Simultaneous Learning of Sentence Clustering and Class Prediction for Improved Document Classification

Minyoung Kim

Department of Electronics & IT Media Engineering, Seoul National University of Science & Technology, Seoul, Korea

ljfis

Abstract

In document classification it is common to represent a document as the so called bag-of-words form, which is essentially a global term distribution indicating how often certain terms appear in a text. Ignoring the spatial statistics (i.e., where in a text they appear) can potentially lead to a suboptimal solution. The key motivation or assumption in this paper is that there may exist underlying segmentation of sentences in a document, and perhaps this partitioning might be intuitively appealing (e.g., each group corresponds to a particular sentiment or gist of arguments). If the segmentation is known somehow, terms belonging to the same/different groups can potentially be treated in an equal/different manner for classification. Based on the idea, we build a novel document classification model comprised of two parts: a sentence tagger that predicts the group labels of sentences, and a classifier that forms the input features as a weighted term frequency vector that is aggregated from all sentences but weighed differently cluster-wise according to the prediction in the first model. We suggest an efficient learning strategy for this model. For several benchmark document classification problems, we demonstrate that the proposed approach yields significantly improved classification performance over several existing algorithms.

Keywords: Machine learning, Document classification, Sequence labeling, Term weighting

1. Introduction

Increased availability of document/text data from social networking services and Internet blogs or news articles demands efficient and accurate algorithms for automatic processing, organization, and indexing of text data. Document classification, the task of classifying a document into one of the pre-defined classes, is certainly the core problem for these higher-level goals. In the fields of natural language processing, machine learning, and data mining, the problem has received significant attention for recent years. The applications are also wide-spread, including among others: predicting users' reading preferences [1, 2], news article classification [3–5], categorizing web pages [6–8], and e-mail spam filtering [9–12], to name just a few.

In document classification it is common to form the so called bag-of-words (BOW) representation, which is essentially a global term distribution, namely how often certain terms appear in a text. Also referred to as the *term frequency (tf)* vector, the BOW representation

Received: Feb. 13, 2017
 Revised : Mar. 20, 2017
 Accepted: Mar. 23, 2017

Correspondence to: Minyoung Kim
 (mikim@seoultech.ac.kr)
 ©The Korean Institute of Intelligent Systems

©This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

characterizes a document by the most frequent words that were used, and thus they often carry the central theme/message of the document. One caveat, however, is that all terms are considered equally important (e.g., regardless of whether stop words or key nouns/verbs). The term weighting approaches remedy this issue by assigning different weights to terms according to their importance. The *inverse document frequency* (*idf*), the most popular term weighting scheme, weighs each term by the reciprocal of the document frequency (i.e., the number of training documents that contain the term), which effectively diminishes the contribution of stop words (e.g., articles *a* or *the*).

Later on, considerable research attempts have been introduced to estimate more discriminative data-driven term weightings for better prediction performance. Most of these approaches exploit the class labels in the training data, specifically collecting information-theoretic statistics such as χ^2 -metric, information gain, and odds ratio, to name a few [13, 14]. For instance, the recent *tf-rf* strategy [15] weighs each term according to the relevance score: in binary classification, defined as the ratio between the numbers of positive and negative training documents that contain the term. The intuition is that terms with high relevance scores are considered as most salient terms for deciding class labels.

Although these term weighting approaches have achieved considerable improvement, they are still based on the BOW representation for the *entire* document. That is, how often certain terms appear does matter, but where they appear is completely ignored. One key insight related to this issue is as follows: the same term behaves differently depending on where it belongs to (e.g., it has stronger impact when used in a phrase with positive sentiment, while much weaker when found in a negative phrase, or vice versa). Consequently, ignoring the spatial statistics can potentially lead to a suboptimal solution. Obviously a document is a sequentially structured collection of sentences, and taking into account such a spatial information is important for improving prediction performance.

The key motivation or assumption in this paper is that there may exist underlying segmentation of sentences in a document, where it is not necessary but this partitioning may be intuitively appealing (e.g., each group corresponds to a particular sentiment or gist of arguments). Here we choose sentences as basic units for clustering although it can be extended to paragraphs or others. If the segmentation is known somehow, terms belonging to the same/different groups can potentially be treated in an equal/different manner for classification: for instance, we may assign identical/different weights (contributions) for the terms

in the final feature representation.

