Classifying Extremely Short Texts by Exploiting Semantic Centroids in Word Mover's Distance Space

Changchun Li, Jihong Ouyang, Ximing Li*
College of Computer Science and Technology, Jilin University, China
Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, China
changchunli93@gmail.com,ouyj@jlu.edu.cn,liximing86@gmail.com

ABSTRACT

Automatically classifying extremely short texts, such as social media posts and web page titles, plays an important role in a wide range of content analysis applications. However, traditional classifiers based on bag-of-words (BoW) representations often fail in this task. The underlying reason is that the document similarity can not be accurately measured under BoW representations due to the extreme sparseness of short texts. This results in significant difficulty to capture the generality of short texts. To address this problem, we use a better regularized word mover's distance (RWMD), which can measure distances among short texts at the semantic level. We then propose a RWMD-based centroid classifier for short texts, named RWMD-CC. Basically, RWMD-CC computes a representative semantic centroid for each category under the RWMD measure, and predicts test documents by finding the closest semantic centroid. The testing is much more efficient than the prior art of K nearest neighbor classifier based on WMD. Experimental results indicate that our RWMD-CC can achieve very competitive classification performance on extremely short texts.

CCS CONCEPTS

• **Computing methodologies → Machine learning algorithms**;
• **Information systems → Clustering and classification**.

KEYWORDS

Extremely Short Texts; Regularized Word Mover's Distance; Semantic Centroid; Hypothesis Margin

ACM Reference Format:
Changchun Li, Jihong Ouyang, Ximing Li. 2019. Classifying Extremely Short Texts by Exploiting Semantic Centroids in Word Mover's Distance Space. In *Proceedings of the 2019 World Wide Web Conference (WWW'19), May 13–17, 2019, San Francisco, CA, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3308558.3313397

Table 1: Examples of document similarity. All values are computed after a removal of the stopwords. *BoW*: the cosine of the BoW representation. *WMD*: the normalized inverse of WMD using a pre-trained *Glove* word embeddings.

Document	*BoW*	*WMD*
I will attend the research meeting. *The conference invited me to participate, I will go.*	0.0	0.89
The relentless pursuit of perfection makes things better. *The constant striving for perfection improves the situation.*	0.13	0.92

1 INTRODUCTION

The amount of extremely short texts available online, such as social media posts and web page titles, has exploded. Automatic classification of extremely short texts is significant for a wide range of content analysis applications, e.g., user interest profiling [41] and query suggestion [31] etc. However, it is acknowledged as a very challenging task [17, 19, 32] due to the sparseness of extremely short texts, where a single short text, e.g., Twitter posts, often contains even less than 10 words.

The major difficulty in extremely short text classification is that there are much less informative word co-occurrences among short texts under the bag-of-words (BoW) representations, therefore the existing algorithms are mainly based on feature expansion [8, 12, 18, 20, 21, 25, 29, 32, 35, 38–40]. The representative works include: the method of [29] uses search engines to obtain auxiliary contexts, and applies traditional classifiers to those expanded short texts; the work in [25] expands document features by hidden topics learnt from a big reference Wikipedia corpus, where the latent Dirichlet allocation (LDA) [3] topic model is used as a hidden topic excavator. Although these feature expansion methods empirically improve the classification performance, they are heavily domain-dependent. For example, the hidden topics learnt from reference corpora may be inconsistent with the current dataset, making the concatenation features less discriminative.

In some sense, we can analyze the difficulty of extremely short text classification with BoW representations from a just different perspective of document similarity misalignment. Because the short texts are extremely sparse, i.e., containing very few words, even semantically close text pairs may merely share any same word by no means, resulting in inaccurate similarity measure under their BoW representations (Examples are shown in Table 1). In this situation, it is intractable to find the generality of short texts from the same category and the difference of ones from different categories, raising up significant difficulty to build the decision plane of classifiers. Fortunately, recently a word mover's distance (WMD) [15] has been developed, which measures the optimal transport from

[*] Corresponding author: liximing86@gmail.com

WWW '19, May 13–17, 2019, San Francisco, CA, USA

ACM ISBN 978-1-4503-6674-8/19/05.
https://doi.org/10.1145/3308558.3313397

the embedded words of one document to those of another one, a special case of earth mover's distance (EMD) [26]. The WMD can effectively measure semantic distances among documents using word embeddings [16, 23], therefore it must be a good choice to short texts, instead of distances, as well as similarities, of BoW. Reviewing examples in Table 1, the WMD can effectively measure short text distances even when they share no any same word.

Driven by the success in WMD, we aim at developing short text classifiers using it. A straightforward previous classifier with WMD is the K nearest neighbour (KNN) with a prefetch and prune scheme [15]. It has shown very competitive classification performance, however, the KNN classifier based on WMD is much testing-inefficient especially for big corpora, where for prediction of a test sample one must compute its WMDs with all training samples. Motivated by this, we would like to train a representative semantic centroid for each category, and predict a test sample by comparing the WMDs between it and these semantic centroids, so as to reduce the testing complexity to the category number. We propose a novel **R**egularized **W**ord **M**over's **D**istance **C**entroid **C**lassifier (**RWMD-CC**), of which the objective is built on hypothesis margin with WMD. In RWMD-CC, a problem is that the objective with WMD must be intractable to compute, because the WMD itself is a optimal transport problem. To solve this, we borrow the idea from regularized WMD [6] for efficient optimization, i.e., computing the gradient of the objective. Experimental results indicate that our RWMD-CC outperforms the state-of-the-art baseline classifiers of short texts.

The major contributions of this paper are outlined as follows:

- We develop a both effective and efficient RWMD-CC method for short text classification.
- We build a loss function using hypothesis margin that is based on the structural risk minimization principle, and borrow the idea from regularized WMD for efficient optimization.
- Empirical results indicate that our RWMD-CC outperforms the state-of-the-arts, and is much more efficient on testing than the existing WMD-based methods.

2 PRELIMINARIES

Formally, the word mover's distance (WMD) [15] is a special case of the earth mover's distance (EMD) [26, 27], also named Wasserstein distance [4]. Therefore, we first introduce the EMD, and then show the WMD in detail.

2.1 Optimal Transport and Earth Mover's Distance

The earth mover's distance is the special case of optimal transport distance. Essentially, the optimal transport distance [34] is a measure of the distance between two probability measures over a finite set $\mathcal{V}$ of size $|\mathcal{V}| = V$, i.e., p_1 and p_2. Given a cost function $c : \mathcal{V} \times \mathcal{V} \rightarrow \mathcal{R}$, the optimal transport distance measures the optimal (i.e., cheapest) way to transport the mass in probability measure p_1 to match that in p_2:

$$W_c\left(p_1, p_2\right) = \inf_{\gamma \in \Pi(p_1, p_2)} \int_{\mathcal{V} \times \mathcal{V}} c\left(v_1, v_2\right) \gamma\left(dv_1, dv_2\right), \qquad (1)$$

where $\Pi(p_1, p_2)$ denotes the set of joint probability measures on $\mathcal{V} \times \mathcal{V}$ with p_1 and p_2 as marginals. When the cost function c is a metric or the p-th power with $p \geq 1$ of one over $\mathcal{V}$, Eq.1 is called the earth mover's distance, also Wasserstein distance.

In terms of the discrete measures, both marginals are in the simplex, i.e., $p_1,\ p_2 \in \Delta^{\mathcal{V}} = \{\mathbf{x} \in \mathcal{R}_+^V \mid \mathbf{x}^\top \mathbf{1} = 1\}$, where $\mathbf{1}$ is the all-one vector. The EMD thus can be written as follows:

$$W_c\left(p_1, p_2\right) = \inf_{\gamma \in \Pi(p_1, p_2)} \langle \gamma, C \rangle \qquad \gamma \mathbf{1} = p_1,\ \gamma^\top \mathbf{1} = p_2, \qquad (2)$$

where $\langle \cdot, \cdot \rangle$ stands for the Frobenius dot-product, $C \in \mathcal{R}^{V \times V}$ is the distance matrix computed by cost function c, also a metric. The time complexity of calculating the EMD between two V-dimensional discrete distributions scales at least in $O(V^3 \log(V))$ [24].

2.2 Word Mover's Distance

The WMD can be seemed as a special case of the discrete EMD, used for measuring distances among documents. Given a document d, we can represent it as a normalized V-dimensional term frequency vector:

$$tf_d = \left(\frac{tf_{d1}}{\sum_{v=1}^{V} tf_{dv}}, \frac{tf_{d2}}{\sum_{v=1}^{V} tf_{dv}}, \cdots, \frac{tf_{dV}}{\sum_{v=1}^{V} tf_{dv}} \right), \qquad (3)$$

where tf_{dv} is the term frequency of word v in document d, and V is the number of unique words.

One can consider that the above normalized term frequency vector as a discrete multinomial distribution over words, also a simplex. In this case, given two documents d_i and d_j, we can use the EMD to measure the distance between them, leading to the WMD, i.e., $W_c\left(tf_{d_i}, tf_{d_j}\right)$.

Here, the cost function is the distance or similarity of the corresponding word embeddings of words occurring in the documents, and also a metric over word embeddings. By using word embeddings, the WMD can take semantic information of words into consideration. In this sense, the WMD actually measures the optimal transport from one document to any other at the semantic level.

