),$$

where

$$p_{\pi}(\boldsymbol{x}) = \pi p(\boldsymbol{x}|y=1) + (1-\pi)\, p(\boldsymbol{x}|y=-1).$$

The evaluation metric defines the goal of this problem. For example, if we want a ranking function that outputs a large value for positive data, and small otherwise, the well-known evaluation metric is AUC (Agarwal et al., 2005). We will discuss other evaluation metrics, which are the accuracy and F_1-measure in Section 4.3.

4 Proposed Framework

In this section, we propose a novel framework for handling this problem systematically. The framework consists of three parts. First, we use relevant keywords to extract pseudo-positive data from the unlabeled data. Then, we employ a symmetric loss function for AUC optimization within the empirical risk minimization framework. Finally, we adjust the threshold of a trained AUC maximizer to further optimize other evaluation metrics. Figure 1 shows an overview of our framework.

4.1 Pseudo-labeling

Here, we propose a simple pseudo-labeling method. We want to emphasize that we do not expect that a pseudo-labeling algorithm is perfect, i.e., pseudo-positive documents may contain negative documents. In Section 4.2, we show that it is possible to maximize AUC effectively even though the labels are highly corrupted.

Since we focus on proposing a general framework that can be used in a wide range of text classification tasks, we use a simple pseudo-labeling method. In practice, one can incorporate prior knowledge such as the specific characteristic of documents to develop a sophisticated and potentially better pseudo-labeling algorithm to further enhance the performance. Our simple pseudo-labeling method is a ranking score based on the document similarities with the cosine similarity between unlabeled documents and keywords. Although we assume some keywords are given, it is hard to guarantee that the given keywords are sufficient to learn effectively. Hence, it is reasonable to sample more similar words to given keywords. In this paper, we used the GloVe model to find similar words (Pennington et al., 2014). With a sufficient number of keywords, we calculate the cosine similarity between the set of extended keywords set and each unlabeled document by simply merging all keywords and extended keywords into one document and measure the similarity between this keyword document and each unlabeled document. Finally, we pseudo-label documents to positive if their cosine similarities are in the top-ϕ%.

4.2 AUC Optimization from Corrupted Labels

After applying Algorithm 1, we can obtain two sets of data, which we call them a corrupted positive set and a corrupted negative set. Let us define θ, $\theta' \in [0, 1]$, which indicate the proportion of positive data for the corrupted positive and negative sets, respectively. We assume a mild condition that our pseudo-labeling method can split the data such that the corrupted positive set has a higher proportion of positive data than another set, i.e., $\theta > \theta'$. Then, we formulate the data from the two sets as

Algorithm 1: A pseudo-labeling algorithm

```
Input: Unlabeled documents $\{x_i^{\mathrm{U}}\}_{i=1}^{n_{\mathrm{U}}}$,
       keyword set $W := \{w_j\}_{j=1}^{n_{\mathrm{W}}}$, threshold
       $\phi \in [0, 100]$, weight of original
       keywords $\alpha \in \mathbb{N}$, sampling factor
       $\gamma \in \mathbb{N}$.
Output: two sets of pseudo-labeled
        documents $X_{\mathrm{CP}}$ and $X_{\mathrm{CN}}$
$X_{\mathrm{CP}} := \{\}$; $X_{\mathrm{CN}} := \{\}$
foreach $w_j$ do
    Add additional ($\alpha$-1) of $w_j$ to $W$;
    Find top-$\gamma$ similar words to $w_j$
      $W_{sim} := \{w_j^i\}_{i=1}^{\gamma}$;
    Add $W_{sim}$ to $W$;
end
Compute similarity $s_i$ between one document
  that merges all keywords
  $W = \{w_j\}_{j=1}^{(\alpha+\gamma)n_{\mathrm{W}}}$ and each document $x_i^{\mathrm{U}}$;
foreach $x_i^{\mathrm{U}}$ do
    if $s_i > 0$ and in the top-$\phi$(%) then
        Add $x_i^{\mathrm{U}}$ to $X_{\mathrm{CP}}$;
    else
        Add $x_i^{\mathrm{U}}$ to $X_{\mathrm{CN}}$;
    end
end
return $X_{\mathrm{CP}}$, $X_{\mathrm{CN}}$
```

$$X_{\mathrm{CP}} := \{\boldsymbol{x}_i^{\mathrm{CP}}\}_{i=1}^{n_{\mathrm{CP}}} \overset{\text{i.i.d.}}{\sim} p_\theta(\boldsymbol{x}),$$
$$X_{\mathrm{CN}} := \{\boldsymbol{x}_j^{\mathrm{CN}}\}_{j=1}^{n_{\mathrm{CN}}} \overset{\text{i.i.d.}}{\sim} p_{\theta'}(\boldsymbol{x}),$$

where

$$p_\theta(\boldsymbol{x}) := \theta p(\boldsymbol{x}|y=1) + (1-\theta)p(\boldsymbol{x}|y=-1),$$
$$p_{\theta'}(\boldsymbol{x}) := \theta' p(\boldsymbol{x}|y=1) + (1-\theta')p(\boldsymbol{x}|y=-1).$$

