

Document-Level Sentiment Analysis through Incorporating Prior Domain Knowledge into Logistic Regression

Nesar Ahmad Wasi

Department of Computer Science
South Asian University, New Delhi, India
nesarahmadwasi17@gmail.com

Muhammad Abulaish, SMIEEE

Department of Computer Science
South Asian University, New Delhi, India
abulaish@ieee.org

Abstract—In this paper, we present a prior domain knowledge-enhanced classification approach for document-level sentiment analysis. The proposed approach is a hybrid category of sentiment classification approaches which uses two types of prior domain knowledge – *general sentiment knowledge* extracted from a lexicon, and knowledge extracted from unlabeled domain data that we call *domain-specific sentiment knowledge*. We combine prior domain knowledge with logistic regression to enhance sentiment classification, and use gradient descent approach to optimize the modified logistic regression model. The novelty of our proposed approach lies in incorporating prior domain knowledge directly into the logistic regression model. The proposed approach is empirically evaluated through extensive experiments over a multi-domain sentiment dataset. It is also compared with three baseline methods and performs significantly better.

Index Terms—Sentiment Analysis, Machine Learning, Logistic Regression, Domain Knowledge, Lexicon.

I. INTRODUCTION

In the era of online social media, mining sentiments from user-generated gigantic data is an advantageous task in many aspects [1]. For instance, new customers can carry out an informed choice based on the opinions shared by the other customers. Similarly, business enterprises may become aware of their customers' responses towards the existing products [2]. However, such kind of user-generated sentiment-bearing data is massive in size and usually unlabeled, and labeling data comes with a cost and certainly requires time and human resources. Document-level sentiment classification is the task of associating a document to the sentiment polarity that it carries. To perform this task, supervised machine learning algorithms are broadly applied [3]. In a supervised learning setting, a classifier is trained on a known labeled training dataset and tested on samples that the classifier has not seen in the training phase. Although, it is easy to work with labeled data, inferencing and extracting information from unlabeled data is a challenging task. In this study, we utilize unlabeled data in form of prior knowledge to help improving the performance of the learning models. The existing literature related to sentiment classification mainly consists of three types of approaches – lexicon-based approaches, corpus-based approaches, and a hybrid approaches. Lexicon-based

approaches aggregate the scores of the sentiment words that appear in sentiment lexicons for determining the sentiment polarity of the documents. On the other hand, corpus-based approaches classify a document using a classifier trained on a labeled document corpus. Finally, the third category of sentiment classification approaches use lexicons with corpus-based approaches in synergy. Our proposed approach falls into the third category.

In this paper, we utilize prior sentiment knowledge which consists of two types of knowledge – *general sentiment knowledge* and knowledge extracted from unlabeled document corpus that we call *domain-specific sentiment knowledge*. The *general sentiment knowledge* are extracted from general-purpose sentiment lexicons, whereas *domain-specific sentiment knowledge* are extracted using the work of [4]. According to [4], if two terms are joined by a coordinating conjunction, then they are considered to have the same sentiment-orientation. On the other hand, if two terms are joined via adversarial conjunctions; or if two terms are joined by a coordinating conjunction, but a negative term like 'not' appears before either of them, then the terms are more likely to have an opposite sentiment-orientation. We incorporate the aforementioned two types of prior sentiment knowledge in logistic regression, and use gradient descent approach to optimize the objective function of our proposed classification model. Our proposed approach is primarily distinct from the existing document-level sentiment classification methods. We combine the knowledge learned from both lexicon as well as unlabeled data to improve the classification task. Further, we incorporate prior knowledge with labeled samples to train classification models. The novelty of the proposed approach is that it can incorporate prior knowledge learned from lexicon and unlabeled data directly into the regression model.

II. RELATED WORKS

There are two ways in the literature which focuses on the course of utilizing lexicon into machine learning approaches. The first approach is to apply two different models in order to get two parametric models and then combine them into a single system. Andreevskaia and Bergler [5] followed the aforemen-

tioned approach. The other approach is to leverage the lexicon-based knowledge directly in the learning of the models. Wilson et al. [6] presented a two-step method in which, first, the expressions that carry polarity are identified and then contextual polarity of the sentiment-bearing expressions is determined using the BoosTexter AdaBoost classifier. Prem et al. [7] presented an approach that integrates the background lexical learning in the form of a feature-class association with a supervised machine learning approach. In particular, they presented two separate models – one used lexical knowledge based on lexicon-loaded terms, and another model was trained over labeled data. Thereafter, the learned distribution of both models is adaptively pooled via multinomial naive Bayes classifier to apprehend information from both models.