Regularized WMD. The authors of [6] formulated a smoothed version of EMD by introducing an entropic regularization term to the objective of EMD. This makes the objective strictly convex, so as to efficiently be solved by Sinkhorn-Knopp algorithm [30]. Following this, we can obtain a regularized WMD (RWMD), a.k.a. dual-Sinkhorn divergence [6], formulated by:

$$W_c^\lambda\left(tf_{d_i}, tf_{d_j}\right) = \left\langle \gamma_\lambda^*, C \right\rangle, \qquad (4)$$

where

$$\gamma_\lambda^* = \underset{\gamma \in \Pi(tf_{d_i}, tf_{d_j})}{\arg\min} \ \langle \gamma, C \rangle - \frac{1}{\lambda} H(\gamma),\ \gamma \mathbf{1} = tf_{d_i},\ \gamma^\top \mathbf{1} = tf_{d_j}, \qquad (5)$$

where $H(\cdot)$ denotes the entropy, and $\lambda > 0$ is a regularization parameter. With λ larger, the closer the regularized WMD is to the original EMD. Cuturi [6] shows that with Sinkhorn-Knopp algorithm [30] the regularized WMD can be solved in $O(V^2)$ time, which is significantly lower than computing the original EMD that costs at least in $O(V^3 \log(V))$ [24].

3 RWMD-BASED CENTROID CLASSIFIER

In this section, we describe the proposed RWMD-CC method for short texts.

3.1 Formulation

For clarity, we first give a formulation of the text classification task.

We use $x_d \in \Delta^{\mathcal{V}}$ and $y_d \in \mathcal{Y} = \{1, 2, \cdots, L\}$ to respectively denote the normalized term frequency vector and the label of short text d, where $\mathcal{V}$ is the set of unique words with size V and L is the number of labels. Then, a training dataset of D short texts is described by $S = \{x_d, y_d\}_{d=1}^{d=D}$. Our goal is to learn a classifier from S for predicting future short texts, i.e., learning a prediction function $f(x|\theta) \rightarrow y$ with parameter θ.

3.2 RWMD-CC

We begin with the centroid-based classification methodology. The mechanism is that given L category centroids $\{\theta_l \in \Delta^{\mathcal{V}}\}_{l=1}^{l=L}$, we predict any future short text x' by finding its closest category centroid.

The goal now is to learn these L category centroids from the training dataset S. Inspired by the prior art [33] and based on the structural risk minimization principle, we design a loss function of RWMD-CC using hypothesis margin, which involves both training errors (> 0) and inverse training margins (< 0). With a squared Frobenius norm regularizer, the loss function with repsect to θ is defined to be:

$$\begin{aligned}\mathcal{L}(\theta) =& \frac{1}{D}\left\{\sum\nolimits_{d=1}^{D} \left|W_c^{\lambda}\Big(g(x_d), g(\theta_R)\Big) - W_c^{\lambda}\Big(g(x_d), g(\theta_M)\Big)\right|_{-}\right.\\ &\left. + \sum\nolimits_{d=1}^{D} \eta\left|W_c^{\lambda}\Big(g(x_d), g(\theta_R)\Big) - W_c^{\lambda}\Big(g(x_d), g(\theta_M)\Big)\right|_{+}\right\}\\ &+ \frac{\lambda'}{2}\|\theta\|_F^2,\end{aligned} \tag{6}$$

where $|z|_{-} = min\{z, 0\}$, $|z|_{+} = max\{z, 0\}$; θ_R and θ_M denote the nearest category centroids to x_d under the RWMD measure with same and different label, respectively; $\eta \geq 1$ is a constant parameter to balance the training errors and inverse training margins in the loss function; $\|\cdot\|_F^2$ is the squared Frobenius norm, and $\lambda' \in [0, 1]$ is a regularization parameter. Here, we introduce $g(\cdot)$ with the constant importance vector $\mathbf{w}$:

$$g(x) = \frac{\mathbf{w} \circ x}{\mathbf{w}^{\top} x},$$

where $\circ$ represents the Hadamard product, to reflect the importance of words for distinguishing the categories in different classification tasks and datasets.

With the category centroids $\hat{\theta}$ obtained by optimizing $\mathcal{L}(\theta)$, for any future short text x', the prediction function of RWMD-CC is defined by:

$$y = \underset{l=1,\cdots,L}{\arg\min}\, W_c^{\lambda}\left(g(x'), g(\hat{\theta}_l)\right). \tag{7}$$

Besides, to obtain normalized centroids and avoid constrained conditions, we use the softmax transformation $\mathfrak{s}(\mu)$ for centroids θ with auxiliary parameters μ:

$$\theta = \mathfrak{s}(\mu), \qquad \theta_{lv} = \frac{\exp(\mu_{lv})}{\sum_{i=1}^{V}\exp(\mu_{li})}. \tag{8}$$

Combining Eq.6 and Eq.8, we reach the following final loss function with respect to the auxiliary parameters μ:

$$\mathfrak{L}(\mu) = \mathcal{L}\Big(\mathfrak{s}(\mu)\Big). \tag{9}$$

3.3 Optimization

We use gradient descent to minimize the loss function $\mathfrak{L}(\mu)$. In the following, we first show the optimization process, and then provide the implementation details.

3.3.1 *The Optimization Algorithm of RWMD-CC.*

In gradient descent, given the gradient $\nabla\mathfrak{L}(\mu)$ and the learning rate φ, the parameters μ are updated by $\mu \leftarrow \mu - \varphi\nabla\mathfrak{L}(\mu)$. In the following, we mainly describe the computation of the gradient $\nabla\mathfrak{L}(\mu)$ and then outline the full algorithm.

The gradient $\nabla\mathfrak{L}(\mu)$. After some simple derivations, we show the gradient $\nabla\mathfrak{L}(\mu)$ as follows:

$$\begin{aligned}\nabla\mathfrak{L}(\mu) &= \frac{\partial\mathcal{L}\Big(\mathfrak{s}(\mu)\Big)}{\partial\mathfrak{s}(\mu)} \times \frac{\partial\mathfrak{s}(\mu)}{\partial\mu}\\ &= \theta \circ \frac{\partial\mathcal{L}(\theta)}{\partial\theta} - \theta\left(\theta^{\top}\frac{\partial\mathcal{L}(\theta)}{\partial\theta}\mathbf{I}_L\right),\end{aligned} \tag{10}$$

where $\mathbf{I}_L$ is the identity matrix with dimension L. The calculation of $\nabla\mathfrak{L}(\mu)$ only depends on $\frac{\partial\mathcal{L}(\theta)}{\partial\theta}$ that is given below.

[Computation of $\frac{\partial\mathcal{L}(\theta)}{\partial\theta}$]

For convenience, we re-write $\mathcal{L}(\theta)$ in Eq.6 as

$$\mathcal{L}(\theta) = \frac{1}{D}\left(\sum\nolimits_{d=1}^{D}\delta\Big(h(x_d, \theta)\Big)h(x_d, \theta)\right) + \frac{\lambda'}{2}\|\theta\|_F^2, \tag{11}$$

where

$$\delta(z) = \begin{cases}\eta & \text{if } z \geq 0,\\ 1 & \text{if } z < 0;\end{cases}$$

$$h(x_d, \theta) = W_c^{\lambda}\Big(g(x_d), g(\theta_R)\Big) - W_c^{\lambda}\Big(g(x_d), g(\theta_M)\Big).$$

Consequently, the gradient of $\mathcal{L}(\theta)$ with respect to each category centroid θ_l can be computed by:

$$\frac{\partial\mathcal{L}(\theta)}{\partial\theta_l} = \frac{1}{D}\left(\sum\nolimits_{d=1}^{D}\delta\Big(h(x_d, \theta)\Big)\frac{\partial h(x_d, \theta)}{\partial\theta_l}\right) + \lambda'\theta_l, \tag{12}$$

where

$$\frac{\partial h(x_d, \theta)}{\partial\theta_l} = \begin{cases}\frac{\partial W_c^{\lambda}\left(g(x_d), g(\theta_l)\right)}{\partial g(\theta_l)} \times \frac{\partial g(\theta_l)}{\partial\theta_l} & \text{if } \theta_R = \theta_l\\ -\left(\frac{\partial W_c^{\lambda}\left(g(x_d), g(\theta_l)\right)}{\partial g(\theta_l)} \times \frac{\partial g(\theta_l)}{\partial\theta_l}\right) & \text{if } \theta_M = \theta_l,\end{cases} \tag{13}$$

and

$$\frac{\partial g(\theta_l)}{\partial\theta_l} = \frac{\text{diag}(\mathbf{w}) - g(x)\mathbf{w}^{\top}}{\mathbf{w}^{\top}x}.$$

The computation of $\frac{\partial\mathcal{L}(\theta)}{\partial\theta_l}$ requires $\frac{\partial W_c^{\lambda}\left(g(x_d), g(\theta_l)\right)}{\partial g(\theta_l)}$, which, unfortunately, is intractably solved from the regularized WMD in Eq.4 directly. Inspired by the recent works on fast earth mover's distance computation [6, 7, 10], we change the regularized WMD

$W_c^\lambda\Big(g(x_d), g(\theta_l)\Big)$ to its dual problem and compute the gradient $\frac{\partial W_c^\lambda\left(g(x_d), g(\theta_l)\right)}{\partial g(\theta_l)}$ by solving the dual problem with well-known Sinkhorn-Knopp algorithm which is also used in [6, 7, 43]. We now show the details.

