Sentiment Classification with Supervised Sequence Embedding

Dmitriy Bespalov[1], Yanjun Qi[2], Bing Bai[2], and Ali Shokoufandeh[1]

[1] Drexel University,
Philadelphia, PA USA
[2] NEC Labs America,
Princeton, NJ USA

Abstract. In this paper, we introduce a novel approach for modeling n-grams in a latent space learned from supervised signals. The proposed procedure uses only unigram features to model short phrases (n-grams) in the latent space. The phrases are then combined to form document-level latent representation for a given text, where position of an n-gram in the document is used to compute corresponding combining weight. The resulting two-stage supervised embedding is then coupled with a classifier to form an end-to-end system that we apply to the large-scale sentiment classification task. The proposed model does not require feature selection to retain effective features during pre-processing, and its parameter space grows linearly with size of n-gram. We present comparative evaluations of this method using two large-scale datasets for sentiment classification in online reviews (Amazon and TripAdvisor). The proposed method outperforms standard baselines that rely on bag-of-words representation populated with n-gram features.

Keywords: Sentiment Classification, Large-Scale Text Mining, Supervised Feature Learning, Supervised Embedding.

1 Introduction

In this paper, we consider the problem of sentiment classification (SC) which is defined as identifying and extracting subjective information from natural language text. Due to the widespread use of electronic media and the explosion of online social-oriented content such as user reviews, sentiment classification [1], has received significant attention in recent years. This task aims to rate polarity of a given text accurately towards a label, predicting whether the expressed opinion in the text is positive, negative, or neutral.

SC task can be viewed as an instance of text categorization (TC) task. Notable varieties of TC include single-label, multi-label [2] or taxonomic hierarchy of labels [3]. Both generative approaches [4–6] and discriminative supervised methods have been applied to TC, and a few semi-supervised attempts [7] as well. Among discriminative models, support vector machines (SVM) are known for their superior performance in TC, and SC task [8, 9] in particular. Previous works on the discriminative TC commonly rely on the so-called *bag-of-words* (BoW) representation that maps variable length text into a fixed-dimensional vector, parameterized by a finite vocabulary. The

P. Flach et al. (Eds.): ECML PKDD 2012, Part I, LNCS 7523, pp. 159–174, 2012.

"bag-of-unigrams" is the most common form of BoW representation utilizing a dictionary of basic words as its vocabulary. Essentially BoW model treats a document as an unordered collection of word-features, and utilizes the frequency distribution of the words as the primary evidence for TC.

There has been increasing evidence that short phrases are more effective than single words (unigrams) for the SC task. For example, the term *good* often appears in positive online reviews, but "not very good" is less likely to appear in positive comments. When using bag-of-unigrams representation, the proximity of "not", "good" and "very" in the text is ignored. A proposed remedy is to extend the bag-of-unigrams model by incorporating n-grams (a contiguous sequence of n words) [1] as features in the vector space representation of the text [10], i.e. so called "bag-of-n-grams" (BoN). However, extending the model to incorporate n-grams (for $n >= 3$) will adversely effect the complexity of parameter space, since the dimensionality of a BoN vector grows exponentially as a function of n. For instance, extending an English word vocabulary $\mathcal{D}$ of size $|\mathcal{D}| = 10,000$ by including the bigrams ($n = 2$) and trigrams ($n = 3$) will add up to $|\mathcal{D}|^2 = 10^8$ and $|\mathcal{D}|^3 = 10^{12}$ additional free parameters, respectively. Feature selection (FS) techniques [11] are popular methods for dealing with the complexity of bag-of-n-grams model. The basic idea of FS is to retain a small subset of features, based on a certain scoring function (statistics), that are suitable for the SC task. Popular FS methods used in classifying text include Information Gain (IG), Chi-Square Test (CHI), Mutual Information (MI), Optimal Orthogonal Centroid feature selection (OCFS) [11]. More recently, Jing *et al.* [12] introduced a generalized framework for popular FS techniques. However, effectiveness of FS methods is often dataset-dependent. Thus, choosing an appropriate FS technique requires empirical validation. Furthermore, estimating optimal hyper-parameters for each FS method considered will require additional cross-validation.

Our work is motivated by the idea of utilizing short phrases as features for large-scale sentiment classification. However, in contrast to BoN model, we propose a different approach to modeling short phrases for SC task. The proposed method projects n-grams into a latent lower-dimensional space using only unigram features and avoids FS preprocessing. To be more specific, the embedding of an n-gram is a combination of the embedding of its composing words. The procedure estimates an embedding of a unigram feature for every position in the n-gram window. In this way, the parameter space of our model grows only linearly with n. The embedding of the whole document is a union of such n-gram embeddings, re-weighted based on their positions in the text. This is consistent with the hypothesis that spatial occurrence of a phrase can influence overall sentiment of an article. The parameters of our method are jointly optimized in an online learning setting through the stochastic gradient descent method [13]. The empirical evaluations demonstrate the proposed model outperforming state-of-art FS method for the SC task.

To summarize, the proposed system performs feature selection in a latent space, promoting phrases that are effective for the SC task. Section 3.3 provides interesting anecdotal evidence to support this claim. We evaluate the performance of the proposed method along with standard baselines on two sentiment classification datasets

[1] We will use "n-gram" and "phrase" interchangeably.

(Amazon[2], TripAdvisor[3]). Our empirical evidence demonstrates that the proposed framework outperforms baseline methods.

2 Supervised Sequence Embedding

In this section we will present an overview of the proposed deep neural network model that 1) represents all sliding n-gram windows in a lower-dimensional latent space; 2) obtains document representation in the latent space, defined as a weighted sum of latent n-grams, where weights are learned from positions of phrases in the document; and 3) estimates a classifier in the document-level latent space, biased towards the prescribed classification task.