Dang et al. [8] proposed a lexicon-enhanced approach in which the features generated using a Parts-Of-Speech (POS) tagger and SentiWordNet lexicon is combined with content-free and content-dependent features for sentiment classification. Tao et al. [9] proposed an approach which is based on the non-negative tri-factorization of the term-document matrix, and constrained on domain-dependent, domain-independent, and some labeled data. The lexicon is used to extract domain-independent features and use them as prior knowledge; whereas, domain-dependent features are extracted from unlabeled data. Fang and Chen [10] presented a method in which they incorporated sentiment lexicon with a learning method to enhance sentiment classification. In their approach, they first employed an aspect classifier to build domain-specific lexicon, and then they applied a sentiment polarity classifier to predict sentiment associated with the sentiment terms. Finally, the results produced by both classifiers are aggregated to produce the final prediction. Yulan He [11] proposed two approaches to incorporate prior knowledge. In first method, Dirichlet prior of the Latent Dirichlet Allocation (LDA) [12] is tweaked, and class labels are treated as topics. In second method, the objective function of the LDA model is modified. Han et al. [13] incorporated the concept of mutual information to produce a domain-specific lexicon.

III. PRIOR KNOWLEDGE EXTRACTION

We integrate prior domain knowledge with classification model to improve the accuracy of the classification task. The prior knowledge is extracted from sentiment lexicon and unlabeled data of the same domain. The general-purpose sentiment lexicons carry a substantial list of general terms with their polarity orientation and score [14]. The knowledge learned from lexicons is called prior sentiment knowledge, and that learned from unlabeled samples is called prior sentiment polarity relation between the words. Both prior sentiment knowledge and prior sentiment polarity relation among words constitute our prior domain knowledge. In order to extract prior sentiment knowledge, we have followed the works of [15] and [16]. In this study, prior sentiment knowledge is denoted by $P \in \mathbb{R}^{|V| \times 1}$, where $|V|$ is the size of vocabulary of the feature space, P is the amalgamation of both prior general sentiment knowledge denoted by $d^{(g)} \in \mathbb{R}^{|V| \times 1}$

and prior domain-specific sentiment knowledge denoted by $d^{(d)} \in \mathbb{R}^{|V| \times 1}$. If a feature v_i appears in a general-purpose sentiment lexicon then it is labeled as positive (or negative) and prior general sentiment knowledge of feature v_i is assigned $d_i^{(g)} = 1$ (or $d_i^{(g)} = -1$); otherwise, prior general sentiment knowledge is assigned 0 (i.e. $d_i^{(g)} = 0$).

In order to extract sentiment orientation of the terms that are not found in sentiment lexicon, we have followed the work of [4], and used the concept of coordinating conjunctions and adversarial conjunctions. Coordinating conjunctions are used to join two terms, which are of the same semantic importance, whereas adversarial conjunctions express opposition or contrast among two terms. For example, *for*, *and*, *nor*, *or*, etc. are coordinating conjunctions, whereas *although*, *but*, *yet*, *still*, *however*, etc. are the examples of the adversarial conjunctions. Two terms are more likely to have the same polarity orientation if they are connected using coordinating conjunctions. Moreover, if two terms that are joined via adversarial conjunctions; or they are joined by coordinating conjunctions, but before either of them a negation term like ‘not’ appears, then relation between such terms is considered have opposite polarity orientation. In order to extract domain-specific prior sentiment knowledge d , we have used the frequencies of $N_{i,j}^s$ and $N_{i,j}^o$, where $N_{i,j}^s$ is the frequency of the features v_i and v_j that share same orientation, and $N_{i,j}^o$ is the frequency of the features v_i and v_j that share opposite-orientation. $D \in \mathbb{R}^{|V| \times |V|}$ denotes the sentiment orientation of the features that are extracted from unlabeled data, and it is calculated using $N_{i,j}^s$ and $N_{i,j}^o$ for features v_i and v_j using equation (1).

$$D_{i,j} = \frac{N_{i,j}^s - N_{i,j}^o}{N_{i,j}^s + N_{i,j}^o + \alpha_0} \quad (1)$$

In equation (1), $\alpha_0 > 0$. If features v_i and v_j have a higher $N_{i,j}^s$ frequency than $N_{i,j}^o$ frequency, then it is more certain that features v_i and v_j have same orientation. In order to compute prior domain-specific sentiment knowledge $d^{(d)}$, we use equation (2).

$$d_i^{(d)} = \frac{\sum_{i \neq j} D_{i,j} * p_j^{(g)}}{\sum_{i \neq j} |D_{i,j} * p_j^{(g)}|} \quad (2)$$

Finally, in order to compute prior domain knowledge P , we amalgamate both general and domain-specific prior sentiment knowledge. For a feature v_i , $d_i^{(g)} \neq 0$ implies that feature v_i is covered by general sentiment knowledge, and we assign $P_i = d_i^{(g)}$; otherwise, it is assigned as $P_i = d_i^{(d)}$.