Based on this motivating idea, we build a novel document classification model which is comprised of two parts. In the first part we aim to segment sentences into a pre-defined set of clusters. This can be done by building and learning a sentence tagger that predicts the group labels of sentences. Specifically we use the probabilistic model conditional random field (CRF) [16, 17] for its ability to capture the sequential statistical dependency of output (group label) variables that can yield high predictive power. The sentence labels required to train the CRF model are typically unavailable in the training data, and we rather estimate them from the error signal evaluated in the second model part. The second part is a conventional classifier (e.g., logistic regression model). However, it is specialized in that we form the input features as the weighted term frequencies that are aggregated from all sentences, but weighed differently cluster-wise according to the prediction in the first model.

We suggest an efficient learning strategy for this model, which also accounts for uncertainty in the group label prediction in an effective way. In particular, we learn the model in an alternating fashion: learn one component model while fixing the other part, and furthermore utilizing the predictions made by the latter. More specifically, we learn the first CRF model using the group labels of the sentences that minimize the classification loss in the second part, and at the same time we learn the classification model using the sentence segmentation information predicted by the CRF model. For several benchmark document classification problems, we demonstrate that the proposed approach yields significantly improved classification performance over several existing algorithms.

2. Problem Setup and Prior Approaches

In this section we formally setup the problem of document classification, and briefly review some related approaches that specifically utilize the term weighting paradigm.

As a training set we are given n samples of document-class pairs, $\mathcal{D} = \{(D_i, c_i)\}_{i=1}^n$. We define the vocabulary as a set of all terms (or words) in the training corpus $\mathcal{V} = \{\omega_k\}_{k=1}^V$ where we denote by V the size of the vocabulary. We deal with the binary classification (The multi-class problems can be straightforwardly handled by standard binarization tricks (e.g., the one-vs-others treatment in [18]), thus $c \in \{+1, -1\}$). For the class label statistics, we let p_k^1 be the number of positive documents in the training set that contain the term ω_k , and n_k^0 be the number of negative training documents where ω_k is absent.

Similarly, p_k^0 and n_k^1 can be defined. The numbers of positive and negative documents can be derived as: $n_+ = p_k^1 + p_k^0$, $n_- = n_k^1 + n_k^0$ for all k (obviously, $n = n_+ + n_-$).

For learning a classifier, it is useful to represent a document as a feature vector. The *tf* representation is one of the most popular methods. Denoted by $\phi(D) = [tf_1, tf_2, \dots, tf_V]^T$ with tf_k being the frequency counts of the term ω_k in D , the *tf* vector is a BOW representation that does not capture any spatial information like where in D the words appear. Motivated by the intuition that certain words are more important than others in classifying documents, several term weighting strategies have been proposed. The key idea is to weigh the entries of the *tf* vector differently. More specifically the final feature representation $x(D)$ is formed by $x(D) = b \otimes \phi(D)$, where $b = [b_1, b_2, \dots, b_V]^T \in \mathbb{R}^V$ is the term weighting vector, and the operator $\otimes$ indicates element-wise product.

For instance, the well-known *idf* scheme forms b as the reciprocals of the numbers of training documents that contain the terms, which has the effect of focusing more on unique terms and ignoring stop words. Some recent approaches further exploit the class labels in the training set to find a term weighting vector that is more discriminative and specialized to the particular classification task at hand. Most of these approaches typically utilize the information-theoretic statistics (e.g., information gain or χ^2 -statistics), essentially judging the importance of individual words by certain imbalance measures on the positive/negative class proportions. Below we list the five most popular term weighting schemes with formal definitions. Details or motivations of these weighting formulations can be found in the accompanying references.