Before presenting the proposed "deep" neural network model, an overview of the notations is in order. Let $\mathcal{D}$ denote the underlying word (unigram) dictionary and $\mathcal{S}$ denote the set of all finite length sequences of words from $\mathcal{D}$. We use $\Gamma = \Gamma(n) \subset \mathcal{S}$ to denote the vocabulary of n-grams in a text corpus. An input text $\mathbf{x} \in \mathcal{S}$ of length N is an ordered sequence $\mathbf{x} = (w_1, \ldots, w_N)$, with $w_i \in \mathcal{D}$. We denote an n-gram from $\mathbf{x}$, with $n < N$, starting at its j-th position as $\gamma_j = (w_j, w_{j+1}, \ldots, w_{j+n-1})$. We denote vectors or matrices with boldface font (e.g., $\mathbf{G}$ or $\mathbf{b}$), and use cursive for scalar variables and functions (e.g., M or $h(\cdot)$). We use $|.|$ to denote the cardinality of a set. Operator $\times$ denotes vector or matrix multiplication, while $\cdot$ will be used to emphasize the multiplication of scalar variables. Let $\mathcal{Y} = \{1, ..., C\}$ denote a set of class labels. $\mathcal{X} \subset \mathcal{S}$ denotes a collection of labeled documents (training set), where $\mathcal{X} = \{(\mathbf{x}_i, y_i)_{i=1,\ldots,L} | \mathbf{x}_i \in \mathcal{X} \ \& \ y_i \in \mathcal{Y}\}$ and $|\mathcal{X}| = L$.

Our model is an alternative to the classification with BoW representation. For text $\mathbf{x} = (w_1 \cdots w_N)$, the BoW model uses a unigrams dictionary to produce a $|\mathcal{D}|$-dimensional vector $\tilde{\mathbf{e}}_\mathbf{x}$ for $\mathbf{x}$:

$$\tilde{\mathbf{e}}_\mathbf{x} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{e}_{w_i}, \tag{1}$$

Here $\mathbf{e}_{w_i}$, also known as "selector", is the canonical basis vector

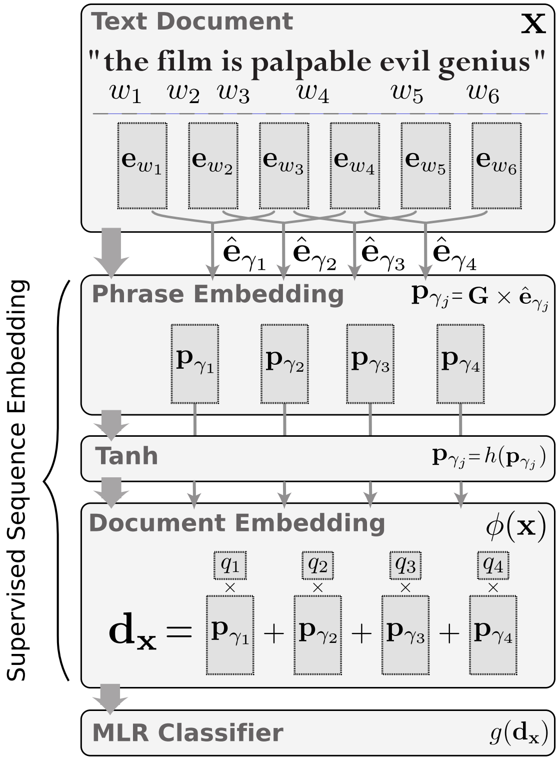
$$\mathbf{e}_{w_i} = (0, \ldots, 0, \underset{\text{at index } w_i}{1}, \ldots, 0)^\top \tag{2}$$

with a single non-zero entry at w_i-th position. It is a common practice to replace the sole non-zero entry of $\mathbf{e}_{w_i}$ with the inverse document frequency of word w_i. As a result, the vector $\tilde{\mathbf{e}}_\mathbf{x}$ in (1) takes the form of TF-IDF weighting.

The BoN extension includes n-grams as additional features [14]. By using all unique phrases of at most n words from Γ as features, we obtain a $|\Gamma|$-dimensional representation of $\mathbf{x}$. That is, BoN maps $\mathbf{x}$ to $|\Gamma|$-dimensional representation, with $|\Gamma| = O(|\mathcal{D}|^n)$. Due to its exploding number of features, BoN normally relies on feature selection methods to control the number of parameters. Differently, our method models n-grams in the latent space while recognizing only $n \cdot |\mathcal{D}|$ unique features and avoiding feature selection pre-processing.

[2] http://times.cs.uiuc.edu/~wang296/Data/TripAdvisor.tar.gz

[3] http://www.cs.jhu.edu/∼mdredze/datasets/sentiment/unprocessed.tar.gz



Text Document
X
"the film is palpable evil genius"
w_1 w_2 w_3 w_4 w_5 w_6
$\mathbf{e}_{w_1}$ $\mathbf{e}_{w_2}$ $\mathbf{e}_{w_3}$ $\mathbf{e}_{w_4}$ $\mathbf{e}_{w_5}$ $\mathbf{e}_{w_6}$
$\hat{\mathbf{e}}_{\gamma_1}$ $\hat{\mathbf{e}}_{\gamma_2}$ $\hat{\mathbf{e}}_{\gamma_3}$ $\hat{\mathbf{e}}_{\gamma_4}$
Phrase Embedding
$\mathbf{p}_{\gamma_j} = \mathbf{G} \times \hat{\mathbf{e}}_{\gamma_j}$
$\mathbf{p}_{\gamma_1}$ $\mathbf{p}_{\gamma_2}$ $\mathbf{p}_{\gamma_3}$ $\mathbf{p}_{\gamma_4}$
Tanh
$\mathbf{p}_{\gamma_j} = h(\mathbf{p}_{\gamma_j})$
Document Embedding
$\phi(\mathbf{x})$
q_1 q_2 q_3 q_4
$\mathbf{d}_{\mathbf{x}} = \mathbf{p}_{\gamma_1} + \mathbf{p}_{\gamma_2} + \mathbf{p}_{\gamma_3} + \mathbf{p}_{\gamma_4}$
MLR Classifier
$g(\mathbf{d}_{\mathbf{x}})$
Supervised Sequence Embedding

Fig. 1. Sentiment Classification using Supervised Sequence Embedding. We consider two variations of the method based on the procedure to compute combining weights q_j.

We refer to the first two stages of the proposed system as the "Supervised Sequence Embedding" (SSE). Figure 1 provides an illustration of the SC system. The first projection step that computes latent embedding of all n-grams in an article is presented in Section 2.1. The second projection that combines latent n-grams to compute article-level embedding is presented in Section 2.2. Section 2.3 presents the third step of the SC system that computes a document-level classifier in the latent space. Our framework is best described using a multi-layer projection as shown in Figure 1.