X_{CP} and X_{CN} are the sets of corrupted positive and corrupted negative data, respectively. We utilize the fact that optimizing AUC can be performed effectively even when the data is highly corrupted by using a symmetric loss function (Charoenphakdee et al., 2019). More specifically, let us consider a corrupted AUC risk. First, let us define $\mathbb{E}_\theta$ and $\mathbb{E}_{\theta'}$ as the expectation over p_θ and $p_{\theta'}$, respectively. A corrupted AUC risk is defined as

$$R^{\ell}_{\text{AUC-Corr}}(g) = \mathbb{E}_\theta[\mathbb{E}_{\theta'}[\ell(g(\boldsymbol{x}) - g(\boldsymbol{x}'))]]. \quad (3)$$

Intuitively, the risk (3) is minimized by a function g that gives a higher value on corrupted positive data over corrupted negative data. The following theorem establishes the relationship between the clean AUC risk and corrupted AUC risk.

Theorem 1 (Charoenphakdee et al. (2019)). *Let* $\varphi^{\ell}(\boldsymbol{x}, \boldsymbol{x}') = \ell(g(\boldsymbol{x}) - g(\boldsymbol{x}')) + \ell(g(\boldsymbol{x}') - g(\boldsymbol{x}))$. *Then* $R^{\ell}_{\text{AUC-Corr}}(g)$ *can be expressed as*

$$\begin{aligned} R^{\ell}_{\text{AUC-Corr}}(g) &= (\pi - \pi')R^{\ell}_{\text{AUC}}(g) \\ &+ \underbrace{(1-\pi)\pi'\mathbb{E}_{\text{P}}[\mathbb{E}_{\text{N}}[\varphi^{\ell}(\boldsymbol{x}^{\text{P}}, \boldsymbol{x}^{\text{N}})]]}_{\text{Excessive term}} \\ &+ \underbrace{\frac{\pi\pi'}{2}\mathbb{E}_{\text{P}'}[\mathbb{E}_{\text{P}}[\varphi^{\ell}(\boldsymbol{x}^{\text{P}'}, \boldsymbol{x}^{\text{P}})]]}_{\text{Excessive term}} \\ &+ \underbrace{\frac{(1-\pi)(1-\pi')}{2}\mathbb{E}_{\text{N}'}[\mathbb{E}_{\text{N}}[\varphi^{\ell}(\boldsymbol{x}^{\text{N}'}, \boldsymbol{x}^{\text{N}})]]}_{\text{Excessive term}}. \end{aligned}$$

Theorem 1 shows that minimizing the corrupted risk $R^{\ell}_{\text{AUC-Corr}}(g)$ implies minimizing both the clean risk $R^{\ell}_{\text{AUC}}(g)$ and excessive terms. However, since we only aim to minimize the clean risk, minimizing the corrupted risk may not effectively minimize the clean risk since excessive terms can be minimized instead and lead to overfitting. Charoenphakdee et al. (2019) showed that with a specific class of losses, excessive terms become constant. The following theorem suggests that by using a family of symmetric losses, the corrupted AUC risk can be expressed as an affine transformation of the clean AUC risk.

Theorem 2 (Charoenphakdee et al. (2019)). *Let* ℓ_{sym} *be a symmetric loss such that* $\ell_{\text{sym}}(z) + \ell_{\text{sym}}(-z) = K$, *where* K *is a positive constant. Then,* $R^{\ell_{\text{sym}}}_{\text{AUC-Corr}}(g)$ *can be expressed as*

$$\begin{aligned} R^{\ell_{\text{sym}}}_{\text{AUC-Corr}}(g) &= (\theta - \theta')R^{\ell_{\text{sym}}}_{\text{AUC}}(g) \\ &+ \frac{K(1-\theta+\theta')}{2}. \end{aligned}$$

According to Theorem 2, we can optimize AUC by using a symmetric loss *without estimating* θ *and* θ' and do not suffer from excessive terms. This insight was also suggested by Menon et al. (2015), but they focused on more general losses, while it has been shown that using a symmetric loss can be preferable (Charoenphakdee et al., 2019). Our experiments also support that using a symmetric loss is preferable, while using a non-symmetric loss still provided reasonable performance. Similarly to AUC, we can also maximize the balanced accuracy effectively with a symmetric loss (see van Rooyen et al. (2015); Charoenphakdee et al. (2019) for more details on the balanced accuracy). Since the result of Charoenphakdee et al. (2019) only suggests that the minimizer of the corrupted risk and clean risk are identical, we provide a theory to explain why a more accurate pseudo-labeling algorithm can improve the performance by proving an estimation error bound in Section 5.

4.3 Optimizing Accuracy and F_1-measure

Here, we review other evaluation metrics and discuss how to adjust our framework for optimizing other evaluation metrics. Many evaluation metrics can be optimized if a suitable threshold and $p(y = 1|\boldsymbol{x})$ are known (Yan et al., 2018), e.g., $\text{sign}[p(y = 1|x) - \frac{1}{2}]$ is the Bayes-optimal solution for the accuracy. It is known that the Bayes-optimal solution of AUC maximization is any function that has a strictly monotonic relationship with $p(y = 1|\boldsymbol{x})$ (Menon and Williamson, 2016). Therefore, finding an appropriate threshold with an AUC maximizer can give an effective classifier (Narasimhan and Agarwal, 2013).