IV. PROPOSED APPROACH

In this section, we discuss prior domain knowledge enhanced logistic regression and formulate a machine learning model, followed by a discussion of the optimization algorithm to solve the proposed machine learning model. In rest of the paper, V and $|V|$ denote the vocabulary and size of the feature space, respectively; $d^{(g)} \in \mathbb{R}^{|V| \times 1}$ and $d^{(d)} \in \mathbb{R}^{|V| \times 1}$

denote prior general sentiment knowledge and domain-specific sentiment knowledge, respectively. $D \in \mathbb{R}^{|V| \times |V|}$ denotes the sentiment orientation of the features extracted from unlabeled data, and finally $P \in \mathbb{R}^{|v| \times 1}$ denotes the prior domain knowledge obtained from both $d^{(g)}$ and $d^{(d)}$.

A. Proposed Machine Learning Model

Given prior domain knowledge, our goal is to incorporate prior knowledge of the same domain to learn a precise and accurate classifier. Equation 3 presents the mathematical formulation of our proposed machine learning model for sentiment classification which is inspired by the models presented in [15] and [16]. In this equation, α and β are non-negative regularization constants for the prior sentiment knowledge and sentiment orientation of the features, respectively. λ is a positive constant for L_2 - norm regularization term. $\mathcal{L}(y^{(i)}, x^{(i)}; \theta)$ is the cost of error while predicting sample $x^{(i)}$ to correct label $y^{(i)}$ using prediction model θ . There are a variety of loss functions, such as *quadratic*, *hinge*, and *logistic* that are commonly used in machine learning algorithms according to the settings and types of the problems. In our proposed approach, we have used logistic loss function.

$$\operatorname{argmin}_{\theta} \frac{1}{m} \sum_{i=1}^m \mathcal{L}(y^{(i)}, x^{(i)}; \theta) + \frac{\lambda}{2m} \sum_{j=1}^n \theta_j^2 - \alpha P^T \theta - \beta \theta^T D \theta \quad (3)$$

In equation (3), prior sentiment knowledge is integrated via $-\alpha P^T \theta$. The term $P^T \theta$ is equivalent to $\sum_{i=1}^{|V|} P_i \theta_i$. Through $P^T \theta$, we ensure that if orientation of a sentiment term is positive (or negative) in prior sentiment knowledge, then its score in sentiment prediction model θ remains positive (or negative) too; otherwise, the loss function is used to penalize. We also integrate the sentiment orientation among words using $-\beta \theta^T D \theta$, where $\theta^T D \theta$ is equivalent to $\sum_{i=1}^{|V|} \sum_{j=1}^{|V|} D_{i,j} \theta_i \theta_j$. If words w_i and w_j tend to share the same sentiment (or opposite sentiment) orientations with each other according to the sentiment orientation of the features D , then our aim is to ensure that the final sentiment orientation remains in the same orientation (or opposite orientation). Otherwise, the loss function is used to penalize. We also incorporate L_2 - norm regularization, which is instigated by the elastic regularization [17]. It is added to the model for increasing its stability. If θ consists of n parameters, then the L_2 - norm is $R(\theta) = \|\theta\|_2^2 = \sum_{j=1}^n \theta_j^2$.

B. Prior Knowledge Enhanced Logistic Regression

Logistic regression [18] belongs to the family of the probabilistic classifiers, and it is commonly used in the supervised learning settings. Logistic regression is widely used to match labeled observation \hat{y} with either of the binary or one of the multiple classes. In binary classification setting, the class label is either positive or negative. Since the sentiment classification in this study is considered as a binary classification problem, we have used binary logistic regression.