2.1 Latent n-Gram Embedding

We formally define the projection step for latent phrase embedding. The formation of n-grams is carried through a sliding window of length n. As illustrated in Figure 1, setting $n = 3$, the first n-gram is (w_1, w_2, w_3), second n-gram (w_2, w_3, w_4), etc. Given a phrase of n adjacent words, we first represent it using n word selectors. Specifically, given $\gamma_j = (w_j, w_{j+1}, \ldots, w_{j+n-1})$, define

$$\hat{\mathbf{e}}_{\gamma_j} = [\mathbf{e}_{w_j}^\top, \mathbf{e}_{w_{j+1}}^\top, \ldots, \mathbf{e}_{w_{j+n-1}}^\top]^\top, \tag{3}$$

where the notation $[\cdot]$ denotes the concatenation of single word selectors into an $n \cdot |\mathcal{D}|$-dimensional vector for each n-gram γ_j. The embedding of the γ_j is then defined as:

$$\mathbf{p}_{\gamma_j} = \mathbf{G} \times \hat{\mathbf{e}}_{\gamma_j}, \tag{4}$$

where $\mathbf{G} \in \mathbb{R}^{M \times n \cdot |\mathcal{D}|}$ is the projection matrix which maps $\hat{\mathbf{e}}_{\gamma_j}$ into a latent space with dimension M. It is important to note that M is a hyperparameter, while parameters of $\mathbf{G}$ are estimated during the learning process using backpropagation.

Since each n-gram is encoded as a sparse vector with n non-zeros in $\hat{\mathbf{e}}_{\gamma_j}$, we can treat (4) as an operation decoupling the embedding parameters for word w_i based on its position with the n-gram γ_j. Matrix $\mathbf{G}$ maintains n latent embedding vectors for every word $w_i \in \mathcal{D}$ depending on its position inside the n-gram. That means one embedding for each possible position within the n-gram for w_i.

2.2 Latent Document Embedding

We use the n-gram embedding to form a vector representation for a text document. The number of phrases in each document is variable depending on its length N. We need a function to compress the information from these n-grams into a fixed length document embedding vector. While there are many possibilities for the combining function, the mean$(\cdot)$ function has been verified by our previous work in [15] to provide a good summarization of a document in the latent space. In this work, we also propose to use a weighted sum function and to learn the weights for each $\gamma_j \in \mathbf{x}$, based on its position in the text. These weights are used to combine latent embedding of γ_j into a document-level representation. Specifically, we define latent embedding of document $\mathbf{x}$ in the latent space as:

$$\phi(\mathbf{x}) \equiv \mathbf{d}_{\mathbf{x}} = \sum_{j=1}^{N} q_j \times h(\mathbf{p}_{\gamma_j}), \tag{5}$$

where $\mathbf{d}_{\mathbf{x}} \in \mathbb{R}^M$, $\mathbf{x} = (w_1, \ldots, w_N)$, and $h(\cdot) = \tanh(\cdot)$[4]. We model the weight of every γ_j using the following mixture model. Let $\gamma_j \in \mathbf{x}$, $|\mathbf{x}| = N$ and j indicate the position of an n-gram in $\mathbf{x}$, and define the weight associated with γ_j as:

$$q_j = \frac{1}{Q} \sum_{k=1}^{K} \text{sigmoid}\left(a_k \cdot \frac{j}{N} + b_k \right), \tag{6}$$

where a_k, b_k are parameters to be learned, $Q = \sum_{j=1}^{N} q_j$, K specifies the number of mixture quantities, and sigmoid$(\cdot)$ is a non-linear transfer function. In the rest of this manuscript, we refer to the model with uniform weights $q_j = \frac{1}{N}$ (i.e., combining function is mean$(\cdot)$) as SSE, while SSE-W is used to denote the model with spatial re-weighting defined in (6). In spatial re-weighting in SSE-W model, it attempts to capture longer "trends" within each document. In this, our work is similar to the work of Lebanon *et al.* [16]. The authors propose a novel semi-parametric generative model for an unsupervised embedding of documents as smooth curves in $\mathbb{R}^{|\mathcal{D}|}$, while preserving spatial information for phrases within a document.

[4] The non-linear function $\tanh(\cdot)$ converts the unbounded range of the input into $[-1, 1]$.

2.3 Classifier

In our evaluations we use Multinomial Logistic Regression (MLR) to carry out SC. Given the document embedding $\mathbf{d_x}$, and C candidate classes, $\boldsymbol{\beta}_i$ represents the coefficient weights for the i-th candidate class. Furthermore, the predicted class label can be calculated as follows:

$$g(\mathbf{x}) = \operatorname*{arg\,max}_{i \in \{1..C\}} \frac{\exp(\boldsymbol{\beta}_i^\top \times \mathbf{d_x})}{1 + \sum_{k \in \{1..C\}} \exp(\boldsymbol{\beta}_k^\top \times \mathbf{d_x})} \tag{7}$$

This classifier can be trained by minimizing the loss function:

$$\mathcal{L}(\mathcal{X}) = - \sum_{i \in \{1..|\mathcal{X}|\}} \log \frac{\exp(\boldsymbol{\beta}_{y_i}^\top \times \mathbf{d}_{\mathbf{x}_i})}{1 + \sum_{j \in \{1..C\}} \exp(\boldsymbol{\beta}_j^\top \times \mathbf{d}_{\mathbf{x}_i})} \tag{8}$$

This latter loss is called "negative log likelihood" in literature.

The proposed supervised embedding method is implemented as a perceptron network composed of four activation layers as shown in Figure 1. We take advantage of the backpropagation process to train this layered network and use stochastic gradient descent (SGD) method for estimating the parameters [13]. For a training set $\mathcal{X}$, instead of calculating true gradient of the objective with all training samples, SGD computes the gradient with a randomly chosen training sample and updates all parameters accordingly. SGD optimization method is scalable and proven to rival the performance of batch-mode gradient descent methods when dealing with large-scale datasets [17].