For the accuracy and F_1-measure, we propose to optimize AUC first and then adjust the threshold to optimize other metrics. It is important to emphasize that a big challenge behind this problem is that *there is no information about the threshold from given data since all given data are unlabeled.*

Without additional assumptions, we argue that it is *theoretically impossible* to draw an optimal threshold to optimize the accuracy and F_1-measure for this problem. Unlike supervised-learning, where we have positive and negative data, the threshold information is provided from labels and we can draw the decision boundary accordingly. On the other hand, one can learn a bipartite ranking function effectively by AUC optimization in our setting, which suggests the idea to learn a reliable ranking function first, and then adjust the threshold.

Suppose that we know the proportion of positive data π in the training data and the class prior of the training and test data are identical. A reasonable threshold $\beta \in \mathbb{R}$ can be given as

$$\pi = \int \text{sign}(g(\boldsymbol{x}) - \beta)p_{\pi}(\boldsymbol{x})d\boldsymbol{x}. \qquad (4)$$

Intuitively, we use a threshold β that decides the top proportion π as positives and negative otherwise, which can optimize the accuracy effectively. This threshold is known as a precision-recall breakeven point, where it is the point where the precision equals to recall (see Kato et al. (2019) for its proof). Therefore, it is a reasonable threshold for the F_1-measure since it is a harmonic mean of precision and recall. With unlabeled documents and π, we can decide β that satisfies the empirical version of Eq. (4).

Existing methods, which may learn a threshold without knowing π, may actually learn a wrong threshold. Table 3 illustrates the failure of a default threshold when learning a classifier in noisy environments and Table 4 shows that much better performance on F_1-measure and accuracy can be recover by adjusting the threshold with Eq. (4). Regarding the assumption that training and test class priors are identical, it is the common assumption if we want to use these metrics for evaluation and is implicitly assumed in the existing work (Jin et al., 2017). If the class priors are not identical, we suggest not to use these metrics since it can be misleading because a good F_1-measure in training data cannot guarantee a good F_1-measure in the test data. In real-world applications, such situations may occur when we do not collect unlabeled data directly from the test environment or the test environment is prone to be changed over time. If it is impossible to estimate π or the test condition can be changed, we suggest to use other evaluation metrics such as AUC or precision at k.

5 Theoretical Analysis

In this section, we provide a theoretical guarantee for our method. Although it has been shown that the risk minimizer of corrupted data is identical to that of clean data (Charoenphakdee et al., 2019), the existing work cannot explain why a good pseudo-labeling algorithm can give a better performance. Throughout this section, we assume that an output of a loss function ℓ_{sym} is bounded by $[0, K]$ without loss of generality for a nonnegative symmetric loss. Here, we prove an estimation error bound, which suggests that a larger gap between $\theta - \theta'$ leads to a tighter bound and thus faster convergence. Note that the gap is largest when we have clean positive data (i.e., $\theta = 1$) and clean negative data (i.e., $\theta' = 0$). Therefore, a good pseudo-labeling is an algorithm that can divide the data with a large gap (i.e., $\theta - \theta'$ is large). All proofs can be found in Appendix A.

Let $\hat{g} \in \mathcal{G}$ be a minimizer of the empirical corrupted AUC risk $\widehat{R}^{\ell_{\mathrm{sym}}}_{\text{AUC-Corr}}$ in the hypothesis class $\mathcal{G}$. Let $g^* \in \mathcal{G}$ be a minimizer of the expected risk of clean AUC risk $R^{\ell_{\mathrm{sym}}}_{\mathrm{AUC}}$. Then, the following lemma establishes the relationship between an estimation error bound of the AUC risk and the uniform deviation bound of the corrupted AUC risk.

Lemma 3. *An estimation error bound of the clean AUC risk R^{ℓ}_{AUC} can be given as follows.*

$$R^{\ell_{\mathrm{sym}}}_{\mathrm{AUC}}(\hat{g}) - R^{\ell_{\mathrm{sym}}}_{\mathrm{AUC}}(g^*) \leq \frac{2}{\theta - \theta'} \sup_{g \in \mathcal{G}} |R^{\ell_{\mathrm{sym}}}_{\text{AUC-Corr}}(g) - \widehat{R}^{\ell_{\mathrm{sym}}}_{\text{AUC-Corr}}(g)|.$$

Next, let $\mathcal{Q}^{\ell_{\mathrm{sym}}}_{\mathcal{G}}$ be a class of functions mapping $\mathcal{X}^2$ to $[0, K]$ such that $\mathcal{Q}^{\ell_{\mathrm{sym}}}_{\mathcal{G}} = \{Q : (\boldsymbol{x}, \boldsymbol{x}') \to \ell_{\mathrm{sym}}(g(\boldsymbol{x}) - g(\boldsymbol{x}')), g \in \mathcal{G}\}$. Given samples $(x_1, \ldots, x_n) \in \mathcal{X}^n$ and $({x'}_1, \ldots, {x'}_m) \in \mathcal{X}^m$ independently and identically drawn from a distribution with densities μ and μ', respectively. The empirical bipartite Rademacher complexity (Usunier et al., 2005) of a function class $\mathcal{Q}^{\ell_{\mathrm{sym}}}_{\mathcal{G}}$ is defined as