[Computation of $\frac{\partial W_c^\lambda\left(g(x_d), g(\theta_l)\right)}{\partial g(\theta_l)}$]

We first give the dual problem of the regularized WMD $W_c^\lambda\Big(g(x_d), g(\theta_l)\Big)$. Following [7], the Lagrange duality of regularized WMD $W_c^\lambda\Big(g(x_d), g(\theta_l)\Big)$ with dual variables $\alpha, \beta \in \mathcal{R}^V$ is given by:

$$
{}^{d}W_c^\lambda\Big(g(x_d), g(\theta_l)\Big) = \sup_{(\alpha,\beta)} \Bigg(\alpha^\top g(x_d) + \beta^\top g(\theta_l) - \sum_{i,j} \frac{e^{-\lambda\left(C(i,j)-\alpha_i-\beta_j\right)}}{\lambda}\Bigg). \tag{14}
$$

From [2], we know that the dual optimal β^* of Eq.14 is, in fact, the gradient $\frac{\partial W_c^\lambda\left(g(x_d), g(\theta_l)\right)}{\partial g(\theta_l)}$ that we need:

$$
\frac{\partial W_c^\lambda\Big(g(x_d), g(\theta_l)\Big)}{\partial g(\theta_l)} = \beta^*. \tag{15}
$$

Then, below we give the details of how to compute the dual optimal β^*. Following Proposition 2 of [7], when given $\mathbf{K} = e^{-\lambda C}$ which is the element-wise exponential of $-\lambda C$, there exists a pair of vectors $(\mathbf{u}, \mathbf{v}) \in \mathcal{R}^V \times \mathcal{R}^V$ such that the optimal solutions of regularized WMD, i.e., γ_λ^* of Eq.4 in Section 2.2 and its dual in Eq.14, i.e., β^*, are respectively given by

$$
\gamma_\lambda^* = \mathrm{diag}(\mathbf{u})\mathbf{K}\mathrm{diag}(\mathbf{v}),\ \beta^* = -\frac{\log(\mathbf{v})}{\lambda} + \frac{\log(\mathbf{v})^\top \mathbf{1}}{\lambda V}\mathbf{1}. \tag{16}
$$

where the term $\frac{\log(\mathbf{v})^\top \mathbf{1}}{\lambda V}\mathbf{1}$ is used to normalize β^* [7]. Besides, based on the Sinkhorn's theorem [30], the pair $(\mathbf{u}, \mathbf{v})$ can be recovered as a fixed point of the Sinkhorn map $(\mathbf{u}, \mathbf{v}) \leftarrow (g(x_d) \oslash \mathbf{K}\mathbf{v}, g(\theta_l) \oslash \mathbf{K}^\top \mathbf{u})$, where $\oslash$ represents the element-wise division. This fixed point iteration can also be written as a single update:

$$
\mathbf{v} \leftarrow g(\theta_l) \oslash \Big(\mathbf{K}^\top \Big(g(x_d) \oslash \mathbf{K}\mathbf{v}\Big)\Big).
$$

Hence, with vector $\mathbf{v}$, the gradient $\frac{\partial W_c^\lambda\left(g(x_d), g(\theta_l)\right)}{\partial g(\theta_l)}$ can be computed by combining Eq.15 and Eq.16:

$$
\frac{\partial W_c^\lambda\Big(g(x_d), g(\theta_l)\Big)}{\partial g(\theta_l)} = \beta^* = -\frac{\log(\mathbf{v})}{\lambda} + \frac{\log(\mathbf{v})^\top \mathbf{1}}{\lambda V}\mathbf{1}. \tag{17}
$$

In summary, we outline the detailed computation procedure of the gradient of the regularized WMD in *Algorithm 1*.
In regularized WMD, λ is the entropic regularization parameter as mentioned in Section 2.2. In our experiments, we use $\lambda = 10$ following [13].

Full algorithm. In summary, we outline the optimization procedure of RWMD-CC in *Algorithm 2*.

3.3.2 *Implementation Details.*

Algorithm 1 Gradient of the regularized WMD

```
Input: $g(x_d)$, $g(\theta_l)$, matrix $\mathbf{K}$, regularization parameter $\lambda > 0$;
Output: $\frac{\partial W_c^\lambda\left(g(x_d), g(\theta_l)\right)}{\partial g(\theta_l)}$.
Initialize $\mathbf{v} = \mathbf{1}$;
while $\mathbf{v}$ has not converged do
    $\mathbf{v} \leftarrow g(\theta_l) \oslash \Big(\mathbf{K}^\top \Big(g(x_d) \oslash \mathbf{K}\mathbf{v}\Big)\Big)$;
end while
$\frac{\partial W_c^\lambda\left(g(x_d), g(\theta_l)\right)}{\partial g(\theta_l)} \leftarrow -\frac{\log(\mathbf{v})}{\lambda} + \frac{\log(\mathbf{v})^\top \mathbf{1}}{\lambda V}\mathbf{1}$;
```

Algorithm 2 Optimization for RWMD-CC

```
Input: Distance matrix $C$, training dataset $S = \{x_d, y_d\}_{d=1}^{d=D}$, importance vector $\mathbf{w}$, regularization parameters $\lambda,\ \lambda' > 0$, and learning rate $\varphi$;
Output: $\mu = \{\mu_l\}_{l=1}^{l=L}$.
Initialize $\theta$;
Calculate $\mu$ by the inverse of Eq.8 with $\theta$;
Calculate $g(x_d)$ for $d = 1, \ldots, D$;
for $t = 1$ to $N_{iter}$ do
    Calculate $\theta$ using Eq.8 with $\mu$;
    Calculate $g(\theta_l)$ for $l = 1, \ldots, L$;
    for $d = 1$ to $D$ do
        for $l = 1$ to $L$ do
            Calculate $\frac{\partial W_c^\lambda\left(g(x_d), g(\theta_l)\right)}{\partial g(\theta_l)}$ using Algorithm 1;
            Calculate $\frac{\partial h(x_d, \theta)}{\partial \theta_l}$ using Eq.13;
        end for
    end for
    Calculate $\frac{\partial \mathcal{L}(\theta)}{\partial \theta_l}$ using Eq.12 for $l = 1, \ldots, L$;
    Calculate $\nabla \mathfrak{L}(\mu)$ using Eq.10;
    $\mu \leftarrow \mu - \varphi \nabla \mathfrak{L}(\mu)$;
end for
```

Initialization of category centroids θ. In RWMD-CC, we use *Class-Feature-Centroid* (CFC) [11] centroids as the initialization values of category centroids θ. In CFC centroids, the unnormalized mass of word v in category l is given by

$$
\theta_{lv} = b^{\frac{DF_v^l}{D_l}} \bullet \log(\frac{L}{CF_v}), \tag{18}
$$

where DF_v^l is word v's document frequency in category l; D_l is the number of documents in category l; CF_v is the number of categories containing word v; and b is a constant larger than one. Here, we set b to $e - 1.0$ (≈ 1.718) following [11].

Setting of importance vector $\mathbf{w}$. The constant importance vector $\mathbf{w}$ in RWMD-CC is set by using *balanced distributional concentration* (bdc) [37]. In bdc, the weight $\mathbf{w}_v$ of word v is given by:

$$
\mathbf{w}_v = 1 + \frac{\sum_{l=1}^{L} \frac{p(v|l)}{\sum_{l=1}^{L} p(v|l)} \log\left(\frac{p(v|l)}{\sum_{l=1}^{L} p(v|l)}\right)}{\log(L)},
$$

where $p(v|l) = \frac{TF_v^l}{TF^l}$ is the proportion of word v in its relevant category l, and TF_v^l, TF^l denote the frequency of word v in category l and the frequency sum of all words in category l, respectively.

3.4 Stochastic Optimization

The computation of the gradient $\frac{\partial \mathcal{L}(\theta)}{\partial \theta}$ needs to pass all training documents per-iteration and may be expensive for big datasets. To address this, we propose a stochastic optimization version of RWMD-CC, namely ***RWMD-CC$_s$***.

In RWMD-CC$_s$, we approximate noisy gradient of each category centroid θ_l by using a randomly selected subset S' per-iteration, instead of the whole training set S:

$$\frac{\partial \mathcal{L}(\theta)}{\partial \theta_l} \approx \frac{1}{D_{mini}} \left(\sum\nolimits_{\{x_d, y_d\} \in S'} \delta\Big(h(x_d, \theta)\Big) \frac{\partial h(x_d, \theta)}{\partial \theta_l} \right) + \lambda' \theta_l, \tag{19}$$

where $D_{mini} = |S'|$ is the mini-batch size. Since the size of S' is greatly less than that of S, i.e., $|S'| = D_{mini} \ll |S| = D$, the computation of gradients in RWMD-CC$_s$ is more effective than that in RWMD-CC, and we use RWMD-CC$_s$ in our experiments.

With the gradient $\frac{\partial \mathcal{L}(\theta)}{\partial \theta}$ approximated by Eq.19, $\nabla \mathfrak{L}(\mu)$ is calculated by Eq.10. In RWMD-CC$_s$, rather than using constant learning rate, we use *Adam* algorithm [14], which computes adaptive learning rate for each parameter, to update μ with the gradient $\nabla \mathfrak{L}(\mu)$.

3.5 Time Complexity

We now discuss the time complexity of RWMD-CC and RWMD-CC$_s$ for both training and testing phases, respectively. For clarity, we revisit some notations: L the number of category centroids, D the number of training documents, D_{mini} the mini-batch size in RWMD-CC$_s$, V the number of unique words; and introduce a new notation D_{avg} the average length of documents.

Training. For simplicity, we only discuss the per-iteration time complexity. In detail, the main time consuming part is the calculation of the gradient of the RWMD (step 9 in Algorithm 2). In our algorithms, we obtain the gradient of the RWMD between a document and a category centroid via the Sinkhorn-Knopp algorithm [6, 10], whose time complexity is $O(D_{avg}V)$. For RWMD-CC, we need to repeat this process between every document of training dataset and every category centroid, i.e., Eq.12, hence require DL times. Consequently, the per-iteration takes $O(DLD_{avg}V)$ time. Since one document in short texts only contains few words, e.g., $D_{avg} = 3.2$ in *Trec* dataset, each iteration of the training phase can be done in minutes. For RWMD-CC$_s$, we use the subset of training dataset with size D_{mini} to approximate the gradient of the objective, i.e., Eq.19, thus the time complexity of per-iteration is $O(D_{mini}LD_{avg}V)$, and per-iteration can be done in seconds.