2.4 Related Methods

The proposed SSE embedding has its roots in a previous model known as "Lookup Table Convolution" (LTC) [15, 18]. LTC constructs a low-dimensional latent embedding for all $\gamma_j \in \mathbf{x}$ by first projecting each word into a latent space, followed by a second projection step to obtain the latent embedding of each n-gram. Specifically, each word $w_j \in \mathcal{D}$ is embedded into the m-dimensional feature space using a word lookup table:

$$LT_{\mathbf{E}}(w_j) = \mathbf{E} \times \mathbf{e}_{w_j} = \mathbf{E}_{w_j}, \tag{9}$$

where the j-th column of the matrix $\mathbf{E} \in \mathbb{R}^{m \times |\mathcal{D}|}$ denotes the embedding vector of the word w_j. Given an n-gram γ_j, the word lookup table applies the same operation to each word inside the n-gram sliding window, producing the vector $\mathbf{z}_{\gamma_j} = [\mathbf{E}_{w_j}^\top, \mathbf{E}_{w_{j+1}}^\top, \ldots, \mathbf{E}_{w_{j+n-1}}^\top]^\top$, with $[\cdot]$ denoting the concatenation of single word embedding into an $n \cdot m$-dimensional vector. The latent embedding for γ_j is then defined as

$$\tilde{\mathbf{p}}_{\gamma_j} = \mathbf{F} \times \mathbf{z}_{\gamma_j} = \mathbf{F} \times [\mathbf{E}_{w_j}^\top, \mathbf{E}_{w_{j+1}}^\top, \ldots, \mathbf{E}_{w_{j+n-1}}^\top]^\top, \tag{10}$$

where projection matrix $\mathbf{F} \in \mathbb{R}^{M \times n \cdot m}$ maps $\mathbf{z}_{\gamma_j}$ into the M-dimensional latent space. This two-step embedding procedure encodes each n-gram in a latent space without the explicit construction of all n-grams. Collobert and Weston [18] empirically validated LTC on six Natural Language Processing (NLP) tasks. Our previous work [15] adopted LTC for sentiment classification.

We emphasize the difference between this work and [15]. First, instead of modeling a lookup table layer followed by a convolutional layer as done in LTC, SSE models the parameters of the latent n-gram embedding directly using matrix G in (4). Second, the SSE-W model uses spatial re-weighting of n-grams, while uniform weights (i.e., mean$(\cdot)$ combining function) are used in LTC (and SSE). This improves the performance in many cases as our experimental evaluations in Section 3 suggest. In addition, the experimental results suggest the SSE model achieves higher SC accuracy, compared to the LTC method described by (9) and (10). Furthermore, training an LTC model using backpropagation requires many vector multiplications to calculate gradients $\partial\mathcal{L}/\partial\mathbf{E}$ and $\partial\mathcal{L}/\partial\mathbf{F}$ due to the multiplicative coupling of $\mathbf{E}$ and $\mathbf{F}$. In contrast, in training SSE models, these computations are largely avoided.

In general, performing dimensionality reduction in the original high-dimensional feature space is a common practice for various classification methods. Popular unsupervised latent embedding methods on text documents includes Latent Semantic Indexing (LSI) [19], or its probabilistic extensions, probabilistic LSI (pLSI) [20], and Latent Dirichlet Allocation (LDA) [21]. However, biasing parameters of the embedding towards specific classification task has not received much attention until recently, such as LTC from [18] and the work of learning to rank with joint word-image embedding in [22].

Our work is also related to the "Deep learning" architecture which has received increasing attention in recent years. Deep architectures have been used to learn complicated functions in natural language processing and computational vision [23]. Each layer in the architecture encodes features at different levels of abstraction, defined as a composition of features computed at the previous layer. Glorot *et al.* [24] utilize a deep learning model to extract the representation of each text review in an unsupervised fashion using stacked Denoising Auto-encoders. With the learned high-level feature representation the authors claim to achieve state-of-the-art performance for domain adaption tasks on sentiment classification data. Socher *et al.* [25] use recursive neural networks to perform simultaneous parsing and classification of both text and image data. In addition, multi-layered neural networks are successfully used for learning language models that estimate conditional probability distribution for word sequences [26, 27].

Another relevant domain to our work is the "string kernel" framework. String kernels and their extensions have been very popular discriminative choices for the protein classification problem, where sequences of amino acids are represented as strings [28, 29]. These kernels map a variable length string into a low-dimensional dense feature space using BoW strategy. Similar approaches have also been applied to text categorization before: see e.g., [30]. A critical component in the string kernel research is the implementation of inexact matching between short sequence segments. These approaches give rise to a family of mismatch kernels [28]. Similarly, the SSE method allows for inexact phrase matching that takes place in the latent space.

3 Experiments

We evaluate the performance of the proposed SSE method on SC task with binary and multi-class setting. For binary classification setting, we only consider positive (1 and 2 stars) or negative (4 and 5 stars) sentiment in the reviews. For multi-class setting we use four available labels (1,2,4 and 5 stars) to evaluate text classification on Amazon and TripAdvisor datasets. In addition, since TripAdvisor contains neutral reviews, we also consider a SC task with five category labels for this dataset.

Amazon dataset contains customer reviews of 25 various categories of goods including apparel, automotive, baby, DVDs, electronics, magazines, and tools and hardware. TripAdvisor dataset contains customer reviews for various hotels across the globe. While TripAdvisor corpus provides rating scores for various aspects (e.g., rooms, location, cleanliness), we only consider overall ratings for this dataset. These are considered some of the largest sentiment classification datasets currently available. For Amazon we use 257,900 samples for training and 110,562 samples for testing, while 55,306 and 10,078 samples from TripAdvisor were used for training and testing, respectively. The development sets contain 10,000 and 5,000 samples for Amazon and TripAdvisor, respectively. In this work, we report classification results obtained using train-development-test splits for Amazon and TripAdvisor datasets. These dataset splits are available for download from our website[5]. It is important to note that the empirical evidence reported in this work are not directly comparable to the results we reported in [15], as we use different splits for Amazon and TripAdvisor datasets. However, to be fair, we make available online the SA results for the proposed SSE method, benchmarked on the split used in our previous publication [15].

Amazon and TripAdvisor datasets contain user-generated reviews where an overall sentiment for each review is quantified with an integer 1 through 5 (a.k.a the 5-star Likert scale). A sentiment score of 1 star corresponds to the lowest (negative) sentiment, while the score of 5 stars corresponds to the highest (positive) sentiment. TripAdvisor dataset contains neutral reviews (rated with 3-stars), while neutral reviews were omitted during the construction of Amazon dataset by their authors. For both datasets, a balanced version of the data splits (i.e., training / testing / development) is created that contain equal number of positive (4 and 5 stars) and negative (1 and 2 stars) reviews.