$$\hat{\mathfrak{R}}_{n,m}(\mathcal{Q}^{\ell_{\mathrm{sym}}}_{\mathcal{G}}) = 2\mathbb{E}_{\sigma,\upsilon} \sup_{Q \in \mathcal{Q}^{\ell_{\mathrm{sym}}}_{\mathcal{G}}} \frac{1}{nm} \sum_{i=1}^{n} \sum_{j=1}^{m} \frac{\sigma_i + \upsilon_j}{2} \left[Q(\boldsymbol{x}_i, {\boldsymbol{x}'}_j)\right],$$

where the inner expectation is taken over $\sigma = (\sigma_1, \ldots, \sigma_n)$ and $\upsilon = (\upsilon_1, \ldots, \upsilon_m)$ which are independent random variables taking values in $\{-1, +1\}$ uniformly.

Then, the bipartite Rademacher complexity can be defined as

$$\mathfrak{R}_{n,m}(\mathcal{Q}^{\ell_{\mathrm{sym}}}_{\mathcal{G}}) = \mathbb{E}_{x_1,\ldots,x_n,{x'}_1,\ldots,{x'}_m} \hat{\mathfrak{R}}_{n,m}(\mathcal{Q}^{\ell_{\mathrm{sym}}}_{\mathcal{G}}).$$

Let $\mathfrak{R}^{\mathrm{Corr}}_{n_{\mathrm{CP}},n_{\mathrm{CN}}}$ be the bipartite Rademacher complexities of pseudo-labeled data where $\mu = p_\theta$ and $\mu' = p'_\theta$. Next, we provide a uniform deviation bound of the corrupted AUC risk.

Lemma 4 (Uniform deviation bound)**.** *For all $Q \in \mathcal{Q}^{\ell_{\mathrm{sym}}}_{\mathcal{G}}$ and $\delta \in (0, 1)$, with probability at least $1 - \delta$, we have*

$$\sup_{g \in \mathcal{G}} \left|R^{\ell_{\mathrm{sym}}}_{\text{AUC-Corr}}(g) - \widehat{R}^{\ell_{\mathrm{sym}}}_{\text{AUC-Corr}}(g)\right| \leq K\sqrt{\frac{(n_{\mathrm{CP}} + n_{\mathrm{CN}}) \log \frac{1}{\delta}}{2n_{\mathrm{CP}} n_{\mathrm{CN}}}} + 2\mathfrak{R}^{\mathrm{Corr}}_{n_{\mathrm{CP}},n_{\mathrm{CN}}}(\mathcal{Q}^{\ell_{\mathrm{sym}}}_{\mathcal{G}}).$$

where the probability is over repeated sampling of data for evaluating $\widehat{R}^{\ell}_{\text{AUC-Corr}}(g)$.

Table 2: Keywords for each dataset

Dataset	Keywords
Subj	wonderful terrible feel happy ugly even horrible interesting funny dramatic romantic compassionate
Custrev	easy excellent nice great good love amazing best awesome perfect definitely better happy compassionate
MPQA	support hope help good great love
AYI	great best excellent friendly awesome nice amazing
20NG	sports baseball hockey

Then, by combining Lemma 3 and Lemma 4, we obtain the following theorem.

Theorem 5 (Estimation error bound). *For all $Q \in \mathcal{Q}_{\mathcal{G}}^{\ell_{\text{sym}}}$ and $\delta \in (0,1)$, with probability at least $1-\delta$, we have*

$$R_{\text{AUC}}^{\ell_{\text{sym}}}(\hat{g}) - R_{\text{AUC}}^{\ell_{\text{sym}}}(g^*) \leq \frac{1}{\theta - \theta'}\left[K\sqrt{\frac{2(n_{\text{CP}} + n_{\text{CN}})\log\frac{1}{\delta}}{n_{\text{CP}}n_{\text{CN}}}}\right] + \frac{4\mathfrak{R}^{\text{Corr}}_{n_{\text{CP}},n_{\text{CN}}}(\mathcal{Q}_{\mathcal{G}}^{\ell_{\text{sym}}})}{\theta - \theta'},$$

where the probability is over repeated sampling of X_{CP} and X_{CN} for training $\hat{g}$.

Theorem 5 shows that a gap $\frac{1}{\theta-\theta'}$ affects the tightness of the bound. When θ and θ' are close to each other, i.e., when the data is highly corrupted, the bound becomes loose. This illustrates the difficulty of the task when pseudo-labeling algorithm performs poorly and we may need more data. Nevertheless, as long as $\theta > \theta'$, with all parametric models with their norm is bounded such as neural networks with weight decay or kernel model, our learning framework is consistent, i.e., the estimation error converges to zero as $n_{\text{CP}}, n_{\text{CN}} \to \infty$.

6 Experimental Results

We present experimental results in this section with evaluation metrics include AUC, macro F_1-measure, accuracy (ACC) and precision at 100 (Prec@100). Prec@100 is the ratio of the true positive data over in the top-100 ranking score.