Given a labeled training dataset of m samples, $\{(x^{(1)}, y^{(1)}), (x^{(2)},$

$y^{(2)}), \dots, (x^{(m)}, y^{(m)})\}$, where $x^{(i)} \in [v_1 \ v_2 \ \dots \ v_{|V|}] \in \mathbb{R}^{|V| \times 1}$ is the input sample that have $|V|$ features, and $y \in 0, 1$ is the associated labels to the samples. The output \hat{y} is assigned 1, if the input observation (sample) belongs to the expected class; otherwise, it is assigned 0. Our aim is to obtain an optimal $\theta^{|V| \times 1}$ for the training samples. Each $\theta_i \in \theta$ is a real number, and it signifies the importance of associated input feature x_i in classification model θ . Thus, we can anticipate a sentiment classification task in which feature *excellent* is positively weighted, and feature *poor* is negatively weighted. To make a decision for the computed probability, we set a threshold, which is also called decision boundary. For instance, the model produces 1 if the computed probability $p(y = 1|x) \geq 0.5$; otherwise, it produces 0.

$$\hat{y} = \begin{cases} 1, & \text{if } p(y = 1|x) \geq 0.5, \\ 0, & \text{otherwise.} \end{cases} \quad (4)$$

Equation (4) produces \hat{y} , which is the model's estimation of true y . Our aim is to produce the estimated \hat{y} as close as possible to the actual y associated with each training sample. To achieve this goal, we need to perform two steps. First, we need to measure how much the observed output \hat{y} deviates from the actual y . This metric is called a loss function or cost function, and the tool to update the weights or parameters θ is an optimization algorithm. Gradient descent is a well-known optimization algorithm, which is discussed in more details in the following sub-section.

$$h_{\theta}(x^{(i)}) = \frac{1}{1 + e^{-\theta x}} \quad (5)$$

The hypothesis for logistic regression is given by equation (5), which is a sigmoid function. The loss function of a binary classification problem for m number of data samples is formulated as equation (6), which we have taken from [18]. In order to get optimal weights or parameter vector θ , we need to minimize equation (6).

$$\begin{aligned} \text{Cost}(\theta) &= -\frac{1}{m} \sum_{i=1}^m (y^{(i)} \log(h_{\theta}(x^{(i)})) \\ &+ (1 - y^{(i)}) \log(1 - h_{\theta}(x^{(i)}))) \end{aligned} \quad (6)$$

C. Optimization Algorithm

Our objective of using gradient descent is to find optimal parameters for the proposed model formulated in equation (3) through minimizing the loss function.

$$\begin{aligned} J(\hat{\theta}) &= \operatorname{argmin}_{\theta} \frac{1}{m} \sum_{i=1}^m \mathcal{L}(y^{(i)}, x^{(i)}; \theta) + \frac{\lambda}{2m} \sum_{j=1}^n \theta_j^2 \\ &- \alpha P^T \theta - \beta \theta^T D \theta \end{aligned} \quad (7)$$

In equation (7), θ represents the parameter vector of the loss function. Gradient descent method is used to find *minima* or *maxima* of the loss function \mathcal{L} . The loss function used in logistic regression is convex, and the beauty of the convex functions is that they converge to the optimum and they do

not stuck in any local minima. It does not matter where the gradient descent algorithm starts, it inevitably converges to the optimum. Gradient descent updates parameter vector of the model by a step-size, and the value of slope $\frac{d}{d\theta}f(x; \theta)$ discounted by the learning rate η is considered as the step-size by which the gradient descent moves [18]. Parameter vector θ is iteratively updated via equation (8),

$$\begin{aligned} \theta_{j+1} = & \theta_j - \eta \frac{d}{d\theta} f(x; \theta) + \frac{\lambda}{2m} \frac{d}{d\theta} \sum_{j=1}^n \theta_j^2 \\ & - \alpha \frac{d}{d\theta} P^T \theta - \beta \frac{d}{d\theta} \theta^T D \theta \end{aligned} \quad (8)$$

The gradient $\frac{d}{d\theta}f(x; \theta)$ is an N -dimensional vector, and it points towards the sharpest slope along each of the N dimensions. Each dimension of θ_j presents the slope as a partial derivative $\frac{\partial}{\partial \theta_j}$ of the loss function. $\nabla_{\theta} \mathcal{L}(f(x; \theta), y)$ is a vector of the partials derivatives $\frac{\partial}{\partial \theta_j}$.