- *idf*: Inverse document frequency; $b_k = \log(\frac{n}{p_k^1 + n_k^1})$.
- *chi2*: The χ^2 -statistics based term weighting [13, 14]; $b_k = \frac{n(p_k^1 n_k^0 - p_k^0 n_k^1)^2}{n_+ n_- (p_k^1 + n_k^1)(p_k^0 + n_k^0)}$.
- *ig*: The information-gain term weighting [13, 14]; $b_k = \frac{p_k^1}{n} \log(\frac{(n/n_+)p_k^1}{p_k^1 + n_k^1}) + \frac{p_k^0}{n} \log(\frac{(n/n_-)p_k^0}{p_k^0 + n_k^0}) + \frac{n_k^1}{n} \log(\frac{(n/n_+)n_k^1}{p_k^1 + n_k^1}) + \frac{n_k^0}{n} \log(\frac{(n/n_-)n_k^0}{p_k^0 + n_k^0})$.
- *rf*: The relevance frequency term weighting [19]; $b_k = \log(2 + \frac{p_k^1}{n_k^1})$, which considers the ratio between the numbers of positive and negative documents that contain the term.
- *or*: The odds-ratio term weighting [13, 14]; $b_k = \frac{p_k^1 n_k^0}{p_k^0 n_k^1}$, which is similar to *rf*, but also takes into account absence of the term.

In practice the prediction performance of the above term weighting methods is often significantly improved over the non-weighted *tf* representation. However, it appears that there is no single robust one that performs well consistently: one strategy performs well on some specific cases while fails severely in other situations. Motivated by this, in our prior work [20] we have proposed several reasonable methods to combine multiple different term weighting vectors that yield a robust classifier for different types of datasets. Although these approaches were successful in many situations, the main drawback is again that a single term weighting vector is constructed for the entire document in a pure BOW manner, failing to account for the spatial (sequential) statistics of the words in a document. In the next section, we introduce a novel document classification model that takes into account the sequential dependency of the sentences constituting the document.

3. Our Approach

We begin with illustrating an example that motivates our approach. Let us consider the sentiment classification task. Even though a document as a whole can be classified as a single sentiment (either positive or negative), some of the individual paragraphs or sentences may express an opposite sentiment as well as different themes from others. It is often the case that the same word is used to express positive meaning in one place, but has a negative impact in another, depending on which type of paragraph/sentence it belongs to. For instance, the word “war” can have a negative effect when it appears in a sentence that contains phrases like “cause death” or “destroy buildings”. However, if it is found in a paragraph/sentence whose main theme is *war against negative things* (e.g., drugs or terrors), or if it is used solely for highlighting some positive meaning (e.g., *we always pursue peace even in the war*), then its negative contribution has to be reduced in the overall sentiment decision. On the other hand, the negative impact should be exaggerated in the former case.

Motivated by this, our main idea is to cluster the sentences (We opt for sentence as a minimal unit to apply the same term weighting vector, but it can be a paragraph or others) in a document into several groups (say, there are K different groups), and for each group $j \in \{1, \dots, K\}$, we apply the same term weighting vector $b^{(j)}$ to the terms that belong to the group j . To do so, one has to decide the group label of each sentence. This should be done in an unsupervised manner due to the expensive cost of human labeling. Also, unlike conventional clustering

that takes samples (sentences in our case) independently with one another, we rather adopt a sequence prediction model (e.g., CRFs [16]) to take into account the sequential statistical dependency among sentences (e.g., two adjacent sentences are more likely to belong to the same group). Detailed model architecture and the learning strategy are described in the subsequent sections.

3.1 Model Architecture

For each training instance (document D_i and its class label c_i), we let T_i be the number of sentences comprising D_i ($i = 1, \dots, n$). We often drop the sample index subscript i (e.g., D instead of D_i) for notational simplicity. For a given document $D = s_1, \dots, s_T$ where s_t indicates the t -th sentence in the sequential order, we compute the term frequency vectors, one for each individual sentence, and denote them by $\Phi = \phi(s_1), \dots, \phi(s_T)$. That is, Φ is a length- T sequence of V -dimensional tf vectors of sentences. We introduce latent variables of group labels that the sentences belong to, $Y = y_1, \dots, y_T$ where y_t is the group label of s_t . We denote the total number of sentence groups by K , so $y_t \in \{1, \dots, K\}$.