6.1 Experiment Setup

Datasets: We used five datasets, namely the Subjectivity dataset (Pang and Lee, 2004) (Subj), Customer reviews (Hu and Liu, 2004) (Custrev), Opinion mining in MPQA corpus (MPQA), Product reviews from Amazon, Yelp, and IMDb (AYI), 20 Newsgroups (Lang, 1995) (20NG) with *baseball* and *hockey* groups as positive. More information on the datasets can be found in Appendix B.

Table 3: Failure of optimizing F_1-measure if thresholds are not adjusted. Full results can be found in Appendix C. By adjusting the threshold in Table 4 can substantially improve the performance.

Methods	Subj	MPQA	AYI	20NG
Maxent	63.4 (0.31)	50.1 (0.22)	42.5 (0.35)	47.4 (0.05)
NB	73.7 (0.23)	53.8 (0.22)	65.8 (0.42)	23.7 (0.25)
RandomForest	33.3 (0.00)	43.5 (0.20)	35.0 (0.20)	47.2 (0.00)
KNN	43.6 (0.23)	51.0 (0.16)	61.6 (0.43)	84.3 (0.26)

Keywords for each dataset are shown in Table 2.

Common setup: First, we need to pseudo-label documents. We used Algorithm 1 to feed pseudo-labeled documents to all methods, where ϕ is set to 90. (α, γ) were set to (3,5) for all datasets except Subj, which was (1,50). We used 50-dimensional features for GloVe word embeddings. For ACC and F_1, we gave true thresholds π to all methods to see the top performance of each method. We also provide the results with varying thresholds and heuristic thresholds in Appendix C, where the trends of performance for each method do not differ much from Table 4. Note that AUC and Prec@100 do not use a threshold to evaluate. The experimental results are reported in the mean value and standard error of 20 trials. "N/A" indicates that an algorithm is not terminated due to too many number of vocabularies.

Baselines: We compared our method with three categories of baselines: the text-feature, GloVe-feature, and zero-shot baselines. For the text-feature baselines, we use naive Bayes (NB), the maximum entropy model (Maxent), and naive Bayes that is implemented for learning from positive and unlabeled data (PU-NB). All implementations were from Natural Language Toolkit (NLTK) (Loper and Bird, 2002). For the GloVe-feature baselines, we used mean word vectors as features and employed a random forest (RandomForest) and K-nearest neighbors (KNN), which were implemented by Scikit-learn (Pedregosa et al., 2011). Finally, the zero-shot baselines did not use unlabeled data but simply GloVe to rank the score of a document (GloVeRanking) and keyword voting (Voting). We also showed the performance when fully-labeled data are given as references. O-Maxent is a maxent model with fully-labeled data and O-Sigmoid is our framework that skips the pseudo-labeling step and uses fully-labeled data for AUC optimization.

Proposed methods: For the AUC optimization part, we used the recurrent convolutional

Table 4: Mean value and standard error of 20 trials for the AUC, F$_1$-measure, accuracy (ACC), and precision at 100 (Prec@100) of learning from relavant keywords and unlabeled documents. Outperforming methods are highlighted in boldface using one-sided t-test with the significance level of 5%. The sigmoid loss is symmetric while the logistic loss is non-symmetric.

