SSentiaA: A Self-Supervised Sentiment Analyzer for Classification From Unlabeled Data

Salim Sazzed  
*Old Dominion University*

Sampath Jayarathna  
*Old Dominion University*

Original Publication Citation
SSentiA: A Self-supervised Sentiment Analyzer for classification from unlabeled data

Salim Sazzed, Sampath Jayarathna

Old Dominion University, Norfolk, VA, USA

ARTICLE INFO


ABSTRACT

In recent years, supervised machine learning (ML) methods have realized remarkable performance gains for sentiment classification utilizing labeled data. However, labeled data are usually expensive to obtain, thus, not always achievable. When annotated data are unavailable, the unsupervised tools are exercised, which still lag behind the performance of supervised ML methods by a large margin. Therefore, in this work, we focus on improving the performance of sentiment classification from unlabeled data. We present a self-supervised hybrid methodology SSentiA (Self-supervised Sentiment Analyzer) that couples an ML classifier with a lexicon-based method for sentiment classification from unlabeled data. We first introduce LRSentiA (Lexical Rule-based Sentiment Analyzer), a lexicon-based method to predict the semantic orientation of a review along with the confidence score of prediction. Utilizing the confidence scores of LRSentiA, we generate highly accurate pseudo-labels for SSentiA that incorporates a supervised ML algorithm to improve the performance of sentiment classification for less polarized and complex reviews. We compare the performances of LRSentiA and SSentiA with the existing unsupervised, lexicon-based and self-supervised methods in multiple datasets. The LRSentiA performs similarly to the existing lexicon-based methods in both binary and 3-class sentiment analysis. By combining LRSentiA with an ML classifier, the hybrid approach SSentiA attains 10%–30% improvements in macro F1 score for both binary and 3-class sentiment analysis. The results suggest that in domains where annotated data are unavailable, SSentiA can significantly improve the performance of sentiment classification. Moreover, we demonstrate that using 30%–60% annotated training data, SSentiA delivers similar performances of the fully labeled training dataset.

1. Introduction

Sentiment Analysis, also known as opinion mining, is the process of categorizing opinions expressed (e.g., positive or negative) in a text document. With the advancement of web 2.0 and escalating popularity of social media, a vast amount of user-generated content regarding products, events, services, etc., has now become available. This paradigm shift necessitates sophisticated computational tools to extract insights from these data. Due to its wide applicability in various real-world problems, sentiment analysis is a well-suited solution for pattern mining in user-generated data. Sentiment analysis has been applied in various domains such as product reviews (Fang & Zhan, 2015; Turney, 2002), restaurant reviews (Kang, Yoo, & Han, 2012), movie recommendation (Turney, 2002), drama review (Sazzed, 2020a), micro-blogs posts (Musto, Semeraro, & Polignano, 2014), election results prediction (Tumasjan, Sprenger, Sandner, & Welpe, 2010), and stock market predictions (Mittal & Goel, 2012).

Researchers have analyzed sentiments incorporated in textual data at various levels, such as aspect, sentence, or document level. In document-level sentiment analysis, the objective is to classify the opinion expressed in a document as positive or negative by considering the whole document. Sentence-level sentiment analysis determines whether a sentence conveys a positive or negative opinion. To determine the sentiment at a more granular level, such as towards an entity, aspect-based sentiment analysis is used.

Both the lexicon-based (Musto et al., 2014; Sazzed, 2020b; Taboada, Brooke, Tofiloski, Voll, & Stede, 2011) and machine learning-based approaches (Gamon, 2004; Go, Bhayani, & Huang, 2009; Liu et al., 2010; Sazzed & Jayarathna, 2019; Tripathy, Agrawal, & Rath, 2016) have been investigated in various studies. Besides, researchers proposed various hybrid approaches (Apel, Chiclana, Carter, & Fujita, 2016; Malandrakis, Kazemzadeh, Potamianos, & Narayanan, 2013; Mudinas, Zhang, & Levene, 2012) combining both. Lexicon-based methods rely on linguistic resources such as sentiment lexicons composed of words and corresponding polarity values. The opinion-conveying positive or negative terms are used to evaluate polarity in a text without using any labeled data (Turney, 2002). Opinion words can express desirable (e.g., good, awesome, etc.) or undesirable (e.g., terrible, pathetic, etc.)
states. Sentiment lexicons can be binary (e.g., (Hu & Liu, 2004)) such as +1 for positive words and -1 for negative words, or it can also contain words associated with the sentiment intensity score (e.g., AFINN (Nielsen, 2011), SentiWordNet (Baccianella, Esuli, & Sebastiani, 2010), and SenticNet (Cambria, Speer, Havasi, & Hussain, 2010)).