$$\nabla_{\theta} \mathcal{L}(f(x; \theta), y) = \begin{bmatrix} \frac{\partial}{\partial \theta_1} \mathcal{L}(f(x; \theta), y) \\ \frac{\partial}{\partial \theta_2} \mathcal{L}(f(x; \theta), y) \\ \vdots \\ \frac{\partial}{\partial \theta_n} \mathcal{L}(f(x; \theta), y) \end{bmatrix}_{N \times 1} \quad (9)$$

$$\begin{aligned} \theta_{j+1} = & \theta_j - \eta \nabla_{\theta} \mathcal{L}(f(x; \theta), y) + \frac{\lambda}{2m} \frac{d}{d\theta} \sum_{j=1}^n \theta_j^2 \\ & - \alpha \frac{d}{d\theta} P^T \theta - \beta \frac{d}{d\theta} \theta^T D \theta \end{aligned} \quad (10)$$

In order to update θ using partial derivative vector, we use equation (10). To find the optimal parameter vector θ in equation (10), the partial derivative with respect to θ is given by equation (11).

$$\begin{aligned} \frac{\partial}{\partial \theta} J(\theta) = & -\frac{1}{m} \sum_{i=1}^m (y^{(i)} \frac{\partial}{\partial \theta} \log(h_{\theta}(x^{(i)}))) \\ & + (1 - y^{(i)}) \frac{\partial}{\partial \theta} \log(1 - h_{\theta}(x^{(i)}))) \\ & + \frac{\partial}{\partial \theta} \frac{\lambda}{2m} \sum_{j=1}^n \theta_j^2 - \frac{\partial}{\partial \theta} \alpha P^T \theta - \frac{\partial}{\partial \theta} \beta \theta^T D \theta \end{aligned} \quad (11)$$

By performing several algebraic steps, we get the final equation to update θ , and we repeat equation (12) until it converges.

$$\begin{aligned} \theta_{j+1} := & \theta_j - \eta \frac{1}{m} \sum_{i=1}^m (h(x^{(i)}) - y^{(i)}) x_j^{(i)} \\ & + \frac{\lambda}{m} \sum_{j=1}^n \theta_j - \alpha P^T \theta - \beta \theta^T D \end{aligned} \quad (12)$$

V. EXPERIMENTAL SETUP AND RESULTS

In this section, we present the experimental setting and results of our proposed approach. We also present a brief discussion of the evaluation metrics that are used to evaluate

the effectiveness of our proposed approach. Finally, we present a comparative evaluation of our proposed approach with three different baseline methods.

TABLE I: Statistics of the dataset

Domain	Positive instances	Negative instances	Unlabeled instances	Total instances
Book	1,000	1,000	973,194	975,194
DVD	1,000	1,000	122,438	124,438
Electronics	1,000	1,000	21,009	23,009
Kitchen	1,000	1,000	17,856	19,856

A. Dataset

We have used a multi-domain sentiment dataset¹, which is originally collected by Blitzer et al. [19]. This dataset contains reviews of 25 different domains from Amazon, out of which we have considered only four domains viz. *Book*, *DVD*, *Electronics*, and *Kitchen* in our experiment. Following the works of Blitzer et al. [19] and Pang et al. [20], reviews having rating greater than 3 are labeled as *positive* and those having rating less than 3 are labeled as *negative*. Reviews with rating 3 are considered as *neutral* and discarded. The list of reviews in each domain is treated as an independent dataset. In every independent domain, there are 1000 positive reviews and 1000 negative reviews and some unlabeled samples. Table I presents a brief statistics of the aforementioned dataset. In our experiments, we have considered two types of samples – labeled samples and unlabeled samples. First, the given samples are preprocessed for cleaning HTML tags, accented characters, new lines, extra spaces, and special characters. Further, the samples are passed through the process of contractions and lemmatization. Thereafter, samples are tokenized and stop-words are removed.

Following preprocessing, mutual information classifier is applied to select top 5000 features. These features are used to generate the term frequency matrix of unigrams for constructing feature vectors. We kept the training and testing samples ratio as 8 : 2, and used Bing Liu’s lexicon [21] to obtain the polarities of the sentiment terms. Since general sentiment lexicon does not cover many domain-specific sentiment terms, we used sentiment polarity relations between the terms that are extracted from the unlabeled data of each domain to determine their sentiment polarity. Domain-specific knowledge is constructed using the concept of terms having the same orientation and opposite orientation. In order to extract prior domain knowledge, we used unigrams to build feature vectors. To train our proposed model, we integrated the training data with prior knowledge, and set the parameter values as $\alpha = 10$, $\beta = 0.01$, $\lambda = 10000$, and learning rate, $\eta = 0.1$. We achieved convergence after performing 200 iterations.