Testing. When predicting a test document, both our RWMD-CC and RWMD-CC$_s$ only compute the RWMDs between the test document and all category centroids following the prediction function Eq.7. Hence the testing time complexity of our algorithms is linear with respect to L. However, for predicting a test document, WMD-based KNN classifiers, such as WMD+KNN, have to compare the WMDs between this test document and all documents of training datasets, so the testing time complexity of WMD-based KNN classifiers is a linear function of D. Obviously, our RWMD-CC and RWMD-CC$_s$ are much more efficient than WMD-based KNN classifiers. Actually, we can consider the centroid as a representative pseudo-document for each category. In this sense, the centroid classifier is a special case of K nearest neighbour (KNN) classifier with these pseudo-documents where $K = 1$.

We will further demonstrate the real run-time of training and testing phases in the experiment.

3.6 Generalization Error of RWMD-CC

Given a set of i.i.d. samples $S = \{x_d, y_d\}_{d=1}^{d=D}$, let $\hat{\theta}$ be the empirical risk minimizer

$$\begin{aligned} \hat{\theta} = \operatorname*{arg\,min}_{\theta \in \Theta} \bigg\{ & \hat{\mathbb{E}}_S \Big[\delta\Big(h(x, \theta)\Big) h(x, \theta) \Big] \\ & = \frac{1}{D} \Big(\sum\nolimits_{d=1}^{D} \delta\Big(h(x_d, \theta)\Big) h(x_d, \theta) \Big) \bigg\}. \end{aligned} \tag{20}$$

Further suppose $\Theta = \mathfrak{s} \circ (\Theta^o \times \cdots \times \Theta^o)$ is the composition of the softmax transformation $\mathfrak{s}$ and a L-valued function space with a base hypothesis space Θ^o of functions mapping into $\mathcal{R}^V$. The softmax transformation $\mathfrak{s}$ guarantees that θ_l lies in the simplex $\Delta^{\mathcal{V}}$.

Theorem 3.1. *For any $\tau > 0$, with probability at least $1 - \tau$, it holds that*

$$\begin{aligned} & \mathbb{E}\Big[\delta\Big(h(x, \hat{\theta})\Big) h(x, \hat{\theta})\Big] \le \inf_{\theta \in \Theta} \mathbb{E}\Big[\delta\Big(h(x, \theta)\Big) h(x, \theta)\Big] \\ & + \eta \left(256 V \sqrt{L^3} \|C\|_\infty \mathfrak{R}_D(\Theta^o) + 64 \sqrt{L^3} \frac{\log V}{\lambda} + \|C\|_\infty \sqrt{\frac{2 \log(1/\tau)}{D}} \right) \\ & + \|C\|_\infty \sqrt{\frac{2 \log(1/\tau)}{D}} \end{aligned} \tag{21}$$

with the constant $\|C\|_\infty = \max_{ij} C_{ij}$. $\mathfrak{R}_D(\Theta^o)$ is the Rademacher complexity [1] measuring the complexity of the hypothesis space Θ^o.

Remark. From Theorem 3.1, there is a constant error, which is introduced by regularized WMD, in risk bound. We can see that the constant error will vanish as $\lambda \to \infty$, which also coincides with the risk bound for the EMD with the standard convergence rate $O(1/\sqrt{D})$. However, as shown in [6], with λ larger, computing regularized WMD costs more time in same dimension. These mean that the value of λ reflects the trade-off between computation speed and approximation accuracy to some degree. Here, we set $\lambda = 10$ following [13].

4 RELATED WORK

In this section, we review some most related works on short text classification and classification using WMD.

4.1 Short Text Classification

Previous works on short text classification are mainly based on feature expansion. They can be roughly divided into two classes. One is to extract auxiliary context information using search engines [8, 29, 32]. Traditional classification algorithms are then applied to such expanded short texts, which are deemed as long pseudo-documents. An limitation is that such algorithms must require high quality search results. The other is to expand document features by employing external knowledgebases such as Wikipedia and WordNet [12, 21, 25, 35]. For example, [25] uses hidden topics of LDA learnt from Wikipedia to enrich document-level features. These algorithms perform well, however, they may be problematic when the pre-trained topics and concepts are inconsistent with the current dataset for certain applications.

Recently, some deep learning-based classifiers have been proposed [9, 36, 42]. To our knowledge, an early representative work is the character to sentence convolutional neural network (CharSCNN) [9]. As suggested by the name, the CharSCNN is based on CNN, and it exploits from character- to sentence-level information to perform short text classification. Other deep learning classifiers are also based on the CNN architecture, and the pre-trained word embeddings are used as the input.

Additionally, many researchers have developed generic short text topic models, e.g., biterm topic model (BTM) [5] and word network topic model [44]. They learn topic representations of short texts and then use a traditional algorithm, e.g., SVMs, as a downstream classifier. Such methodology empirically performed well on classification evaluations of short texts.

4.2 Classification Using WMD

Since the WMD is a measure of document distance, a most straightforward classification method is WMD+KNN [15]. A recent work [13] incorporates the supervision (i.e., labels) into a WMD learning procedure, leading to supervised WMD (S-WMD). Empirically, S-WMD+KNN performed better than WMD+KNN. In contrast to those methods, our RWMD-CC computes a representative centroid for each category. Roughly, RWMD-CC can reduce the number of calculating WMDs from D to L when predicting each test document.

5 EXPERIMENTS

In this section, we empirically compare our RWMD-CC against the existing baseline algorithms of short texts.

In early experiments, we have evaluated the performance of RWMD-CC and RWMD-CC$_s$ (the stochastic optimization version of RWMD-CC) on all the datasets, and found that the performance gap between RWMD-CC and RWMD-CC$_s$ is very small. That is, the stochastic optimization does not affect the performance of RWMD-CC. Hence we only show RWMD-CC$_s$ in the experiments.

5.1 Experimental Settings

Datasets. We choose three datasets of short texts, including *Trec*[1], *biomedical*[2] and *StackOverFlow*[2]. For all datasets, we remove the standard stopwords, words with term frequency less than 5, and empty documents. The statistics of datasets are shown in Table 2. Besides, we employ pre-trained *GloVe*[3] word embeddings and randomly set embeddings of the words that are not included in the vocabulary of pre-trained word embeddings. We use the cosine distance, i.e., $1 - cos(w_i, w_j)$, which is used to measure the distance between any word embeddings pair $\{w_i, w_j\}$, as the cost function c in the regularized WMD to compute the distance matrix C.

Table 2: Statistics of datasets. *#doc*: the number of documents. *#word*: the number of unique words. D_{avg}: the average document length. *#label*: the number of categories. *Rate*: the word cover rate of GloVe word embeddings.

Dataset	*#doc*	*#word*	D_{avg}	*#label*	*Rate* (%)
Trec	5,864	1,058	3.2	6	99.3
biomedical	19,970	4,472	7.3	20	94.7
StackOverFlow	19,768	2,651	3.9	20	78.7

Baseline algorithms. We choose four existing algorithms for comparison. Details are given as below.

- **TFIDF + *K*NN**: We use tf-idfs as the representations of short texts, and then perform KNN as a downstream classifier with counting the similarities between short texts by cosine measure.
- **BTM + SVMs**: We first learn the topic representations of short texts using BTM [5], and then perform SVMs[4] as a downstream classifier.
- **WMD + *K*NN**: This is the first use of WMD for classification in its original paper [15]. The code is available on the net[5].
- **S-WMD + *K*NN**: S-WMD [13] is a supervised extension of WMD, and also uses stochastic optimization. The code is available on the net[6].

For the KNN-based methods, the neighborhood number K is tested from $\{1, \ldots, 19\}$. For BTM+SVMs, WMD+KNN and S-WMD+KNN, we use default parameter settings in their papers and codes, except the mini-batch size of S-WMD+KNN, which is set to 256.

Evaluation metrics. The comparison is based on two popular classification metrics, i.e., macro-averaging F1 and micro-averaging F1. The macro-averaging F1 (Macro-F1) and micro-averaging F1 (Micro-F1) are the two different types of average of F1 score and used to measure the classification performance for the whole corpus. The F1 score is a combined form of precision (π) and recall (ρ):

$$\pi_l = \frac{TP_l}{TP_l + FP_l}, \quad \rho_l = \frac{TP_l}{TP_l + FN_l}, \quad F1_l(\pi_l, \rho_l) = \frac{2\pi_l \rho_l}{\pi_l + \rho_l},$$

where TP_l (True Positives) is the number of documents assigned correctly to category l; FP_l (False Positives) is the number of documents that do not belong to category l but are assigned incorrectly to it by classifier; FN_l (False Negatives) is the number of documents that belong to category l but are assigned incorrectly to other ones by classifier. Specifically, Micro-F1 and Macro-F1 are the F1 score of the whole dataset and the average F1 score of all categories, respectively.

Parameter settings. For RWMD-CC$_s$, in our experiments, its parameters are empirically set as follows: $\lambda = 10$, $b = e - 1.0$, $\eta = 60.0$, $\lambda' = 0.001$, $D_{mini} = 256$ and $N_{iter} = 50$. All the experiments are carried on a Linux server with Intel Xeon 3.10 GHz CPU and 16G memory.