Ultimately we aim to model the document-class distribution $P(c|D)$ for a given document $D = s_1, \dots, s_T$. Using simple algebra, the model can be written as:

$$P(c|D) = \sum_Y P(Y|D) \cdot P(c|Y, D). \quad (1)$$

In (1) the summation is over all possible sentence label sequences, which amounts to exponential (K^T) possible instances. Later we will discuss in detail how to circumvent this computational issue via slice-wise independent approximation. The model (1) is comprised of two components: the first one $P(Y|D)$ is a sentence label predictive model while the second part is the document classification model that takes both tf vectors and labels of sentences as input covariates. In what follows we describe each model in greater details.

The first model $P(Y|D)$ is essentially a *sequence* prediction model, and we adopt the popular conditional random fields (CRFs) [16]. The CRF models are successful in diverse sequence prediction problems for their effective modeling of the sequential dependency among the samples in a sequence. The main idea of the CRFs is to represent a conditional distribution as a log-linear model, where the linear feature function is designed to capture the input-output functional relationship as well as the sequentially structured dependency over the out-

put variables. Specifically in our sentence label prediction model, we introduce linear weight parameter vectors $\theta_j \in \mathbb{R}^V$ ($j = 1, \dots, K$), where each θ_j is applied uniformly to all sentences belonging to the group j . We also define the temporal dependency score by the parameter $\theta_{j,k}$ for two adjacent sentence labels j and k ($j, k = 1, \dots, K$). Then the model can be formally written as follows:

$$P(Y|D) \propto \exp \left(\sum_{t=1}^T \sum_{j=1}^K I(y_t = j) \theta_j^\top \phi(s_t) + \sum_{t=2}^T \sum_{j,k=1}^K I(y_t = j, y_{t-1} = k) \theta_{j,k} \right). \quad (2)$$

We let θ be the whole parameters of the model, that is, $\theta = \{\{\theta_j\}_{j=1}^K, \{\theta_{j,k}\}_{j,k=1}^K\}$.

The second model $P(c|Y, D)$ decides the class label of the document, and we consider the sentence labels Y discovered from the first part as input covariates. Specifically we form a feature vector $\psi(Y, D)$ as:

$$\psi(Y, D) = \sum_{t=1}^T \sum_{j=1}^K I(y_t = j) b^{(j)} \otimes \phi(s_t), \quad (3)$$

where $\{b^{(j)}\}_{j=1}^K$ are the term weighting vectors. In (3) each $b^{(j)}$ is applied to all sentences that are labeled as j , and thus considered as a term weighting vector responsible for the sentence group j . Although it is possible to learn $\{b^{(j)}\}$ entirely from data, this may lead to overfitting. Rather we use the fixed term weighting vectors as described in Section 2 (i.e., *idf*, *chi2*, *ig*, *rf*, and *or* vectors). We then have $K = 5$. In this way we urge the model to partition a document into several groups, where a specific term weighting vector is applied to each group. With the feature vector in (3) as an input, we define the classification model as a simple logistic classifier:

$$P(c = +1|Y, D) = \frac{\exp(w^\top \psi(Y, D))}{1 + \exp(w^\top \psi(Y, D))}, \quad (4)$$

where $w \in \mathbb{R}^V$ is the parameter vector of the model.

3.2 Learning Strategy

Although it is ideal to directly optimize (1) with respect to the model parameters θ and w , this becomes computationally infeasible due to the exponentially (in the number of sentences in a document) many term to be considered. Rather we propose a learning strategy similar to the expectation maximization

(EM) [21], but the parameter optimization alternates between two models. Note that each partial optimization has to deal with the difficult expectation over all possible Y 's (e.g., the objective function for optimizing the second model (over w) is $\mathbb{E}[P(c|Y, D)]$ where the expectation is with respect to the CRF predictive distribution $P(Y|D)$). We approximate the objective by finding the most plausible instance Y (In fact we do soft label assignment for each y_t instead of hard assignment as discussed shortly), and fix it during the parameter learning. The other optimization is done similarly by replacing the expectation with the most plausible point estimate. The details are described in the below.