For performance evaluation, we have considered standard data mining metrics defined in equations (13), (14), (15), (16) in which TP, FP, and FN represent *true positives*, *false*

¹<https://www.cs.jhu.edu/~mdredze/datasets/sentiment/>

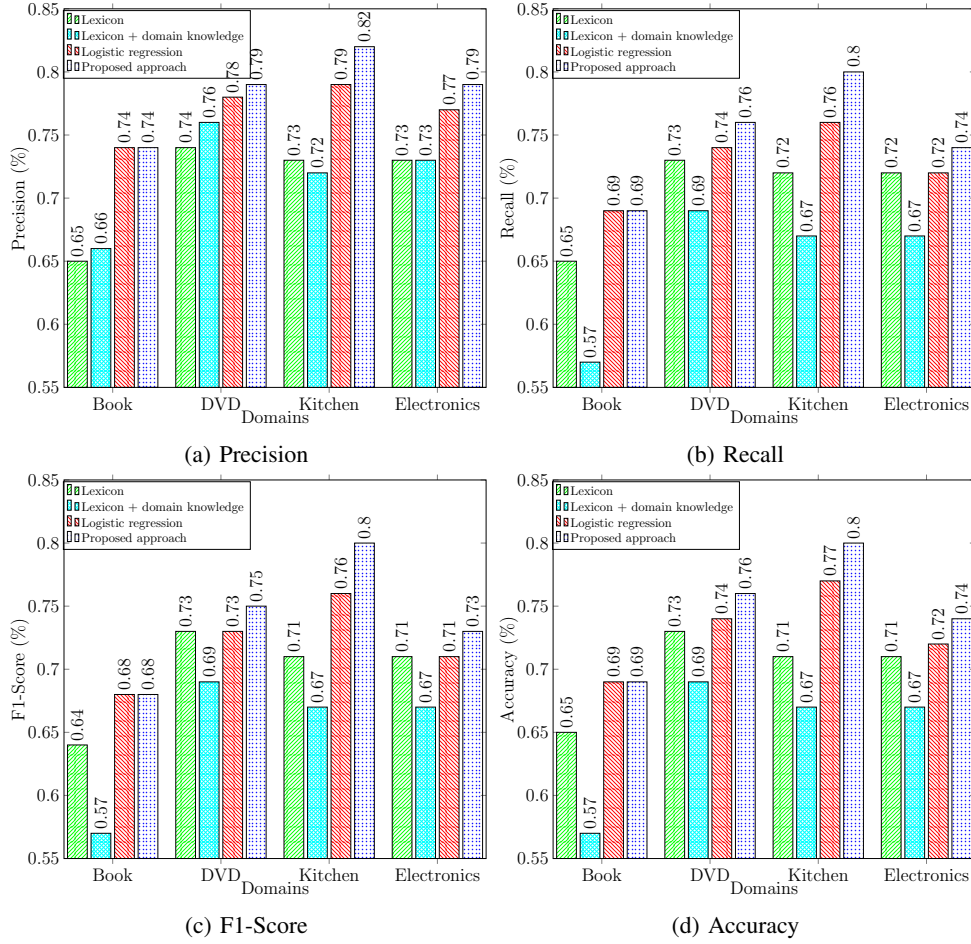


Fig. 1: Visualization of the performance evaluation results of our proposed approach and baseline methods

positives, and false negatives, respectively, and k represents the set of all classes in the dataset.

$$\text{Precision}_{macro} = \frac{1}{|k|} \sum_{c_i \in k} \frac{TP(c_i)}{TP(c_i) + FP(c_i)} \quad (13)$$

$$\text{Recall}_{macro} = \frac{1}{|k|} \sum_{c_i \in k} \frac{TP(c_i)}{TP(c_i) + FN(c_i)} \quad (14)$$

$$F1_{macro} = 2 \times \frac{\text{Precision}_{macro} \times \text{Recall}_{macro}}{\text{Precision}_{macro} + \text{Recall}_{macro}} \quad (15)$$

$$\text{Accuracy} = \frac{\# \text{correctly predicted samples}}{\# \text{total samples}} \quad (16)$$

B. Performance Evaluation Results

We present the performance evaluation results of our proposed approach with three baseline methods that are briefly discussed in the following paragraphs.

Lexicon-based sentiment classifier: In this method, we only used the prior knowledge obtained from Bing Liu’s lexicon [21], which has been compiled over the years. In total, it

contains around 6800 sentiment terms of the English language, out of which 4800 are negative terms and around 2000 are positive terms.