Table 1 provides the number of unique phrases for $n \in \{1, 2, 3, 5\}$ found in the training sets. Clearly, when $n \geq 2$, a feature selection technique is necessary, not only to improve the classification accuracy but also to keep the optimization tractable. For both datasets, we follow the method used in [31] to limit the vocabulary size by retaining n-grams with the highest mutual information (MI) shared by the binary labels (positive or negative). For the Amazon dataset, we use training split to select 25,000 grams per category, then concatenate the phrases to form the vocabulary used in the experiments. For TripAdvisor dataset, we also use an MI-based procedure to limit the vocabulary size to 500,000 n-grams computed for the entire training corpus.

For one of the baseline methods we use a linear SVM classifier, which is trained on BoN document representation. We obtain BoN representation with TF-IDF and $n \in \{1, 2, 3, 5\}$. We restrict our evaluation to the linear kernel because of the corpus size and the number of features used in describing each document. Prior research showed linear SVM achieving state-of-art performance on SC tasks (see e.g., [8] or [9]). In addition, we use a linear perceptron classifier trained on BoN as another baseline. We believe the latter choice is relevant, since the main objective of this work is to test the merit of the proposed SSE against the BoW model populated with n-grams.

[5] http://mst.cs.drexel.edu/datasets/ECML2012

Table 1. Unique phrase counts $|\mathbf{\Gamma}|$ for each dataset. Numbers are in thousands.

n-gram size	Amazon	TripAdvisor	RCV1 23k	RCV1 380k
$n = 1$	448	158	124	262
$n = 2$	6,446	1,175	2,400	6,364
$n = 3$	23,400	5,172	9,535	30,377
$n = 5$	78,864	21,741	35,118	262,586

We use **SVM** and **Prc** to denote the SVM and linear perceptron classifiers, respectively. The BoN representation will be denoted with **BoW-ng**, while $|\mathbf{\Gamma}|$ denotes the number of unique phrases (in thousands) for the training sets. We use **LTC** for referring to the Lookup Temporal Convolution method presented in [15]. We denote the proposed SSE method with $mean(\cdot)$ used for combining function as **SSE**. SSE method with spatial re-weighting of n-grams defined in (6) is identified as **SSE-W**. The rest of this section is organized as follows. We provide important implementation details in Section 3.1. Sentiment classification results are discussed in Section 3.2. Section 3.3 provides anecdotal evidence that selecting phrases with highest prediction responses from a trained SSE model can be used for "sentiment summarization". We also provide an illustration of the estimated spatial weights for trained SSE-W model. In addition we demonstrate that SSE-W can be augmented with an alternative combining function q_j that captures strength of sentiment in text, but not polarity. Finally, in Section 3.4 we present topic categorization results on Reuters dataset [32] to demonstrate that SSE model is applicable to text categorization tasks other than SC.

3.1 Implementation Details

In our implementation we used the following formulation of TF-IDF. For every n-gram $\gamma_j \in \mathbf{x}$ where document $\mathbf{x} \in \mathcal{X}$, the weight for γ_j was calculated using the formula: $\mathrm{tfidf}(\gamma_j, \mathbf{x}, \mathcal{X}) = \frac{1}{|\mathbf{x}|} \cdot \mathrm{tf}(\gamma_j, \mathbf{x}) \cdot \mathrm{idf}(\gamma_j, \mathcal{X})$, where $\mathrm{idf}(\gamma_j, \mathcal{X}) = \log \frac{|\mathcal{X}|}{|\{\mathbf{x}_i \in \mathcal{X} : \gamma_j \in \mathbf{x}_i\}|}$, and $\mathrm{tf}(\gamma_j, \mathbf{x})$ returns the number of times term γ_j appears in $\mathbf{x}$.

We used the LIBLINEAR[6] SVM toolkit. For each SC task, the penalty parameter C was set using grid search with $C = \{2^{-8}, 2^{-7}, \ldots, 2^{10}, 2^{11}\}$, performed on the development set. Then the reported classification error was computed on the testing set with the optimal penalty parameter found. Perceptron-based methods (LTC, SSE, SSE-W and Prc BoW) were implemented using the Torch5[7] machine learning library. A development set was used to select the best model during the training of all perceptron classifiers. During the training procedure the model was evaluated at regular intervals on the entire development set, and the best performing model was retained. After the training was completed, this model was used to compute the classification error rate for the testing set, which is the number reported in all of our experiments below.

The perceptron classifiers were trained with a fixed learning rate 0.05. The dimensionality of the latent space for all perceptron-based methods was set to $M = 50$ in all

[6] `http://www.csie.ntu.edu.tw/~cjlin/liblinear/`

[7] `http://torch5.sourceforge.net/`

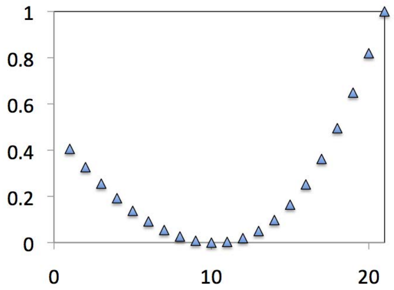
Table 2. Macro-average classification error rate for SVM with BoN, where $n \in \{1, 2, 3, 5\}$. Macro-average error rate is calculated as mean of per-label classification error rates. $2 \cdot \star$ denote binary classification setting, while $4 \cdot \star$ and $5 \cdot \star$ identify multi-class setting with four and five categories, respectively. The numbers marked with † (or ‡) are statistically significantly better than **SVM BoW-1g** with $p < 0.0001$ (or $p < 0.01$).