First, we fix the CRF parameters θ in $P(Y|D)$ and optimize the logistic classifier over w . Instead of using the most plausible label sequence Y (i.e., point estimate), we further exploit the prediction uncertainty by adopting the soft label assignment probabilities $P(y_t|D)$ (K -dim vector) for each t . The sentence-wise label distribution $P(y_t|D)$ can be found easily by the forward-backward inference algorithm in the CRF model. The technical details can be found in [16]. Once we evaluate $P(y_t|D)$ for all t , the feature vector $\psi(Y, D)$ in (3) can be extended to the label-averaged one, denoted by $\psi(D)$ (with no dependency on Y now):

$$\psi(D) = \sum_{t=1}^T \sum_{j=1}^K P(y_t = j|D) b^{(j)} \otimes \phi(s_t). \quad (5)$$

Unlike using a single point estimate Y in (3), all plausible labels (i.e., high $P(y_t = j|D)$) are effectively accounted for in (5).

Then using (5) as input feature vectors, we can learn w from standard logistic regression training. It is the maximum likelihood estimation, and all the training instances are considered. Also, it is typical to add the regularization term $\|w\|_2^2$ in the optimization for better generalization performance. More formally, we solve:

$$\max_w \sum_{i=1}^n \left(c_i w^\top \psi(D_i) - \log(1 + e^{c_i w^\top \psi(D_i)}) \right) + \gamma \|w\|_2^2, \quad (6)$$

where $\gamma \geq 0$ is the constant that trades off the model regularizer against the log-likelihood.

Next, with w fixed in $P(c|Y, D)$, we learn the CRF parameters θ . In the original model (1), one has to marginalize out Y with respect to the unnormalized distribution $P(c|Y, D)$. Unfortunately, this is computationally infeasible, and another possibility is to approximate it by the single instance Y that maximizes $P(c|Y, D)$. However, finding the single best instance also leads

to a highly difficult combinatorial optimization. Instead, we aim to estimate the sentence-wise label distributions $P(y_t|D)$'s in (5). That is, we regard as optimization variables the soft assignment probability vectors $\xi_t = [P(y_t = 1), \dots, P(y_t = K)]^\top$ for $t = 1, \dots, T$, and maximize the log-likelihood of the logistic regression model. More specifically, for each training sample i , we solve:

$$\max_{\{\xi_t\}_{t=1}^T, \psi} c_i w^\top \psi - \log \left(1 + \exp(c_i w^\top \psi) \right) \quad (7)$$

$$\text{s.t. } \psi = \sum_{t=1}^T \sum_{j=1}^K \xi_t(j) b^{(j)} \otimes \phi(s_t) \quad (8)$$

$$\xi_t \geq 0, \sum_{j=1}^K \xi_t(j) = 1, t = 1, \dots, T. \quad (9)$$

Here (8) indicates that the feature vector ψ for the logistic regression is parameterized by the soft assignment vectors ξ_t 's. Also, (9) enforces ξ_t 's to be probability distributions for all t . In this way we relax the problem into a continuous optimization that can be solved easily by the simple projected gradient ascent method: for each t , (i) update $\xi_t \leftarrow \xi_t + \eta \nabla_{\xi_t} h$ where h is the objective function in (7) and $\eta (> 0)$ is the learning rate, and (ii) project ξ_t to the probability simplex by zeroing out all negative entries of ξ_t and dividing it by the sum of the entries. The maximizer vectors ξ_t 's are then used in the CRF learning. In the CRF training [16], since we need training samples of paired sequences (Y, D) , we generate (sample) the label sequence Y from ξ_t 's optimized in the previous step. The CRF training is done straightforwardly with the sampled label sequences.

Overall, we alternate the two steps discussed so far (i.e., learning w with θ fixed, and learning θ with w fixed) until convergence.

4. Evaluation

In this section the prediction performance of the proposed approach is tested on several document classification datasets. We note that the experimental setups and datasets used are very similar to those in our prior work [20] where we briefly describe them below. Our approach is contrasted with various term weighting methods (discussed in Section 2) including the strategies to combine these base term weighting vectors proposed in our prior work [20]. Specifically we compare our approach with two previous weight mixing approaches: (i) loss minimization with the convex hull constraint (denoted by *mix*-

ch) and (ii) mini-max robust classifier formulation (referred to as *mini-max*). The features learned in these methods are fed into the SVM classifier for test prediction. Our approach of simultaneous learning of the sentence label tagger and the document class predictor is denoted by *lab-class*.