A hybrid classifier using lexicon and domain knowledge: In this method, we combined the prior sentiment knowledge learned from lexicon with the domain knowledge extracted from the unlabeled data samples. In this method, we combined the prior sentiment knowledge learned from lexicon with the domain knowledge extracted from the unlabeled data samples.

Logistic regression: It is used as a base model in which we did not incorporate any prior knowledge. We optimized the parameter weights using the gradient descent approach discussed in the previous sections.

The performance evaluation results of our proposed approach and aforementioned baseline methods are visualized in Figure 1. It can be observed from this figure that our proposed approach outperforms all three baseline methods over all domain-specific datasets. Table II presents domain-wise top-10 terms that are positively and negatively weighted by our proposed approach and the conventional logistic regression. It can be observed from this table that our proposed method has assigned top weights to more number of terms that are actually positive or negative terms in comparison to the conventional

TABLE II: Top positively and negatively weighted sentiment terms generated by our proposed approach and the conventional logistic regression. The correctly identified positive and negative terms are underlined

Domain	Sentiment Polarity	Top-10 Terms	
		Logistic Regression	Proposed Approach
Book	Positive	<u>great</u> , also, read, and, <u>recommend</u> , life, <u>excellent</u> , good, still, <u>love</u>	<u>great</u> , also, <u>recommend</u> , <u>excellent</u> , read, <u>love</u> , good, life, <u>wonderful</u> , and
	Negative	<u>not</u> , author, <u>no</u> , <u>bad</u> , even, <u>but</u> , would, page, like, say,	<u>not</u> , author, <u>no</u> , <u>bad</u> , <u>but</u> , even, <u>waste</u> , page, would, <u>nothing</u>
DVD	Positive	<u>great</u> , <u>good</u> , <u>love</u> , see, well, enjoy, batman, really, always, play	<u>great</u> , <u>good</u> , <u>love</u> , see, enjoy, <u>well</u> , always, keaton, job, life
	Negative	<u>bad</u> , <u>not</u> , <u>no</u> , <u>could</u> , talk, nothing, <u>would</u> , <u>waste</u> , like, look	<u>bad</u> , <u>not</u> , <u>no</u> , <u>could</u> , nothing, <u>waste</u> , try, talk, instead, plot
Electronics	Positive	<u>great</u> , <u>good</u> , price, <u>little</u> , like, as, well, easy, use, need,	<u>great</u> , <u>good</u> , price, like, <u>easy</u> , <u>well</u> , <u>little</u> , <u>love</u> , as, <u>happy</u>
	Negative	<u>not</u> , <u>return</u> , <u>would</u> , buy, support, customer, product, <u>bad</u> , try, get	<u>not</u> , <u>return</u> , <u>would</u> , buy, <u>bad</u> , customer, product, <u>waste</u> , try, support
Kitchen	Positive	<u>great</u> , easy, as, good, <u>love</u> , perfect, use, need, little, price	<u>great</u> , easy, good, <u>love</u> , perfect, as, price, use, <u>nice</u> , find
	Negative	<u>not</u> , <u>return</u> , product, <u>back</u> , month, send, <u>break</u> , time, item, <u>would</u>	<u>not</u> , <u>return</u> , <u>break</u> , product, month, <u>back</u> , <u>waste</u> , send, time, item

logistic regression. It confirms that our proposed is effective for document-level sentiment classification.

VI. CONCLUSION

In this paper, we have proposed a prior domain knowledge enhanced sentiment classification approach for document-level sentiment analysis. We have used two types of prior knowledge and incorporated in the training samples to train sentiment classifier, and gradient descent method is employed to optimize the modified logistic regression. The experimental results on a multi-domain sentiment dataset suggest that our proposed approach significantly improves the task of document-level sentiment classification and performs significantly better in comparison to the baseline methods. As a result, it can be concluded that utilizing prior domain knowledge during a classifier’s training phase is more effective than using the classifier without any prior domain knowledge for document-level sentiment classification.