Method	Amazon		TripAdvisor		
	$2 \cdot \star$	$4 \cdot \star$	$2 \cdot \star$	$4 \cdot \star$	$5 \cdot \star$
SVM BoW-1g	10.68	35.78	8.97	35.41	46.41
SVM BoW-2g	6.60†	28.26†	7.60‡	33.68‡	**44.68**‡
SVM BoW-3g	**6.39**†	**27.98**†	**7.46**‡	33.50‡	45.12
SVM BoW-5g	6.48†	28.02†	7.53‡	**33.45**‡	46.41

the experiments. We set these parameters according to our prior experience in designing perceptron-based classification systems, and did not subject them to the empirical selection in this work. We selected the length of the latent phrase for SSE and SSE-W methods (i.e., size of n-gram) after evaluating SVM classification performance with BoN and $n \in \{1, 2, 3, 5\}$. These results are presented in Table 2. We selected $n = 5$ when modeling latent phrases in SSE and SSE-W methods. Finally, we fixed $K = 3$ in (6), which was motivated by our assumption that phrases appearing in the beginning or at the end of each text are the most effective at predicting text labels. The results presented in Section 3.3 support this hypothesis.

3.2 Classification Results

Table 3 presents SC results using macro-average error rate, defined as mean of per-label classification error rates. For completeness of presentation, we also provide micro-average classification error rates in Table 4, computed over all test samples regardless



(a) SSE-W with spatial weights (6)

5-gram	Weight
is an extremely good book	3.58
is just a good book	3.19
book is a good buy	2.94
overall a very good book	2.84
book is a good choice	2.81
book is a very good	2.76
book is still very good	1.70
a good book just because	1.15
unless good books are just	1.05

(b) SSE-W with weights $q_j = \hat{\mathbf{G}} \times \hat{\mathbf{e}}_{\gamma_j}$

Fig. 2. **(a)** Illustration of spatial weights in SSE-W model trained on the Amazon dataset. The values of the spatial weights were computed for a "synthetic" text with 25 words. The weights are scaled into range $[0, 1]$ for illustration purposes. **(b)** Select 5-grams and their combining weights. The weights are computed using the model $q_j = \hat{\mathbf{G}} \times \hat{\mathbf{e}}_{\gamma_j}$ trained on Amazon dataset with binary classification setting.

of their labels. It is worth noting that splits for Amazon and TripAdvisor are balanced in terms of binary sentiment polarity, thus binary classification error rates in Table 4 match the macro-average results from Table 3. In the five experiments conducted SSE-W method outperforms the SVM baseline. However, only in the multi-class setting SSE-W method results in statistically significant improvement over SSE model, with $p < 0.0001$ for Amazon and $p < 0.01$ for TripAdvisor datasets. In case of binary classification on Amazon, improvement of SSE-W over SSE is only statistically significant with $p < 0.4$. These results suggest that spatial re-weighting of phrases only becomes relevant when predicting sentiment on the Likert scale with multiple labels. Also, when predicting binary sentiment, the presence of certain phrases, regardless of their positions within the text, is sufficient for the task.

Table 3. Macro-average classification error rate. Macro-average error rate is calculated as mean of per-label classification error rates. $2 \cdot \star$ denote binary classification setting, while $4 \cdot \star$ and $5 \cdot \star$ identify multi-class setting with four and five categories, respectively. The numbers marked with † (or ‡) are statistically significantly better than **SVM BoW-3g** with $p < 0.0001$ (or $p < 0.01$).

Method	**Amazon**		**TripAdvisor**		
	$2 \cdot \star$	$4 \cdot \star$	$2 \cdot \star$	$4 \cdot \star$	$5 \cdot \star$
SVM BoW-3g	6.39	27.98	7.46	33.50	45.12
Prc BoW-3g	6.55	26.45†	7.54	34.73	43.58
SSE	5.69†	25.30†	**6.90**	34.22	42.88‡
SSE-W	**5.63**†	**24.61**†	7.01	**32.25**	**40.54**†
LTC	7.05	-	8.49	-	-

3.3 Illustrative Examples

We now present several illustrative examples obtained using SSE and SSE-W models trained on the sentiment datasets with multi-class $4 \cdot \star$ setting. Table 5 shows three non-overlapping 5-grams with highest weights obtained from selected TripAdvisor reviews. The weight for each phrase γ_j is set using $\max_{i \in \{1 \ldots C\}} \boldsymbol{\beta}_i^\top \times \phi(\gamma_j)$, where $\phi(\gamma_j)$ denotes latent embedding of γ_j. The trained SSE model for TripAdvisor is used to compute embedding of each phrase γ_j separately.

Table 4. Micro-average classification error rate. Micro-average error rate, computed over all test samples, regardless of their labels is reported. The numbers marked with † (or ‡) are statistically significantly better than **SVM BoW-3g** with $p < 0.0001$ (or $p < 0.01$).

Method	**Amazon**		**TripAdvisor**		
	$2 \cdot \star$	$4 \cdot \star$	$2 \cdot \star$	$4 \cdot \star$	$5 \cdot \star$
SVM BoW-3g	6.39	23.45	7.46	32.00	43.07
Prc BoW-3g	6.55	23.00‡	7.54	33.94	43.05
SSE	5.69†	22.40†	**6.90**	33.90	42.21
SSE-W	**5.63**†	**22.05**†	7.01	**31.41**	**40.76**‡

We also present sample illustration of spatial weights obtained from the SSE-W model trained on the Amazon dataset with binary classification setting. The values of the spatial weights were computed for a "synthetic" text with 25 words. The weights are scaled into range $[0, 1]$ for illustration purposes. Please refer to Figure 2a for the illustration. We note the obtained weights have a straightforward interpretation – phrases or sentences that appear in the beginning or at the end of each review are more likely to express strong sentiment that defines the polarity of the review.

In addition to spatial information, weights q_j in (6) can be computed with other models. For example, latent projection layer from (4) can be used to compute weights q_j directly from the sequence features. In this case, another projection $\hat{\mathbf{G}}$ is estimated, where $\hat{\mathbf{G}} \in \mathbb{R}^{1 \times n \cdot |\mathcal{D}|}$ and $q_j = \hat{\mathbf{G}} \times \hat{\mathbf{e}}_{\gamma_j}$. To illustrate our point, we use the SSE model trained on the binary Amazon dataset to initialize the modified SSE-W model. The modified SSE-W model is then trained on the Amazon dataset, while keeping the projection parameters $\mathbf{G}$ unchanged. We then use weights $q_j = \hat{\mathbf{G}} \times \hat{\mathbf{e}}_{\gamma_j}$ to sort all 5-grams from the Amazon's testing set. Figure 2b lists several top-scoring 5-grams that we have selected from the sorted list of the 5-grams that contained words "good" and "book". We note that this model captures only sentiment strength and the obtained list contains phrases that carry both positive and negative sentiment. The estimated weights q_j, presented in Figure 2b, support this argument. For example, one can argue that "is an extremely good book" carries stronger (positive) sentiment than "book is a good choice", which in turn has stronger sentiment than "a good book just because".