[1] http://trec.nist.gov/data/tweets/

[2] https://github.com/jacoxu/STC2

[3] https://nlp.stanford.edu/projects/glove/. The 100-dimensional vectors trained on Wikipedia 2014 and Gigaword 5 are used.

[4] http://scikit-learn.org/stable/

[5] https://github.com/mkusner/wmd

[6] https://github.com/gaohuang/S-WMD

Table 3: Results of classification accuracy (mean±std) with 5-fold cross validation. The best results of short text classifiers are highlighted in boldface. "‡" means that the gain of RWMD-CC$_s$ is statistically significant at 0.05 level.

Metric	Dataset	BoW		WMD		
		TFIDF + *K*NN	BTM + SVMs	WMD + *K*NN	S-WMD + *K*NN	RWMD-CC$_s$
Micro-F1	*Trec*	$0.8281 \pm 0.0063^{\ddagger}$	$0.6223 \pm 0.0122^{\ddagger}$	0.9389 ± 0.0086	$\mathbf{0.9543 \pm 0.0031}$	0.9425 ± 0.0081
	biomedical	$0.5596 \pm 0.0072^{\ddagger}$	$0.5129 \pm 0.0071^{\ddagger}$	$0.6063 \pm 0.0212^{\ddagger}$	$0.6255 \pm 0.0056^{\ddagger}$	$\mathbf{0.6763 \pm 0.0057}$
	StackOverFlow	$0.6613 \pm 0.0083^{\ddagger}$	$0.5880 \pm 0.0053^{\ddagger}$	$0.7914 \pm 0.0113^{\ddagger}$	$0.7847 \pm 0.0130^{\ddagger}$	$\mathbf{0.8470 \pm 0.0062}$
Macro-F1	*Trec*	$0.8477 \pm 0.0092^{\ddagger}$	$0.6074 \pm 0.0178^{\ddagger}$	0.9444 ± 0.0063	$\mathbf{0.9585 \pm 0.0018}$	0.9428 ± 0.0087
	biomedical	$0.5534 \pm 0.0077^{\ddagger}$	$0.5195 \pm 0.0065^{\ddagger}$	$0.5971 \pm 0.0225^{\ddagger}$	$0.6226 \pm 0.0052^{\ddagger}$	$\mathbf{0.6677 \pm 0.0060}$
	StackOverFlow	$0.6616 \pm 0.0079^{\ddagger}$	$0.6095 \pm 0.0044^{\ddagger}$	$0.7899 \pm 0.0111^{\ddagger}$	$0.7842 \pm 0.0130^{\ddagger}$	$\mathbf{0.8460 \pm 0.0057}$

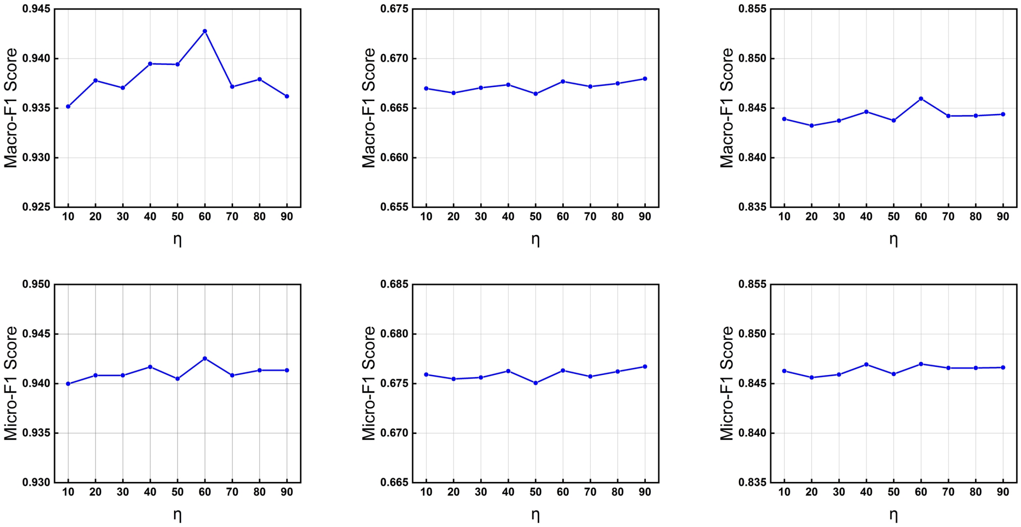

Figure 1: Classification performance by varying η on *Trec* (left), *biomedical* (middle) and *StackOverFlow* (right).

5.2 Performance Comparison

For all methods, we perform a 5-fold cross validation and report the average scores. The classification results over all datasets are shown in Table 3. Overall, our RWMD-CC$_s$ performs better than the baselines in most cases. Some observations are made as follows:
Comparing RWMD-CC$_s$ against baselines: First, RWMD-CC$_s$ consistently outperforms BoW-based methods on Micro-F1 and Macro-F1 scores, especially the scores of RWMD-CC$_s$ are even about 0.18 higher than those of TFIDF+*K*NN on *StackOverFlow*. Second, RWMD-CC$_s$ performs better than other WMD-based methods on *biomedical* and *StackOverFlow*, e.g., about 0.06 higher than S-WMD+*K*NN on *StackOverFlow*. RWMD-CC$_s$ only has a small decrease compared with S-WMD+*K*NN on *Trec*. The possible reason is the category-imbalanced nature of *Trec*, in which the number of documents of each category is $\{1300, 1309, 95, 1288, 1005, 867\}$, and thus some documents of other categories may be misclassified into the category, which contains fewer documents, for centroid-based classifiers.

Comparing WMD-based methods against BoW-based methods: WMD-based methods (including WMD+*K*NN, S-WMD+*K*NN and RWMD-CC$_s$) significantly dominate BoW-based methods (including TFIDF+*K*NN and BTM+SVMs) on two evaluation metrics on all three datasets. For example, the Micro-F1 scores of WMD+*K*NN beat TFIDF+*K*NN by 0.11, 0.05 and 0.13 on *Trec*, *biomedical*, and *StackOverFlow*, respectively. It means that for short text classification the WMD-based classifiers are much better than the classifiers based on BoW representations. The result is expected since the WMD can effectively measure semantic distances among documents with word embeddings and hence reduce the document similarity misalignment, especially for short texts. This is also an evidence of that the WMD is a better distance measure than the distances or similarities based on BoW.
Observation about word embeddings: The classification performance of RWMD-CC$_s$ is insensitive to the word embeddings. For instance, on *StackOverFlow* the performance of RWMD-CC$_s$ is still very competitive even the cover rate of pre-trained GloVe word

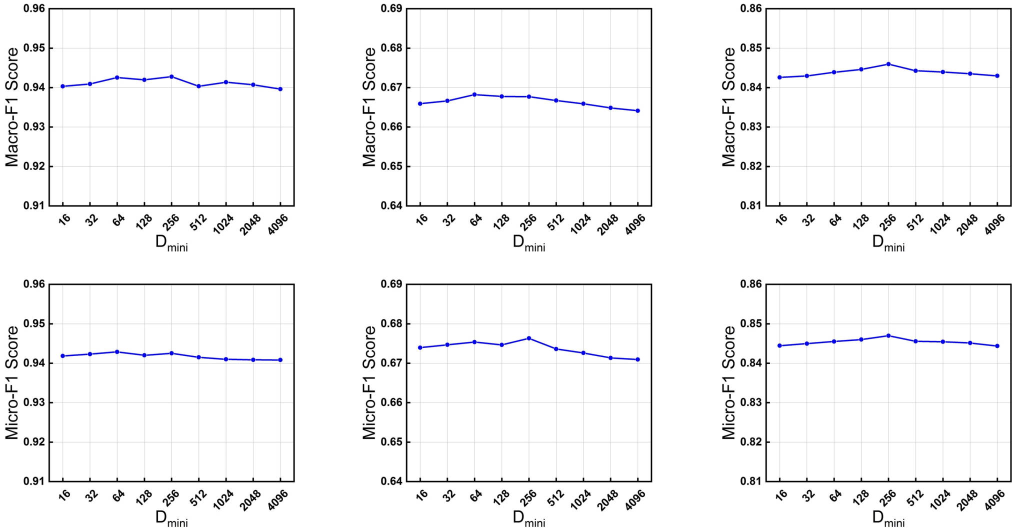

Figure 2: Classification performance by varying D_{mini} on *Trec* (left), *biomedical* (middle) and *StackOverFlow* (right).

Table 4: Time cost of WMD+KNN, S-WMD+KNN and RWMD-CC$_s$ on *Trec* (top section), *biomedical* (middle section) and *StackOverFlow* (bottom section). Init. Time: time cost during initializing parameters. Iter. Time: per-iteration time in the training phase. Test Time: time cost of predicting all test documents. h: hours, m: minutes, s: seconds.

Method	**Init. Time**	**Iter. Time**	**Test Time**
WMD+KNN	–	–	8m 49s
S-WMD+KNN	21m 57s	2s	8m 17s
RWMD-CC$_s$	–	1s	18s
WMD+KNN	–	–	1h 46m
S-WMD+KNN	4h 34m	3s	1h 35m
RWMD-CC$_s$	–	3s	5m 38s
WMD+KNN	–	–	3h 4m
S-WMD+KNN	4h 52m	4s	2h 44m
RWMD-CC$_s$	–	2s	3m 40s

embeddings is only 78.7%. Therefore, we argue that RWMD-CC$_s$ is practical even big collections of word embeddings lack.

5.3 Efficiency Comparison

We compare the running time of the three WMD-based methods, including RWMD-CC$_s$, WMD+KNN and S-WMD+KNN.