We particularly consider three popular benchmark datasets in text classification: Reuters-21578, WebKB, and 20 Newsgroups. The Reuters 21578 dataset (<http://www.daviddlewis.com/resources/testcollections/reuters21578/>) consists of documents on the Reuters newswire in 1987. For the R8 dataset comprised of documents from 8 classes including *crude*, *earn*, *ship*, and so on, we collect about 7,000 documents with vocabulary size approximately 16,000. The WebKB dataset (<http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-20/www/data/>) contains approximately 4,000 web pages from various universities in US. The task is to classify documents in one of the four classes: *student*, *faculty*, *course*, and *project*. The number of unique terms in the dataset is about 8,000. The 20 Newsgroups (<http://people.csail.mit.edu/jrennie/20Newsgroups/>) is a relatively large dataset comprised of about 20,000 newsgroup documents classified into one of the 20 different subject categories. Some of the class categories are considerably overlapped taxonomically, making accurate prediction difficult. The corpus consists of about 70,000 terms.

We apply the conventional pre-processing for the data, mainly the word stemming that replaces all grammatical variations of words with root words. For this purpose we use the famous Porter's Stemmer algorithm [22]. For each of these datasets, we randomly split the data into 50%/50% training/test sets, repeated for 10 random runs to report the average performance. As mentioned earlier, we transformed the multi-class problems into multiple binary classification tasks by the one-vs-others treatment.

Since the class distributions in the datasets are considerably imbalanced (even more severe due to the one-vs-others binarization), we employ the F -score as a performance measure for the competing classification methods. Although it is well-known in the information retrieval literature, here we briefly describe how the F -scores can be computed. The F -score basically harnesses both precision and recall scores of the classifier. The precision (p) is the proportion of the true positives (TP) out of the predicted positives (PP), that is, $p = \frac{TP}{PP}$, while the recall (r) is the ratio of the true positives out of the entire positive instances, $r = \frac{TP}{WP}$ where WP is the number of positive documents in the test set. The F -score is then defined as $\frac{2pr}{p+r}$.

Due to the binarization of the original multi-class prob-

lems, we need a way to average the multiple F -scores from the individual problems. One reasonable approach introduced in [19], also adopted in this paper, is in two folds: (i) micro-averaged F -score defined as $\frac{2PR}{P+R}$ where $P = \frac{\sum_c TP(c)}{\sum_c PP(c)}$ and $R = \frac{\sum_c TP(c)}{\sum_c WP(c)}$ are based on the averaged positives (here $TP(c)$, $PP(c)$, and $RP(c)$ are from the c -th binary problem) and (ii) macro-averaged F -score defined to be $\frac{2P'R'}{P'+R'}$ $P' = (1/K) \sum_c p(c)$ and $R' = (1/K) \sum_c r(c)$ with the averaged precision and recall (here $p(c)$ and $r(c)$ are the precision and the recall for the c -th problem).

We report the prediction results of the competing approaches in Table 1. For the F -scores averaged over 10 random repetitions, the best performing single term-weighting approaches (among *idf*, *chi2*, *ig*, *rf*, and *or* strategies) as well as two mixed term-weighting methods (*mix-ch* and *mini-max*) in [20] are depicted to be compared with our approach (*lab-class*). The results indicate that the proposed approach performs the best among the competing term weighting strategies and mixing algorithms, consistently for all datasets. This can be attributed to the fact that our approach effectively takes into account the spatial term/word information via the two-stage modeling: sentence-wise segmentation for tagging theme/sentiment group labels for sentences, and the classification strategy that applies different term weightings according to the estimated theme groups.