Table 5. Summarization for select TripAdvisor reviews, obtained as the top three non-overlapping 5-grams. The trained TripAdvisor SSE model with multi-class $4 \cdot \star$ setting is used to calculate phrase weights.

Review Text	**Rating**	**5-gram**	**Weight**
disappointing choice this **is one of the worst** large hotels i have ever visited . the suite i had was filthy , and the food from room service **was barely edible (the** caesar salad was dangerously inedible) . there is no wifi . two lamps do not work . feels like a decrepit ocean liner . despite the view and the location , i would **avoid this place at all** cost .	$\star$	is one of the worst avoid this place at all was barely edible (the	34.1 31.3 28.6
noisy air conditioning on NUMBERnd floor ! ! we stayed one night in the sand villa , in a room on the NUMBERnd floor overlooking the pool . the room was comfortable . there was a loud rumbling noise , seemingly from something like a big central air conditioner , that continued all night . it was about as loud as a plane during flight - - certainly not , but not pleasant either . the **staff was pleasant and helpful** , but because of the noise i **would not stay there again** .	$\star\star$	staff was pleasant and helpful noisy air conditioning on would not stay there again	30.6 22.1 20.6
very nice experience the frenchmen **is a very nice place** to stay . the rooms were decorated nicely and the courtyard with the **jacuzzi and pool were beautiful** . above all , the staff was probably the friendliest i ’ ve ever encountered . very outgoing and pleasant . **the only bad thing i** could say about it is that the rooms were just a little small , but for a single person or a close couple , it was fine .	$\star\star\star\star$	the only bad thing i jacuzzi and pool were beautiful is a very nice place	17.3 16.7 16.3
stylish and great staff i stayed at the hotel globus in may NUMBER as a single female traveller . the room was small but very stylish and spotless . **the staff were all fantastic** and very friendly . **good breakfast and excellent location** for the railway station and easy reach of all florence ’ s attractions . i ’ m going back to florence in december and will be staying there again .	$\star\star\star\star\star$	the staff were all fantastic stylish and great staff i good breakfast and excellent location	26.8 22.1 20.6

3.4 Topic Categorization

In addition to SC task we consider topic categorization using Reuters dataset (RCV1)[8]. The original Reuters Corpus (RCV1) contains train-test split of 23,149 and 781,265 documents, respectively. The documents in the RCV1 corpus are categorized with 103 topics. The main focus of our research is the development of text classification methods that can efficiently handle large-scale data. Thus, we also create a new split for RCV1 with 380,000 training samples. Following the procedure of Lewis *et al.* [32] we select documents with IDs between 2,286 and 383,792 for training in the new split. We denote the split with the original (smaller) training set with **RCV1 23k**, while the split with larger training set is denoted by **RCV1 380k**. To obtain a development set for both splits of RCV1, we randomly sample 10,000 documents from the corresponding testing sets. We restrict our evaluations to the four topics with the largest number of positive examples in the entire Reuters Corpus: CCAT (ALL Corporate-Industrial) GCAT (All Government and Social), MCAT (ALL Securities and Commodities Trading and Markets) and C15 (Corporate and Industrial Performance). For RCV1 dataset we limit the vocabulary size to the 500,000 most frequent n-grams selected using training set only.

Table 6. Macro-average classification error rate for RCV1 dataset. Macro-average error rate is calculated as mean of per-label classification error rates. The numbers marked with † (or ‡) are statistically significantly better than **SVM BoW-2g** with $p < 0.0001$ (or $p < 0.01$).

Method	RCV1 23k				RCV1 380k			
	CCAT	GCAT	MCAT	C15	CCAT	GCAT	MCAT	C15
SVM BoW-2g	5.82	5.42	5.60	7.62	**4.07**	4.47	3.95	4.93
SSE	5.74	4.79†	**4.41**†	6.21†	4.29	**3.81**†	**3.42**†	5.76
SSE-W	**5.71**‡	**4.70**†	4.45†	**5.50**†	4.15	**3.81**†	3.47†	**4.28**†

Table 6 presents text categorization results for RCV1 dataset. SSE-W method outperforms the SVM baseline in all but one experiment. In addition, SSE-W does not improve classification over the SSE model for the MCAT topic, and the classification improvements are rather small for the CCAT and GCAT topics. On the other hand, the improvement of SSE-W over the SSE method is statistically significant with $p < 0.0001$ in the case of C15 topic. We speculate these results can be attributed to the nature of the topics considered. Indeed, MCAT, CCAT and GCAT are high-level topics in RCV1, with each assigned to news articles that describe broad range of concepts. Furthermore, C15 identifies articles only related to corporate and industrial performance, thus allowing SSE-W model to identify the spatial distribution of the effective phrases, that C15 articles exhibit.

4 Conclusions and Future Work

This work presents a supervised method (SSE) for the latent embedding of n-grams. The experimental results show improved text classification performance over the

[8] We use raw text features, instead of stemmed words as used in the original RCV1 publication.

baseline classifiers trained on BoN models. In addition, the proposed extension to the model (SSE-W) incorporates the relative position of the phrases when forming latent representation of a document. SSE-W model improves sentiment classification accuracy over SSE model that uses uniform weights ($q_j = \frac{1}{N}$).

We limit our empirical evaluation in this work to document-level text classification, focusing on sentiment analysis problem. We believe the SSE model can also be applied to sequence classification in general, where a sequence is an ordered list of events that can be described using a finite set of features. In the future work, we plan to investigate the merit of the proposed system for various sequence classification tasks. For example, the task of classifying protein sequences [33] or query log sequences to identify human users [34]. In addition, we plan to consider applying the framework to other modalities where BoW representation is used. For instance, object recognition in images is another promising direction of our future work.