Dataset	Evaluation	Proposed framework		Text-feature baselines			GloVe-feature baselines		Zero-shot baselines		Oracle	
		Sigmoid	Logistic	PU-NB	NB	Maxent	RandomForest	KNN	GloVeRanking	Voting	O-Maxent	O-Sigmoid
Subj	AUC	**88.1 (0.35)**	84.1 (0.30)	55.4 (0.13)	85.0 (0.18)	84.6 (0.20)	82.4 (0.27)	73.6 (0.29)	81.7 (0.19)	70.2 (0.24)	97.4 (0.06)	93.6 (0.11)
	F$_1$	**80.1 (0.38)**	76.0 (0.32)	47.1 (0.20)	76.3 (0.16)	76.3 (0.24)	75.1 (0.27)	63.6 (0.32)	74.5 (0.13)	63.5 (0.18)	92.0 (0.13)	86.4 (0.14)
	ACC	**80.1 (0.38)**	76.0 (0.32)	55.0 (0.13)	76.3 (0.16)	76.3 (0.24)	75.1 (0.27)	65.0 (0.28)	74.5 (0.13)	64.1 (0.18)	92.0 (0.13)	86.4 (0.14)
	Prec@100	**96.3 (0.60)**	**95.1 (0.60)**	0.9 (0.09)	**95.9 (0.33)**	94.7 (0.39)	93.2 (0.50)	91.5 (0.59)	**95.2 (0.54)**	85.8 (0.93)	99.3 (0.15)	97.8 (0.27)
Custrev	AUC	71.2 (0.34)	70.7 (0.34)	59.7 (0.32)	**74.2 (0.35)**	69.9 (0.46)	62.6 (0.47)	67.9 (0.40)	55.3 (0.39)	67.4 (0.32)	75.0(0.38)	78.5(0.30)
	F$_1$	**63.6 (0.41)**	**63.1 (0.31)**	58.4 (0.35)	**63.3 (0.49)**	62.1 (0.43)	58.3 (0.44)	60.8 (0.33)	53.0 (0.40)	38.9 (0.00)	68.5 (0.42)	70.2 (0.29)
	ACC	66.5 (0.40)	66.0 (0.31)	64.8 (0.32)	**69.8 (0.37)**	65.8 (0.39)	61.9 (0.40)	66.6 (0.29)	56.6 (0.37)	63.7 (0.00)	71.4 (0.41)	72.5 (0.28)
	Prec@100	91.2 (0.46)	91.5 (0.49)	**99.2 (0.18)**	89.8 (0.55)	91.2 (0.57)	80.9 (0.49)	86.3 (0.85)	75.1 (0.81)	87.3 (0.66)	82.2 (0.35)	86.9 (0.28)
MPQA	AUC	**80.4 (0.44)**	78.7 (0.37)	52.1 (0.27)	56.4 (0.31)	56.7 (0.23)	69.1 (0.55)	60.1 (0.23)	63.6 (0.26)	56.0 (0.12)	78.3 (0.25)	86.8 (0.18)
	F$_1$	**71.7 (0.44)**	69.8 (0.31)	46.7 (0.23)	54.3 (0.28)	53.1 (0.20)	62.4 (0.45)	23.8 (0.00)	57.5 (0.17)	23.8 (0.00)	69.8 (0.19)	77.9 (0.22)
	ACC	**75.6 (0.39)**	74.0 (0.27)	47.1 (0.24)	62.4 (0.28)	58.4 (0.17)	67.4 (0.39)	31.2 (0.00)	63.3 (0.17)	31.2 (0.00)	72.8 (0.20)	81.0 (0.19)
	Prec@100	**81.5 (0.97)**	77.5 (1.02)	10.8 (3.37)	69.5 (0.86)	63.8 (1.93)	76.9 (1.06)	78.7 (0.80)	50.6 (0.60)	74.7 (0.69)	94.8 (0.46)	90.5 (0.52)
AYI	AUC	**76.0 (0.41)**	**75.6 (0.43)**	60.5 (0.39)	71.2 (0.41)	60.7 (0.46)	70.1 (0.55)	72.5 (0.39)	62.4 (0.53)	61.0 (0.33)	84.6 (0.32)	81.1 (0.40)
	F$_1$	**69.3 (0.36)**	**68.8 (0.40)**	58.9 (0.47)	61.6 (0.38)	56.6 (0.36)	64.5 (0.53)	65.5 (0.52)	58.7 (0.51)	33.5 (0.00)	76.8 (0.37)	73.0 (0.39)
	ACC	**69.3 (0.36)**	**68.8 (0.40)**	60.1 (0.41)	62.5 (0.35)	56.8 (0.36)	64.6 (0.53)	65.8 (0.44)	58.7 (0.51)	50.5 (0.00)	76.9 (0.37)	73.0 (0.39)
	Prec@100	**87.5 (0.55)**	**87.5 (0.62)**	74.5 (2.20)	85.1 (0.71)	70.2 (1.00)	77.2 (0.99)	82.5 (0.69)	72.4 (0.91)	79.2 (0.87)	95.6 (0.47)	90.1 (0.73)
20NG	AUC	96.4 (0.12)	96.0 (0.15)	N/A	77.1 (0.21)	57.6 (0.32)	**96.8 (0.16)**	94.7 (0.16)	95.0 (0.17)	62.9 (0.22)	65.5 (0.46)	99.0 (0.05)
	F$_1$	**90.8 (0.20)**	**90.6 (0.21)**	N/A	58.4 (0.22)	52.4 (0.25)	89.6 (0.28)	86.7 (0.59)	**90.5 (0.18)**	9.6 (0.00)	56.8 (0.29)	94.1 (0.14)
	ACC	**96.5 (0.08)**	**96.4 (0.08)**	N/A	70.2 (0.31)	81.6 (0.11)	96.1 (0.10)	94.5 (0.35)	**96.4 (0.07)**	10.6 (0.00)	83.5 (0.11)	97.8 (0.05)
	Prec@100	**99.5 (0.15)**	**99.1 (0.24)**	N/A	0.4 (0.11)	17.6 (0.77)	**99.5 (0.15)**	97.6 (0.38)	97.5 (0.28)	85.2 (1.03)	32.0 (1.31)	99.9 (0.07)

Table 5: Mean F$_1$-measure and standard error of 20 trials with varying thresholds with different $\hat{\pi}$ in Eq. (4).