We compare these algorithms in initializing time, per-iteration training time and testing time. To be fair, we use the following settings: (a) for all datasets, we randomly select 20% from full dataset as the test set and remain 80% as the training set; (b) for all methods, we set the entropic regularization parameter λ of regularized WMD to 10; (c) specifically for RWMD-CC$_s$ and S-WMD+KNN, we set the mini-batch size to 256 in their training phases. All experiments are carried with **single thread** on a Linux server with Intel Xeon 3.10 GHz CPU and 16G memory. Table 4 shows the running time results averaged on 100 runs. Some observations are made as follows:

Comparison of testing time: RWMD-CC$_s$ is significantly faster than WMD+KNN and S-WMD+KNN when predicting test documents, e.g., WMD+KNN and S-WMD+KNN cost about 3 hours on *StackOverFlow* dataset, as comparison, RWMD-CC$_s$ merely 4 minutes. This is consistent with the time complexity analysis mentioned in the section 3.5. That is because when predicting a test document, RWMD-CC$_s$ only calculates the WMDs between this test document and the category centroids, WMD+KNN and S-WMD+KNN, in contrast, have to compute the WMDs between it and all documents of training datasets. We thus argue that RWMD-CC$_s$ is a more practical choice, especially for real world applications, where fast response is required.

Comparison of training time: Thanks to stochastic optimization, both the training phases of RWMD-CC$_s$ and S-WMD+KNN spend very little time. Therefore, the training of RWMD-CC$_s$ is also efficient for big corpora.

Comparison of initializing time: S-WMD+KNN takes huge time during initializing parameters, e.g., about 5 hours on *StackOverFlow* dataset, in contrast, our RWMD-CC$_s$ spends no time for initialization. This again demonstrates RWMD-CC$_s$ is more practical.

5.4 Evaluation of Parameters

In this subsection, we empirically evaluate two crucial parameters of RWMD-CC$_s$, including the trade-off parameter η and mini-batch size D_{mini}.

We first evaluate the impact of different η values over the set $\{10, 20, \cdots, 90\}$. The results are shown in Figure 1. Roughly, both Macro-F1 and Micro-F1 scores of RWMD-CC$_s$ have a small change in all settings with the η value varying from 10 to 90. That is to say, RWMD-CC$_s$ is insensitive to η, making it more robust in real world applications. Besides, the best scores are achieved at $\eta = 60$ in most cases. We thus fix $\eta = 60$ in our experiments, and suggest choosing the value of η from 50 to 70 for short texts in practice.

Then, we examine the classification performance of different D_{mini} values over the set $\{16, 32, \cdots, 4096\}$. As the results shown in Figure 2, we argue that RWMD-CC$_s$ is insensitive to the mini-batch size D_{mini}. Furthermore, the best performance is achieved when $D_{mini} = 256$. Thus, in the experiments, we use $D_{mini} = 256$ as the default setting of RWMD-CC$_s$, which is also the suggested value in practice.

6 CONCLUSION

In this paper, we propose a RWMD-CC method for extremely short text classification. Our RWMD-CC is based on the popular document distance, e.g., WMD, which measures the distance at the semantic level. We learn a representative semantic centroid for each category, and predict any text sample depending on the WMDs between this sample and all semantic centroids. To achieve this, we build a loss function with hypothesis margin that is based on the structural risk minimization principle. For fast optimization, we follow the spirit of the regularized WMD and stochastic optimization. Experimental results indicate that our RWMD-CC outperforms the state-of-the-art baseline methods, and more importantly it is much more efficient than the existing WMD-based KNN methods during prediction.

A STATISTICS LEARNING BOUNDS

We establish the proof of Theorem 3.1 in this section. For simpler notation, given a sequence $S = \{x_d, y_d\}_{d=1}^{d=D}$ of i.i.d. training samples, we denote the empirical risk $\hat{R}_S$ and expected risk R as

$$\hat{R}_S(\theta) = \hat{\mathbb{E}}_S\Big[\delta\big(h(x,\theta)\big)h(x,\theta)\Big] = \frac{1}{D}\Big(\sum\nolimits_{d=1}^{D}\delta\big(h(x_d,\theta)\big)h(x_d,\theta)\Big),$$
$$R(\theta) = \mathbb{E}\Big[\delta\big(h(x,\theta)\big)h(x,\theta)\Big]. \tag{22}$$

where

$$\delta(z) = \begin{cases} \eta & \text{if } z \geq 0, \\ 1 & \text{if } z < 0; \end{cases} \qquad h(x_d,\theta) = W_c^\lambda\big(x_d,\theta_R\big) - W_c^\lambda\big(x_d,\theta_M\big),$$

θ_R and θ_M denote the nearest category centroids to x_d under RWMD measure with same label and different label, respectively.

Lemma A.1 (Lemma B.1 of [10]). *Let $\hat{\theta}$, $\theta^* \in \Theta$ be the minimizer of the empirical risk $\hat{R}_S$ and expected risk R, respectively. Then*

$$R(\hat{\theta}) \leq R(\theta^*) + 2\sup_{\theta\in\Theta}|R(\theta) - \hat{R}_S(\theta)|.$$

To bound the risk for $\hat{\theta}$, we need to establish a uniform concentration bound for the loss function $\mathcal{L}(\theta)$ in Eq.11. Therefore, we define a loss function space induced by the hypothesis space Θ as

$$\mathcal{H} = \Big\{\mathfrak{h} : x \mapsto \delta\big(h(x,\theta)\big)h(x,\theta) : \theta \in \Theta\Big\}. \tag{23}$$

The uniform concentration bound depends on the "complexity" of $\mathcal{H}$, which is measured by the empirical *Rademacher complexity* defined below.

Definition A.2 (Rademacher Complexity [1]). Let $\mathcal{F}$ be a family of mapping from $\mathcal{Z}$ to $\mathcal{R}$, and $S = (z_1, \ldots, z_D)$ a fixed sample from $\mathcal{Z}$. The empirical Rademacher complexity of $\mathcal{F}$ with respect to S is defined as

$$\hat{\mathfrak{R}}_S(\mathcal{F}) = \mathbb{E}_\sigma\left[\sup_{f\in\mathcal{F}}\frac{1}{D}\sum_{d=1}^{D}\sigma_d f(z_d)\right], \tag{24}$$

where $\sigma = (\sigma_1, \ldots, \sigma_d, \ldots, \sigma_D)$, with σ_d's independent uniform random variables taking values in $\{+1, -1\}$. σ_d's are called the Rademacher random variables. The Rademacher complexity is defined by taking expectation with respect to the samples S,

$$\mathfrak{R}_D(\mathcal{F}) = \mathbb{E}_S\Big[\hat{\mathfrak{R}}_S(\mathcal{F})\Big].$$

Theorem A.3. *Let $\mathcal{H} = \Big\{\mathfrak{h} : x \mapsto \delta\big(h(x,\theta)\big)h(x,\theta) : \theta \in \Theta\Big\}$. For any $\tau > 0$, with probability at least $1 - \tau$, the following holds for all $\mathfrak{h} \in \mathcal{H}$,*

$$\mathbb{E}[\mathfrak{h}] - \hat{\mathbb{E}}_S[\mathfrak{h}] \leq 2\mathfrak{R}_D(\mathcal{H}) + (\eta + 1)\|C\|_\infty\sqrt{\frac{\log(1/\tau)}{2D}}, \tag{25}$$

where $\mathfrak{R}_D(\mathcal{H})$ is Rademacher complexity of the loss function space $\mathcal{H}$ associated to Θ, and $\|C\|_\infty = \max_{ij} C_{ij}$.

Following the definition of $\mathcal{H}$, $\mathbb{E}[\mathfrak{h}] = R(\theta)$ and $\hat{\mathbb{E}}_S[\mathfrak{h}] = \hat{R}_S(\theta)$. Hence, a uniform control for the deviation of the empirical risk from the expected risk is given by the Theorem A.3.

Theorem A.4 (McDiarmid's Inequality). *Let $S = \{X_1, \ldots, X_D\}$ be i.i.d. random variables with size D from $\mathscr{X}$. Assume there exists $C > 0$ such that $f : \mathscr{X}^D \rightarrow \mathcal{R}$ satisfies the following stability condition*

$$|f(x_1, \ldots, x_d, \ldots, x_D) - f(x_1, \ldots, x_d', \ldots, x_D)| \leq C \tag{26}$$

for all $d = 1, \ldots, D$ and any $x_1, \ldots, x_D, x_d' \in \mathscr{X}$. Then for any $\varepsilon > 0$, denoting $f(X_1, \ldots, X_D)$ by $f(S)$, it holds that

$$\mathbb{P}(f(S) - \mathbb{E}[f(S)] \geq \varepsilon) \leq \exp\Big(-\frac{2\varepsilon^2}{DC^2}\Big). \tag{27}$$

Lemma A.5. *Let the constant $\|C\|_\infty = \max_{ij} C_{ij}$ is the maximum in the distance matrix C, then $0 \leq W_c^\lambda(\cdot,\cdot) \leq \|C\|_\infty$.*

Proof. For any x and θ_l, let γ_λ^* be the relaxed transport plan that solves (5), then

$$W_c^\lambda(x,\theta_l) = \Big\langle\gamma_\lambda^*, C\Big\rangle \leq \|C\|_\infty\sum_{v_1,v_2}\gamma_{v_1,v_2} = \|C\|_\infty.$$

□

Lemma A.6. *Let the constant $\|C\|_\infty = \max_{ij} C_{ij}$ is the maximum in the distance matrix C, then $-\|C\|_\infty \leq h(x,\theta) \leq \|C\|_\infty$, $-\|C\|_\infty \leq \delta\big(h(x,\theta)\big)h(x,\theta) \leq \eta\|C\|_\infty$.*

Lemma A.6 can be easily proved following the definitions of $\delta(z)$ and $h(x,\theta)$ with Lemma A.5.