5. Conclusion

In this paper we have addressed the drawback of the conventional document classification methods that are based on the BOW representation. Our approach effectively deals with the spatial statistics of words in a document by incorporating a sentence segmentation model. To account for the sequential dependency of the sentence theme labels, we have utilized the CRF structured tagging model, while the document classifier model exploits the predicted sentence labels in its prediction by applying label-wise different term weightings. Each of these two components in our model is learned from each other by the principle: best (soft) labels are optimized in one component, then exploited in learning the other. The proposed alternating learning algorithm turns out to be effective in accurate classification. Empirically, the proposed approach yields statistically significant performance improvement over existing term weighting strategies. While we have used fixed base term weighting vectors in the model training, it is also interesting to see the model behavior when the base vectors are also learned

Table 1. *F*-scores

Methods	Reuters-21578		WebKB		20-Newsgroup	
	Micro <i>F</i> -score	Macro <i>F</i> -score	Micro <i>F</i> -score	Macro <i>F</i> -score	Micro <i>F</i> -score	Macro <i>F</i> -score
Single TW	0.8453 $\pm$ 0.0073	0.8460 $\pm$ 0.0071	0.8984 $\pm$ 0.0033	0.8990 $\pm$ 0.0070	0.5958 $\pm$ 0.0058	0.5961 $\pm$ 0.0053
<i>mix-ch</i>	0.9132 $\pm$ 0.0050	0.9090 $\pm$ 0.0077	0.9012 $\pm$ 0.0035	0.9046 $\pm$ 0.0011	0.6613 $\pm$ 0.0005	0.6595 $\pm$ 0.0025
<i>mini-max</i>	0.9087 $\pm$ 0.0076	0.9136 $\pm$ 0.0052	0.9190 $\pm$ 0.0040	0.9219 $\pm$ 0.0046	0.6742 $\pm$ 0.0004	0.6622 $\pm$ 0.0009
<i>lab-class</i>	0.9257 $\pm$ 0.0064	0.9268 $\pm$ 0.0071	0.9258 $\pm$ 0.0037	0.9289 $\pm$ 0.0036	0.6967 $\pm$ 0.0005	0.6875 $\pm$ 0.0013

Single TW indicates the best performing single term-weighting strategy among five methods in Section 2. The proposed approach is denoted as *lab-class*.

simultaneously. This is left as our future work.

Conflict of Interest

No potential conflict of interest relevant to this article was reported.

Acknowledgements

This study was supported by Seoul National University of Science & Technology.

References

- [1] P. M. Domingos and M. J. Pazzani, "Beyond independence: conditions for the optimality of the simple Bayesian classifier," in *Proceedings of 1996 International Conference on Machine Learning*, Bari, Italy, 1996, pp. 105-112.
- [2] K. Lang, "Newsweeder: learning to filter netnews," in *Proceedings of 1995 International Conference on Machine Learning*, Tahoe City, CA, pp. 331-339.
- [3] T. Joachims, "Text categorization with support vector machines: learning with many relevant features," in *Proceedings of 10th European Conference on Machine Learning*, Chemnitz, Germany, 1998, pp. 137-142.
- [4] J. R. Li and K. Yang, "News clustering system based on text mining," in *Proceedings of 2010 IEEE International Conference on Advanced Management Science*, Chengdu, China, 2010. <http://dx.doi.org/10.1109/ICAMS.2010.5553112>
- [5] A. N. Chy, M. H. Seddiqui, and S. Das, "Bangla news classification using naive Bayes classifier," in *Proceedings of 16th International Conference on Computer and Information Technology*, Khulna, Bangladesh, 2013. <http://dx.doi.org/10.1109/ICCITech.2014.6997369>
- [6] M. Craven, D. DiPasquo, D. Freitag, A. McCallum, T. Mitchell, K. Nigam, and S. Slattery, "Learning to extract symbolic knowledge from the World Wide Web," in *Proceedings of the 15th National Conference on Artificial Intelligence*, Madison, WI, 1998.
- [7] J. Shavlik and T. Eliassi-Rad, "Intelligent agents for Web-based tasks: an advice-taking approach," in *Proceedings of the 15th National Conference on Artificial Intelligence*, Madison, WI, 1998.
- [8] H. Youquan, X. Jianfang, and X. Cheng, "An improved naive Bayesian algorithm for web page text classification," in *Proceedings of 8th International Conference on Fuzzy Systems and Knowledge Discovery*, Shanghai, China, 2011, pp. 1765-1768.
- [9] D. D. Lewis and K. A. Knowles, "Threading electronic mail: a preliminary study," *Information Processing and Management*, vol. 33, no. 2, pp. 209-217, 1997. [https://doi.org/10.1016/S0306-4573\(96\)00063-5](https://doi.org/10.1016/S0306-4573(96)00063-5)
- [10] M. Sahami, S. Dumais, D. Heckerman, and E. Horvitz, "A Bayesian approach to filtering junk E-mail," in *Proceedings of the 15th National Conference on Artificial Intelligence*, Madison, WI, 1998.
- [11] W. Y. Liu, L. Wang, and T. Wang, "Online supervised learning from multi-field documents for email spam filtering," in *Proceedings of 2010 International Conference on Machine Learning and Cybernetics*, Qingdao, China, 2010. <http://dx.doi.org/10.1109/ICMLC.2010.5580676>
- [12] U. Pandey and S. Chakravarty, "A survey on text classification techniques for e-mail filtering," in *Proceed-*