Dataset	Methods	0.05	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
Subj	Sigmoid	43.3 (0.17)	51.2 (0.24)	63.9 (0.34)	72.5 (0.27)	77.9 (0.37)	**80.1 (0.38)**	78.7 (0.32)	73.7 (0.29)	64.7 (0.32)	51.8 (0.30)
	Maxent	37.1 (0.15)	41.9 (0.18)	52.5 (0.27)	62.1 (0.19)	70.6 (0.27)	**76.3 (0.24)**	74.5 (0.25)	64.1 (0.25)	50.3 (0.22)	39.3 (0.13)
	RandomForest	42.1 (0.22)	49.9 (0.15)	61.6 (0.22)	69.1 (0.26)	73.5 (0.29)	**75.1 (0.27)**	73.5 (0.17)	69.0 (0.21)	61.2 (0.24)	49.6 (0.23)
	GloVe Ranking	43.1 (0.24)	50.5 (0.25)	61.9 (0.25)	69.3 (0.27)	73.6 (0.18)	**74.5 (0.13)**	72.8 (0.18)	68.0 (0.16)	60.6 (0.23)	49.4 (0.22)
20NG	Sigmoid	79.9 (0.23)	**91.2 (0.18)**	78.4 (0.20)	68.2 (0.15)	60.0 (0.14)	52.7 (0.13)	45.3 (0.15)	37.8 (0.13)	29.7 (0.13)	20.5 (0.13)
	Maxent	51.5 (0.19)	**52.3 (0.24)**	51.5 (0.21)	49.5 (0.16)	46.6 (0.15)	43.0 (0.16)	38.1 (0.14)	32.9 (0.15)	26.3 (0.16)	19.0 (0.13)
	RandomForest	79.4 (0.33)	**89.8 (0.29)**	78.5 (0.17)	68.2 (0.15)	59.8 (0.16)	52.6 (0.13)	45.4 (0.12)	37.7 (0.16)	29.6 (0.13)	20.3 (0.10)
	GloVe Ranking	79.2 (0.34)	**90.7 (0.17)**	78.2 (0.22)	67.9 (0.19)	59.9 (0.14)	52.3 (0.11)	45.0 (0.10)	37.6 (0.10)	29.3 (0.11)	20.1 (0.13)

neural networks (RCNN) model (Lai et al., 2015) with two layer long short-term memory (LSTM) (Hochreiter and Schmidhuber, 1997). We used Adam (Kingma and Ba, 2014) as an optimization method. We used the symmetric sigmoid loss (Sigmoid). We also show the results of the non-symmetric logistic loss (Logistic) to validate the improvement from using a symmetric loss. Implementation details are provided in Appendix B.

6.2 End-to-end Classification Performance

Table 4 shows the classification performance. It can be observed that our proposed framework outperforms other baselines in many cases. Moreover, the sigmoid loss (Sigmoid) outperforms the logstic loss (Logistic), which agrees with Theorem 2. Nevertheless, the performance of the logistic loss is desirable compared with baselines. It can be observed that GloveRanking, which does not use unlabeled data can outperform many baselines in 20NG. This can be due to the fact that pseudo-labeling data were corrupted and degraded the performance of a classifier heavily. In Table 4, for MPQA and 20NG, our proposed framework with symmetric losses is able to outperform O-Maxent in the F$_1$-measure, AUC score, and accuracy without having access to labels. One possible reason that O-Maxent does not perform well on 20NG is class imbalance.

Table 5 shows the accuracy and F$_1$-measure with varying thresholds. The true class prior of Subj and 20NG were 0.5 and 0.11, respectively. It can be observed that the threshold based on Eq. (4) gives a desirable performance when $\hat{\pi}$ is close to π. For Prec@100, we can see that Prec@100 for 20NG for the sigmoid loss in our framework and RandomForest is 99.5, without any labeled data, which indicates that we may pick top-100 documents as positive with almost perfect precision.

7 Conclusion

We proposed a theoretically-grounded framework for learning from relevant keywords and unlabeled documents. Our framework is highly flexible and can guarantee any heuristic pseudo-labeling method as long as an algorithm can divide unlabeled documents into two sets of data with different proportions of positive data. Experiments showed the usefulness of the proposed method.

Acknowledgement

NC was supported by MEXT scholarship and MS was supported by JST CREST Grant Number JPMJCR1403. Part of the research was done while NC was visiting Chulalongkorn University with the support of U. Tokyo IST Research Internship Program.

References

Shivani Agarwal, Thore Graepel, Ralf Herbrich, Sariel Har-Peled, and Dan Roth. 2005. Generalization bounds for the area under the roc curve. *JMLR*, 6(Apr):393–425.

Peter L Bartlett, Michael I Jordan, and Jon D McAuliffe. 2006. Convexity, classification, and risk bounds. *JASA*, 101(473):138–156.

Shai Ben-David, Nadav Eiron, and Philip M Long. 2003. On the difficulty of approximately maximizing agreements. *Journal of Computer and System Sciences*, 66(3):496–514.

Olivier Bousquet, Stéphane Boucheron, and Gábor Lugosi. 2003. Introduction to statistical learning theory. In *Summer School on Machine Learning*, pages 169–207. Springer.

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

Nontawat Charoenphakdee, Jongyeong Lee, and Masashi Sugiyama. 2019. On symmetric losses for learning from corrupted labels. *ICML*.

Xingyuan Chen, Yunqing Xia, Peng Jin, and John Carroll. 2015. Dataless text classification with descriptive lda. In *AAAI*.

Aaron M Cohen and William R Hersh. 2005. A survey of current work in biomedical text mining. *Briefings in bioinformatics*, 6(1):57–71.

Corinna Cortes and Mehryar Mohri. 2004. Auc optimization vs. error rate minimization. In *NeurIPS*, pages 313–320.

Vitaly Feldman, Venkatesan Guruswami, Prasad Raghavendra, and Yi Wu. 2012. Agnostic learning of monomials by halfspaces is hard. *SIAM Journal on Computing*, 41(6):1558–1590.

George Forman. 2003. An extensive empirical study of feature selection metrics for text classification. *JMLR*, 3(Mar):1289–1305.