Proof of Theorem A.3. Inspired by the proof of Theorem B.3 of [10], the Theorem A.3 can be easily proved by using Lemma A.6 and Theorem A.4. The details can be found in the proof of Theorem B.3 of [10]. □

Now, we establish the Rademacher complexity $\mathfrak{R}_D(\mathcal{H})$. For each instance (x_d, y_d) from S, we denote the category centroids θ_l with the same label y_d as θ_l^+ and the ones with a different label than y_d as θ_l^-, then $\theta_R \in \{\theta_l^+\}$ and $\theta_M \in \{\theta_l^-\}$. Thus, we can re-write the function $h(x, \theta) = W_c^\lambda(x, \theta_R) - W_c^\lambda(x, \theta_M)$ as

$$h(x, \theta) = \left(\min_{\theta_l^+} \left\{ W_c^\lambda(x, \theta_l^+) \right\} - \min_{\theta_l^-} \left\{ W_c^\lambda(x, \theta_l^-) \right\} \right). \tag{28}$$

Then, we define a function $\mathfrak{m} : [-\|C\|_\infty, \|C\|_\infty] \to \mathcal{R}$ and a real function space $\mathcal{F}$ as follows:

$$\mathfrak{m} : z \mapsto z\delta(z), \tag{29}$$

$$\mathcal{F} = \left\{ \mathfrak{f} : (x, \vartheta) \mapsto W_c^\lambda(x, \vartheta) \right\}. \tag{30}$$

Theorem A.7. *Let the function $\mathfrak{m}$ and the function space $\mathcal{F}$ be defined as above. For the loss function space $\mathcal{H}$, it holds that*

$$\mathfrak{R}_D(\mathcal{H}) \le 16\eta\sqrt{L^3}\mathfrak{R}_D(\mathcal{F}). \tag{31}$$

Theorem A.8 (Rademacher Vector Contraction Inequality [22]). *Let $\mathcal{F}$ be a class of real functions, and $\mathcal{H} \subset \mathcal{F} = \mathcal{F}_1 \times \cdots \times \mathcal{F}_K$ be a K-valued function class. If $\mathfrak{t} : \mathcal{R}^K \mapsto \mathcal{R}$ is a L_m-Lipschitz continuous function and $\mathfrak{t}(\mathbf{0}) = 0$, then*

$$\hat{\mathfrak{R}}_S(\mathfrak{t} \circ \mathcal{H}) \le \sqrt{2} L_m \sum_{k=1}^{K} \hat{\mathfrak{R}}_S(\mathcal{F}_k). \tag{32}$$

Proof of Theorem A.7. With the function $\mathfrak{m}$, the loss function space $\mathcal{H} = \left\{ \mathfrak{h} : x \mapsto \delta\big(h(x, \theta)\big) h(x, \theta) : \theta \in \Theta \right\}$ can be written as $\mathcal{H} = \mathfrak{m} \circ \mathcal{G}$, where $\mathcal{G} = \left\{ \mathfrak{g} : x \mapsto h(x, \theta) : \theta \in \Theta \right\}$ is a function space.

Now, we first show the function $\mathfrak{m}$ is Lipschitz continuous with constant η, then give $\mathfrak{R}_D(\mathcal{G})$, which is the Rademacher complexity of $\mathcal{G}$.

With the definition of function $\mathfrak{m}$ and δ, $\eta \ge 1$, it holds that

$$|\mathfrak{m}(z_1) - \mathfrak{m}(z_2)| = \left\{ \begin{array}{ll} |\eta z_1 - \eta z_2| = \eta |z_1 - z_2| & \text{if } z_1, z_2 \ge 0 \\ |\eta z_1 - z_2| = \eta z_1 + (-z_2) & \text{if } z_1 \ge 0, z_2 < 0 \\ |z_1 - \eta z_2| = (-z_1) + \eta z_2 & \text{if } z_1 < 0, z_2 \ge 0 \\ |z_1 - z_2| & \text{if } z_1, z_2 < 0 \end{array} \right\} \le \eta |z_1 - z_2|.$$

Thus, $\mathfrak{m}$ is Lipschitz with constant η and $\mathfrak{m}(0) = 0$.

As next step, we establish the Rademacher complexity $\mathfrak{R}_D(\mathcal{G})$. Following Eq.28, the functions $\mathfrak{g}$ contained in $\mathcal{G}$ can be expressed as the sum of two terms of identical form, one multiplied by -1, such as

$$x \mapsto \min_{\theta_l} \left\{ W_c^\lambda(x, \theta_l) \right\}, \tag{33}$$

where the minimum is taken over at most L terms, L denotes the number of category centroids. Let $\mathcal{M}$ be the function space, in which the form of functions is Eq.33. Note that $\mathfrak{R}_D(\sum_{k=1}^K \mathcal{F}_k) \le \sum_{k=1}^K \mathfrak{R}_D(\mathcal{F}_k)$ holds for all real function classes $\mathcal{F}_k$ by Theorem 12 of [1] due to the triangle inequality. Further, following Theorem 12 of [1] multiplying a function space by -1 does not change its Rademacher complexity. Therefore, we can upper bound $\mathfrak{R}_D(\mathcal{G})$ by twice the complexity of the function space $\mathcal{M}$:

$$\mathfrak{R}_D(\mathcal{G}) \le 2\mathfrak{R}_D(\mathcal{M}). \tag{34}$$

Following [28], the function that calculates the minimum of L values as follows,

$$(z_1, \ldots, z_L) \mapsto \min\{z_1, \ldots, z_L\},$$

is Lipschitz continuous with constant $\sqrt{8L}$. We apply Theorem A.8 to the $\sqrt{8L}$-Lipschitz continuous function min and the function space

$$\underbrace{\mathcal{F} \times \cdots \times \mathcal{F}}_{L \text{ copies}}.$$

It holds

$$\hat{\mathfrak{R}}_S(\mathcal{M}) \le \sqrt{2}\sqrt{8L} \sum_{l=1}^{L} \hat{\mathfrak{R}}_S(\mathcal{F}) = 4\sqrt{L^3}\hat{\mathfrak{R}}_S(\mathcal{F}). \tag{35}$$

Then, following Theorem 12 of [1] and combining Eq.34 with Eq.35, we have

$$\mathfrak{R}_D(\mathcal{H}) = \mathfrak{R}_D(\mathfrak{m} \circ \mathcal{G}) \le 2\eta\mathfrak{R}_D(\mathcal{G}) \le 16\eta\sqrt{L^3}\mathfrak{R}_D(\mathcal{F}).$$

□

Before establishing $\mathfrak{R}_D(\mathcal{F})$, we provide some useful properties of regularized WMD.

Lemma A.9 (Lemma 2 of [43]). *For two discrete measures p_1, $p_2 \in \Delta^{\mathcal{V}}$, the regularized WMD $W_c^\lambda(p_1, p_2)$ and the EMD $W_c(p_1, p_2)$ satisfy the following relationship,*

$$W_c(p_1, p_2) \le W_c^\lambda(p_1, p_2) \le W_c(p_1, p_2) + \frac{2}{\lambda} \log V,$$

where $V = |\mathcal{V}|$.

Then, we define a function space $\mathcal{F}_{EMD}$ corresponding to the family of the EMD as follows:

$$\mathcal{F}_{EMD} = \left\{ \mathfrak{f}_{EMD} : (x, \vartheta) \mapsto W_c(x, \vartheta) \right\}.$$

From Theorem 4 of [43], the Rademacher complexities of $\mathcal{F}$ and $\mathcal{F}_{EMD}$ satisfy

$$\mathfrak{R}_D(\mathcal{F}) \le \mathfrak{R}_D(\mathcal{F}_{EMD}) + \frac{\log V}{\lambda}. \tag{36}$$

In the following, we relate the Rademacher complexity of $\mathcal{F}_{EMD}$ with the hypothesis space Θ^o by using the similar processing method in [10] and give the proof of Theorem 3.1.

Proof of Theorem 3.1. Consider a function space $\mathcal{F}_{EMD}$ with a softmax transformation $\mathfrak{s}$

$$\mathcal{F}_{EMD} = \left\{ f_{EMD} : (x, \mu) \mapsto W_c(\mathfrak{s}(x), \mathfrak{s}(\mu)) \right\},$$

From Proposition B.10 of [10], we know that the $\mathcal{R}^V \times \mathcal{R}^V \mapsto \mathcal{R}$ map $\iota : (z, z') \mapsto W_c(\mathfrak{s}(z), \mathfrak{s}(z'))$ is a $2\sqrt{2}\|C\|_\infty$-Lipschitz continuous function and $\iota(\mathbf{0}, \mathbf{0}) = 0$. We apply Theorem A.8 to the $2\sqrt{2}\|C\|_\infty$-Lipschitz continuous function ι, and it holds

$$\hat{\mathfrak{R}}_S(\mathcal{F}_{EMD}) \le \sqrt{2} \cdot 2\sqrt{2}\|C\|_\infty V \hat{\mathfrak{R}}_S(\Theta^o). \tag{37}$$

The conclusion of Theorem 3.1 follows by combining Eq.37 with Eq.36, Theorem A.7, Theorem A.3 and Lemma A.1. □

ACKNOWLEDGMENTS

We would like to acknowledge support for this project from the National Natural Science Foundation of China (NSFC) (No.61876071, No.61602204).