- ings of 2010 Second International Conference on Machine Learning and Computing, Bangalore, India, 2010. <http://dx.doi.org/10.1109/ICMLC.2010.61>
- [13] F. Debole and F. Sebastiani, "Supervised term weighting for automated text categorization," in *Proceedings of the 2003 ACM Symposium on Applied Computing*, Melbourne, FL, 2003. <http://dx.doi.org/10.1145/952532.952688>
- [14] Z. H. Deng, S. W. Tang, D. Q. Yang, M. Z. L. Y. Li, and K. Q. Xie, "A comparative study on feature weight in text categorization," in *Advanced Web Technologies and Applications 2004: Lecture Notes in Computer Science, Vol. 3007*, J. X. Yu et al., Eds. Berlin: Springer, 2004, pp. 588-597. http://dx.doi.org/10.1007/978-3-540-24655-8_64
- [15] M. Lan, C. L. Tan, and H. B. Low, "Proposing a new term weighting scheme for text categorization," in *Proceedings of the 21st National Conference on Artificial Intelligence*, Menlo Park, CA, 2006, pp. 763-768.
- [16] J. D. Lafferty, A. McCallum, and F. C. N. Pereira, "Conditional random fields: probabilistic models for segmenting and labeling sequence data," in *Proceedings of 2001 International Conference on Machine Learning*, Williamstown, MA, 2001, pp. 282-289.
- [17] F. Sha and F. Pereira, "Shallow parsing with conditional random fields," in *Proceedings of 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, Edmonton, Canada, 2003, pp. 134-141. <http://dx.doi.org/10.3115/1073445.1073473>
- [18] K. Crammer and Y. Singer, "On the algorithmic implementation of multiclass kernel-based vector machines," *The Journal of Machine Learning Research*, vol. 2, pp. 265-292, 2001.
- [19] M. Lan, C. L. Tan, J. Su, and Y. Lu, "Supervised and traditional term weighting methods for automatic text categorization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 4, pp. 721-735, 2009. <http://dx.doi.org/10.1109/TPAMI.2008.110>
- [20] M. Kim, "Robust algorithms for combining multiple term weighting vectors for document classification," *International Journal of Fuzzy Logic and Intelligent Systems*, vol. 16, no. 2, pp. 81-86, 2016. <https://doi.org/10.5391/IJFIS.2016.16.2.81>
- [21] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the Royal Statistical Society Series B*, vol. 39, no. 1, pp. 1-38, 1977.
- [22] M. F. Porter, "An algorithm for suffix stripping," *Program*, vol. 14, no. 3, pp. 130-137, 1980. <http://dx.doi.org/10.1108/eb046814>

Minyoung Kim received his BS and MS degrees both in Computer Science and Engineering from Seoul National University, South Korea. He earned a PhD degree in Computer Science from Rutgers University

in 2008. From 2009 to 2010 he was a postdoctoral researcher at the Robotics Institute of Carnegie Mellon University. He is currently an associate professor in Department of Electronics and IT Media Engineering at Seoul National University of Science and Technology in Korea. His primary research interest is machine learning and computer vision. His research focus includes graphical models, motion estimation/tracking, discriminative models/learning, kernel methods, and dimensionality reduction.

E-mail: mikim@seoultech.ac.kr