Wei Gao and Zhi-Hua Zhou. 2015. On the consistency of auc pairwise optimization. In *IJCAI*, pages 939–945.

Aritra Ghosh, Naresh Manwani, and PS Sastry. 2015. Making risk minimization tolerant to label noise. *Neurocomputing*, 160:93–107.

James A Hanley and Barbara J McNeil. 1982. The meaning and use of the area under a receiver operating characteristic (roc) curve. *Radiology*, 143(1):29–36.

Alan Herschtal and Bhavani Raskutti. 2004. Optimising area under the roc curve using gradient descent. In *ICML*, page 49. ACM.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, pages 1735–1780.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *KDD*, pages 168–177. ACM.

Xia Hu and Huan Liu. 2012. Text analytics in social media. In *Mining text data*, pages 385–414. Springer.

Chung-Chi Huang and Zhiyong Lu. 2015. Community challenges in biomedical text mining over 10 years: success, failure and the future. *Briefings in bioinformatics*, 17(1):132–144.

Yiping Jin, Dittaya Wanvarie, and Phu Le. 2017. Combining lightly-supervised text classification models for accurate contextual advertising. In *IJCNLP*, volume 1, pages 545–554.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*.

Masahiro Kato, Takeshi Teshima, and Junya Honda. 2019. Learning from positive and unlabeled data with a selection bias. *ICLR*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *ICLR*.

Siwei Lai, Liheng Xu, Kang Liu, and Jun Zhao. 2015. Recurrent convolutional neural networks for text classification. In *AAAI*.

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

Ximing Li and Bo Yang. 2018. A pseudo label based dataless naive bayes algorithm for text classification with seed words. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1908–1917.

Bing Liu and Lei Zhang. 2012. A survey of opinion mining and sentiment analysis. In *Mining text data*, pages 415–463. Springer.

Edward Loper and Steven Bird. 2002. Nltk: the natural language toolkit. *arXiv preprint cs/0205028*.

Henry B Mann and Donald R Whitney. 1947. On a test of whether one of two random variables is stochastically larger than the other. *The annals of mathematical statistics*, pages 50–60.

Colin McDiarmid. 1989. On the method of bounded differences. *Surveys in Combinatorics*, 141(1):148–188.

Walaa Medhat, Ahmed Hassan, and Hoda Korashy. 2014. Sentiment analysis algorithms and applications: A survey. *Ain Shams engineering journal*, 5(4):1093–1113.

Aditya Menon, Brendan van Rooyen, Cheng Soon Ong, and Bob Williamson. 2015. Learning from corrupted binary labels via class-probability estimation. In *ICML*, pages 125–134.

Aditya Krishna Menon and Robert C Williamson. 2016. Bipartite ranking: a risk-theoretic perspective. *JMLR*, 17(1):6766–6867.

Harikrishna Narasimhan and Shivani Agarwal. 2013. On the relationship between binary classification, bipartite ranking, and binary class probability estimation. In *NeurIPS*, pages 2913–2921.

Kamal Nigam, Andrew Kachites McCallum, Sebastian Thrun, and Tom Mitchell. 2000. Text classification from labeled and unlabeled documents using em. *Machine learning*, 39(2-3):103–134.

Bo Pang and Lillian Lee. 2004. A sentimental education: Sentiment analysis using subjectivity. In *ACL*, pages 271–278.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. Scikit-learn: Machine learning in Python. *JMLR*, 12:2825–2830.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*, pages 1532–1543.

Sashank J Reddi, Satyen Kale, and Sanjiv Kumar. 2018. On the convergence of Adam and beyond. In *ICLR*.

Brendan van Rooyen, Aditya Krishna Menon, and Robert C Williamson. 2015. An average classification algorithm. *arXiv preprint arXiv:1506.01520*.

Fabrizio Sebastiani. 2002. Machine learning in automated text categorization. *ACM computing surveys (CSUR)*, 34(1):1–47.

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

Nicolas Usunier, Massih-Reza Amini, and Patrick Gallinari. 2005. A data-dependent generalisation error bound for the auc. In *ICML Workshop*.

Vladimir Vapnik. 1998. *Statistical learning theory. 1998*, volume 3. Wiley, New York.

Bowei Yan, Oluwasanmi Koyejo, Kai Zhong, and Pradeep Ravikumar. 2018. Binary classification with karmic, threshold-quasi-concave metrics. *arXiv preprint arXiv:1806.00640*.

Lian Yan, Robert H Dodier, Michael Mozer, and Richard H Wolniewicz. 2003. Optimizing classifier performance via an approximation to the wilcoxon-mann-whitney statistic. In *ICML*, pages 848–855.

Daniel Zeng, Hsinchun Chen, Robert Lusch, and Shu-Hsing Li. 2010. Social media analytics and intelligence. *IEEE Intelligent Systems*, 25(6):13–16.

Tong Zhang. 2004. Statistical behavior and consistency of classification methods based on convex risk minimization. *Annals of Statistics*, pages 56–85.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *NeurIPS*, pages 649–657.

Zhi-Hua Zhou. 2017. A brief introduction to weakly supervised learning. *National Science Review*, 5(1):44–53.