REFERENCES

- [1] M. Abulaish, M. Rahimi, H. Ebrahemi, and A. K. Sah, “Sentilangn: A language-neutral graph-based approach for sentiment analysis in microblogging data,” in *Proceedings of the 18th IEEE/WIC/ACM International Conference on Web Intelligence (WI), Thessaloniki, Greece*, pp. 461–465, October 14–17, 2019.
- [2] A. Kamal, M. Abulaish, and Jahiruddin, “Ontolsa – an integrated text mining system for ontology learning and sentiment analysis,” in *Sentiment Analysis and Ontology Engineering, Studies in Computational Intelligence 639* (W. Pedrycz and S.-M. Chen, eds.), pp. 399–423, Springer, 2016.
- [3] A. Kamal and M. Abulaish, “Statistical features identification for sentiment analysis using machine learning techniques,” in *Proceedings of the International Symposium on Computational and Business Intelligence, Delhi, India*, pp. 178–181, IEEE, August 24–26, 2013.
- [4] V. Hatzivassiloglou and K. R. McKeown, “Predicting the semantic orientation of adjectives,” in *Proceedings of the 35th Annual Meeting of the ACL and 8th Conference of the European Chapter of the ACL*, pp. 174–181, 1997.
- [5] A. Andreevskaia and S. Bergler, “When specialists and generalists work together: Overcoming domain dependence in sentiment tagging,” in *Proceedings of the 46th Annual Meeting of the ACL*, pp. 290–298, 2008.
- [6] T. Wilson, J. Wiebe, and P. Hoffmann, “Recognizing contextual polarity in phrase-level sentiment analysis,” in *Proceedings of the Conference on Human Language Technology and EMNLP*, p. 347–354, 2005.
- [7] P. Melville, W. Gryc, and R. D. Lawrence, “Sentiment analysis of blogs by combining lexical knowledge with text classification,” in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, p. 1275–1284, 2009.
- [8] Y. Dang, Y. Zhang, and H. Chen, “A lexicon-enhanced method for sentiment classification: An experiment on online product reviews,” *IEEE Intelligent Systems*, vol. 25, no. 4, p. 46–53, 2010.
- [9] T. Li, Y. Zhang, and V. Sindhwani, “A non-negative matrix tri-factorization approach to sentiment classification with lexical prior knowledge,” in *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, p. 244–252, 2009.
- [10] J. Fang and B. Chen, “Incorporating lexicon knowledge into SVM learning to improve sentiment classification,” in *Proceedings of the Workshop on Sentiment Analysis where AI meets Psychology (SAAIP)*, pp. 94–100, 2011.
- [11] Y. He, “Incorporating sentiment prior knowledge for weakly supervised sentiment analysis,” *ACM Transactions on Asian Language Information Processing (TALIP)*, vol. 11, no. 2, pp. 1–19, 2012.
- [12] D. M. Blei, A. Y. Ng, and M. I. Jordan, “Latent dirichlet allocation,” *Journal of machine Learning research*, vol. 3, no. null, p. 993–1022, 2003.
- [13] H. Han, J. Zhang, J. Yang, Y. Shen, and Y. Zhang, “Generate domain-specific sentiment lexicon for review sentiment analysis,” *Multimedia Tools and Applications*, vol. 77, no. 16, pp. 21265–21280, 2018.
- [14] K. Al-Rowaily, M. Abulaish, N. A.-H. Haldar, and M. Al-Rubaiana, “Bisal – a bilingual sentiment analysis lexicon to analyze dark web forums for cyber security,” *Digital Investigation*, vol. 14, pp. 53–62, 2015.
- [15] F. Wu, S. Wu, Y. Huang, S. Huang, and Y. Qin, “Sentiment domain adaptation with multi-level contextual sentiment knowledge,” in *Proceedings of the 25th ACM International Conference on Information and Knowledge Management*, p. 949–958, 2016.
- [16] F. Wu, C. Wu, and J. Liu, “Imbalanced sentiment classification with multi-task learning,” in *Proceedings of the 27th CIKM*, p. 1631–1634, 2018.
- [17] H. Zou and T. Hastie, “Regularization and variable selection via the elastic net,” *Journal of the royal statistical society: series B (statistical methodology)*, vol. 67, no. 2, pp. 301–320, 2005.
- [18] D. Jurafsky and J. H. Martin, *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice Hall PTR, 1st ed., 2000.
- [19] J. Blitzer, M. Dredze, and F. Pereira, “Biographies, Bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification,” in *Proceedings of the 45th Annual Meeting of the ACL*, pp. 440–447, 2007.
- [20] B. Pang, L. Lee, and S. Vaithyanathan, “Thumbs up? sentiment classification using machine learning techniques,” in *Proceedings of the EMNLP*, pp. 79–86, 2002.
- [21] M. Hu and B. Liu, “Mining and summarizing customer reviews,” in *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, p. 168–177, 2004.