REFERENCES

[1] Peter L. Bartlett and Shahar Mendelson. 2002. Rademacher and Gaussian Complexities: Risk Bounds and Structural Results. *Journal of Machine Learning Research* 3 (2002), 463–482.

[2] Dimitris Bertsimas and John N Tsitsiklis. 1997. *Introduction to Linear Optimization*. Athena Scientific Belmont, MA.

[3] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet Allocation. *Journal of Machine Learning Research* 3 (2003), 993–1022.

[4] Vladimir Igorevich Bogachev and Aleksandr Viktorovich Kolesnikov. 2012. The Monge-Kantorovich Problem: Achievements, Connections, and Perspectives. *Russian Mathematical Surveys* 67, 5 (2012), 785–890.

[5] Xueqi Cheng, Xiaohui Yan, Yanyan Lan, and Jiafeng Guo. 2014. BTM: Topic Modeling over Short Texts. *IEEE Transactions on Knowledge and Data Engineering* 26, 12 (2014), 2928–2941.

[6] Marco Cuturi. 2013. Sinkhorn Distances: Lightspeed Computation of Optimal Transport. In *Neural Information Processing Systems*. 2292–2300.

[7] Marco Cuturi and Arnaud Doucet. 2014. Fast Computation of Wasserstein Barycenters. In *International Conference on Machine Learning*. 685–693.

[8] Honghua (Kathy) Dai, Lingzhi Zhao, Zaiqing Nie, Ji-Rong Wen, Lee Wang, and Ying Li. 2006. Detecting Online Commercial Intention (OCI). In *International Conference on World Wide Web*. 829–837.

[9] Cícero Nogueira dos Santos and Maíra Gatt. 2014. Deep Convolutional Neural Networks for Sentiment Analysis of Short Texts. In *International Conference on Computational Linguistics*. 69–78.

[10] Charlie Frogner, Chiyuan Zhang, Hossein Mobahi an Mauricio Araya-Polo, and Tomaso Pogglo. 2015. Learning with a Wasserstein Loss. In *Neural Information Processing Systems*. 2053–2061.

[11] Hu Guan, Jingyu Zhou, and Minyi Guo. 2009. A Class-Feature-Centroid Classifier for Text Categorization. In *International Conference on World Wide Web*. 201–210.

[12] Xia Hu, Nan Sun, Chao Zhang, and Tat-Seng Chua. 2009. Exploiting Internal and External Semantics for the Clustering of Short Texts Using World Knowledge. In *ACM International Conference on Web Search and Data Mining*. 353–362.

[13] Gao Huang, Chuan Guo, Matt J. Kusner, Yu Sun, Kilian Q. Weinberger, and Fei Sha. 2016. Supervised Word Mover's Distance. In *Neural Information Processing Systems*. 4862–4870.

[14] Diederik P Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. *arXiv preprint arXiv:1412.6980* (2014).

[15] Matt J. Kusner, Yu Sun, Nicholas I. Kolkin, and Kilian Q. Weinberger. 2015. From Word Embeddings To Document Distances. In *International Conference on Machine Learning*. 957–966.

[16] Ximing Li, Jinjin Chi, Changchun Li, Jihong Ouyang, and Bo Fu. 2016. Integrating Topic Modeling with Word Embeddings by Mixtures of vMFs. In *International Conference on Computational Linguistics*. 151–160.

[17] Ximing Li, Changchun Li, Jinjin Chi, and Jihong Ouyang. 2018. Short Text Topic Modeling by Exploring Original Documents. *Knowledge and Information Systems* 56, 2 (2018), 443–462.

[18] Ximing Li, Changchun Li, Jinjin Chi, Jihong Ouyang, and Chenliang Li. 2018. Dataless Text Classification: A Topic Modeling Approach with Document Manifold. In *ACM International Conference on Information and Knowledge Management*. 973–982.

[19] Ximing Li, Yue Wang, Ang Zhang, Changchun Li, Jinjin Chi, and Jihong Ouyang. 2018. Filtering out the Noise in Short Text Topic Modeling. *Information Sciences* 456 (2018), 83–96.

[20] Ximing Li and Bo Yang. 2018. A Pseudo Label based Dataless Naive Bayes Algorithm for Text Classification with Seed Words. In *International Conference on Computational Linguistics*. 1908–1917.

[21] Yuhua Li, David McLean, Zuhair A. Bandar, James D. O'Shea, and Keeley Crockett. 2006. Sentence Similarity Based on Semantic Nets and Corpus Statistics. *IEEE Transactions on Knowledge and Data Engineering* 16, 8 (2006), 1138–1150.

[22] Andreas Maurer. 2016. A Vector-contraction Inequality for Rademacher Complexities. In *International Conference on Algorithmic Learning Theory*. 3–17.

[23] Tomas Mikolov, Wen tau Yih, and Geoffrey Zweig. 2013. Linguistic Regularities in Continuous Space Word Representations. In *Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 2292–2300.

[24] Ofir Pele and Michael Werman. 2009. Fast and Robust Earth Mover's Distances. In *International Conference on Computer Vision*. 460–467.

[25] Xuan-Hieu Phan, Le-Minh Nguyen, and Susumu Horiguchi. 2008. Learning to Classify Short and Sparse Text & Web with Hidden Topics from Large-scale Data Collections. In *International Conference on World Wide Web*. 91–100.

[26] Yossi Rubner, Carlo Tomasi, and Leonidas J. Guibas. 1998. A Metric for Distributions with Applications to Image Databases. In *International Conference on Computer Vision*. 59–66.

[27] Yossi Rubner, Carlo Tomasi, and Leonidas J Guibas. 2000. The Earth Mover's Distance as a Metric for Image Retrieval. *International Journal of Computer Vision* 40, 2 (2000), 99–121.

[28] Petra Schneider, Michael Biehl, and Barbara Hammer. 2009. Adaptive Relevance Matrices in Learning Vector Quantization. *Neural Computation* 21, 12 (2009), 3532–3561.

[29] Dou Shen, Rong Pan, Jian-Tao Sun, Jeffrey Junfeng Pan, Kangheng Wu, Jie Yin, and Qiang Yang. 2006. Query Enrichment for Web-query Classification. *ACM Transactions on Information Systems* 24, 3 (2006), 320–352.

[30] Richard Sinkhorn. 1967. Diagonal Equivalence to Matrices with Prescribed Row and Column Sums. *The American Mathematical Monthly* 74, 4 (1967), 402–405.

[31] Yang Song, Dengyong Zhou, and Li wei He. 2012. Query Suggestion by Constructing Term-Transition Graphs. In *ACM International Conference on Web Search and Data Mining*. 353–362.

[32] Aixin Sun. 2012. Short Text Classification Using Very Few Words. In *ACM SIGIR Conference on Research and Development in Information Retrieval*. 1145–1146.

[33] Songbo Tan and Xueqi Cheng. 2007. Using Hypothesis Margin to Boost Centroid Text Classifier. In *ACM Symposium on Applied Computing*. 398–403.

[34] Cédric Villani. 2008. *Optimal Transport: Old and New*. Springer Berlin Heidelberg.

[35] Fang Wang, Zhongyuan Wang, Zhoujun Li, and Ji-Rong Wen. 2014. Concept-based Short Text Classification and Ranking. In *ACM International Conference on Information and Knowledge Management*. 1069–1078.

[36] Peng Wang, Bo Xu, Jiaming Xu, Guanhua Tian, Cheng-Lin Liu, and Hongwei Hao. 2016. Semantic Expansion Using Word Embedding Clustering and Convolutional Neural Network for Improving Short Text Classification. *Neurocomputing* 174, Part B (2016), 806–814.

[37] Tao Wang, Yi Cai, Ho-fung Leung, Zhiwei Cai, and Huaqing Min. 2015. Entropy-Based Term Weighting Schemes for Text Categorization in VSM. In *IEEE International Conference on Tools with Artificial Intelligence*. 325–332.

[38] Yang Wang, Xuemin Lin, Lin Wu, Wenjie Zhang, Qing Zhang, and Xiaodi Huang. 2015. Robust Subspace Clustering for Multi-view Data by Exploiting Correlation Consensus. *IEEE Transactions on Image Processing* 24, 11 (2015), 3939–3949.

[39] Yang Wang, Lin Wu, Xuemin Lin, and Junbin Gao. 2018. Multiview Spectral Clustering via Structured Low-Rank Matrix Factorization. *IEEE Transactions on Neural Networks and Learning Systems* 29, 10 (2018), 4833–4843.

[40] Yang Wang, Wenjie Zhang, Lin Wu, Xuemin Lin, Meng Fang, and Shirui Pan. 2016. Iterative Views Agreement: An Iterative Low-rank based Structured Optimization Method to Multi-view Spectral Clustering. In *International Joint Conference on Artificial Intelligence*. 2153–2159.

[41] Jianshu Weng, Ee-Peng Lim, Jing Jiang, and Qi He. 2010. TwitterRank: Finding Topic-sensitive Influential Twitterers. In *ACM International Conference on Web Search and Data Mining*. 261–270.

[42] Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level Convolutional Networks for Text Classification. In *Neural Information Processing Systems*. 649–657.

[43] Peng Zhao and Zhi-Hua Zhou. 2018. Label Distribution Learning by Optimal Transport. In *AAAI Conference on Artificial Intelligence*. 4506–4513.

[44] Yuan Zuo, Jichang Zhao, and Ke Xu. 2016. Word Network Topic Model: A Simple but General Solution for Short and Imbalanced Texts. *Knowledge and Information Systems* 48, 2 (2016), 379–398.