References

1. Pang, B., Lee, L.: Opinion mining and sentiment analysis. Foundations and Trends in Information Retrieval 2(1-2), 1–135 (2008)
2. Zhu, S., Ji, X., Xu, W., Gong, Y.: Multi-labelled classification using maximum entropy method. In: Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2005, pp. 274–281. ACM, New York (2005)
3. Sun, A., Lim, E.P.: Hierarchical text classification and evaluation. In: Proceedings of the 2001 IEEE International Conference on Data Mining, ICDM 2001, pp. 521–528. IEEE Computer Society, Washington, DC (2001)
4. McCallum, A., Nigam, K.: A comparison of event models for naive bayes text classification. In: AAAI 1998 Workshop on Learning for Text Categorization, vol. 752, pp. 41–48 (1998)
5. Nigam, K.: Using maximum entropy for text classification. In: IJCAI 1999 Workshop on Machine Learning for Information Filtering, pp. 61–67 (1999)
6. Yi, K., Beheshti, J.: A hidden markov model-based text classification of medical documents. J. Inf. Sci. 35, 67–81 (2009)
7. Nigam, K., McCallum, A.K., Thrun, S., Mitchell, T.: Text classification from labeled and unlabeled documents using em. Mach. Learn. 39, 103–134 (2000)
8. Mirowski, P., Ranzato, M., LeCun, Y.: Dynamic auto-encoders for semantic indexing. In: Proceedings of the NIPS 2010 Workshop on Deep Learning (2010)
9. Paltoglou, G., Thelwall, M.: A study of information retrieval weighting schemes for sentiment analysis. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, ACL 2010, pp. 1386–1395. Association for Computational Linguistics, USA (2010)
10. Cavnar, W., Trenkle, J.: N-gram-based text categorization. Ann. Arbor. MI 48113(2), 161–175 (1994)
11. Yan, J., Liu, N., Zhang, B., Yan, S., Chen, Z., Cheng, Q., Fan, W., Ma, W.Y.: Ocfs: optimal orthogonal centroid feature selection for text categorization. In: Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2005, pp. 122–129. ACM, New York (2005)

12. Jing, H., Wang, B., Yang, Y., Xu, Y.: A General Framework of Feature Selection for Text Categorization. In: Perner, P. (ed.) MLDM 2009. LNCS, vol. 5632, pp. 647–662. Springer, Heidelberg (2009)
13. Bottou, L.: Stochastic Learning. In: Bousquet, O., von Luxburg, U., Rätsch, G. (eds.) Machine Learning 2003. LNCS (LNAI), vol. 3176, pp. 146–168. Springer, Heidelberg (2004)
14. Robertson, S., Zaragoza, H.: The probabilistic relevance framework: Bm25 and beyond. Found. Trends Inf. Retr. 3, 333–389 (2009)
15. Bespalov, D., Bai, B., Qi, Y., Shokoufandeh, A.: Sentiment classification based on supervised latent n-gram analysis. In: ACM Conference on Information and Knowledge Management, CIKM (2011)
16. Lebanon, G., Mao, Y., Dillon, J.: The locally weighted bag of words framework for document representation. J. Mach. Learn. Res. 8, 2405–2441 (2007)
17. Bottou, L.E., Cun, Y.L.: Large scale online learning. In: NIPS 2003. MIT Press (2004)
18. Collobert, R., Weston, J.: A unified architecture for natural language processing: Deep neural networks with multitask learning. In: International Conference on Machine Learning, ICML (2008)
19. Deerwester, S., Dumais, S.T., Furnas, G.W., Landauer, T.K., Harshman, R.: Indexing by latent semantic analysis. Journal of The American Society for Information Science 41(6), 391–407 (1990)
20. Hofmann, T.: Probabilistic latent semantic indexing. In: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 50–57. ACM Press, New York (1999)
21. Blei, D., Ng, A., Jordan, M.: Latent dirichlet allocation. The Journal of Machine Learning Research 3, 993–1022 (2003)
22. Weston, J., Bengio, S., Usunier, N.: Large scale image annotation: learning to rank with joint word-image embeddings. Machine learning 81(1), 21–35 (2010)
23. Bengio, Y.: Learning Deep Architectures for AI. Now Publishers Inc., Hanover (2009)
24. Glorot, X., Bordes, A., Bengio, Y.: Domain adaptation for large-scale sentiment classification: A deep learning approach. In: Proceedings of the 28th International Conference on Machine Learning (ICML 2011). Omnipress, Bellevue (June 2011)
25. Socher, R., Lin, C.C.Y., Ng, A., Manning, C.: Parsing natural scenes and natural language with recursive neural networks. In: Getoor, L., Scheffer, T. (eds.) Proceedings of the 28th International Conference on Machine Learning (ICML 2011), pp. 129–136. ACM, New York (June 2011)
26. Bengio, Y., Ducharme, R., Vincent, P., Operationnelle, D.D.E.R.: A neural probabilistic language model. Journal of Machine Learning Research 3, 1137–1155 (2000)
27. Morin, F.: Hierarchical probabilistic neural network language model. In: AISTATS 2005, pp. 246–252 (2005)
28. Leslie, C.S., Eskin, E., Weston, J., Noble, W.S.: Mismatch string kernels for SVM protein classification. In: NIPS, pp. 1417–1424 (2002)
29. Weston, J., Leslie, C., Ie, E., Zhou, D., Elisseeff, A., Noble, W.S.: Semi-supervised protein classification using cluster kernels. Bioinformatics 21(15), 3241–3247 (2005)
30. Lodhi, H., Saunders, C., Shawe-Taylor, J., Cristianini, N., Watkins, C.: Text classification using string kernels. J. Mach. Learn. Res. 2, 419–444 (2002)
31. Blitzer, J., Dredze, M., Pereira, F.: Biographies, bollywood, boomboxes and blenders: Domain adaptation for sentiment classification. In: ACL, pp. 187–205 (2007)

32. Lewis, D.D., Yang, Y., Rose, T.G., Li, F., Dietterich, G., Li, F.: Rcv1: A new benchmark collection for text categorization research. Journal of Machine Learning Research 5, 361–397 (2004)
33. Deshpande, M., Karypis, G.: Evaluation of Techniques for Classifying Biological Sequences. In: Chen, M.-S., Yu, P.S., Liu, B. (eds.) PAKDD 2002. LNCS (LNAI), vol. 2336, pp. 417–431. Springer, Heidelberg (2002)
34. Duskin, O., Feitelson, D.G.: Distinguishing humans from robots in web search logs: preliminary results using query rates and intervals. In: Proceedings of the 2009 Workshop on Web Search Click Data. WSCD 2009, pp. 15–19. ACM, New York (2009)