Both approaches of sentiment analysis (i.e., lexicon-based and machine learning-based) have their strengths and weaknesses. The lexicon-based methods have an advantage over supervised ML-based methods as they do not require labeled data for predicting unseen instances. The cost associated with the data labeling process makes a fully labeled training set often infeasible, whereas unlabeled data can be obtained relatively easily. Besides, the accuracy of supervised ML methods can vary across domains and can be affected by parameter tuning. However, the lexicon-based approaches have their limitations too. The explicit lexicon-based method often cannot distinguish the classes when the margin between classes is too small. The complexity and noise present in the dataset can also affect the performance of the lexicon-based system. In contrast, the supervised ML methods learn implicit patterns from the labeled data, thus show better performance in determining the polarity of complex cases.

Combining both approaches to form a hybrid classifier can increase robustness, accuracy, and overall generalization capability of sentiment classification. The integration step fuses learning-based methods to rule-based methods in both the lexicon generation and the sentiment classification stage. In the lexicon generation step, machine learning-based approaches have been applied for determining the weights of opinion words (Cambria et al., 2010; Thelwall, Buckley, Paltoglou, Cai, & Kappas, 2010). In the prediction step, the hybrid method can utilize lexical features (Melville, Gryc, & Lawrence, 2009; Pandey, Rajpoot, & Saraswat, 2017) or limited labeled data for supervised training to combine with the rule-based method (Andreevskaia & Bergler, 2008).

1.1. Motivation and research objective

Although the lexicon-based methods do not require labeled data, they yield low accuracy due to various reasons such as lexicon coverage problem, considering word-level polarity without context, etc. Therefore, over the years, a plethora of research has been conducted on hybrid sentiment analysis leveraging fully or partially labeled data set along with the polarity lexicon.

However, only a few works (He & Zhou, 2011; Qiu, Zhang, Hu, & Zhao, 2009; Tan, Wang, & Cheng, 2008; Zhang, Ghosh, Dekhil, Hsu, & Liu, 2011; Zhang & He, 2013; Zhang, Zhao, Qiu, & Hu, 2009) introduced methods that work without using any labeled data (i.e., self-supervised). Even these existing self-supervised methods have several limitations such as:

- They were applied to either very few datasets or small datasets with few thousands of samples or datasets from a single domain. Therefore, their effectiveness in large datasets or multi-domain datasets is not known.
- Their applicability was only shown for binary-level sentiment analysis.
- The correlation between the chosen threshold value used for pseudo-label selection and the accuracy of the selected pseudo-labels was not provided.

Therefore, the efficacy of the self-learning approach in unlabeled data needs to be analyzed further from several perspectives. In this research, we aim to address the missing pieces which have not been investigated in the existing study. Our research objectives are as follows:

- Introduce a method that does not require manually annotated data in the learning process.
- Show the applicability of the self-supervised approach in large datasets from multiple domains.
- Investigate the efficacy of self-supervised learning for sentiment classification at a finer-level of granularity (i.e., 3-class classification).
- Infer how to select the highly accurate pseudo-labels from a lexicon-based method to minimize error propagation into ML classifiers.

We propose a self-supervised hybrid methodology, SSentiA, which functions in a fully unsupervised manner without using any labeled data. SSentiA can be applied to both sentence-level and document-level sentiment classification (i.e., review with one or multiple sentences). We evaluate SSentiA using four large review datasets, TripAdvisor, IMDB, Amazon, and Clothing and one small dataset, Cornell movie review. The datasets are selected from multiple domains to show the efficacy of SSentiA. In addition to applying SSentiA for binary classification, we explore its performance in 3-class sentiment analysis.

SSentiA operates in two steps: first, it employs a lexicon-based classifier, LRSentiA, to generate highly accurate pseudo-labels. LRSentiA predicts the semantic orientation of a review utilizing a binary-level polarity lexicon. Additionally, it provides the confidence score of the prediction. The Chi-squared test shows a correlation between the confidence score and accuracy of the predictions of LRSentiA; hence predictions with high confidence scores are highly accurate and can be used as pseudo-label. Next, SSentiA utilizes the highly confident predictions gleaned from LRSentiA. SSentiA employs these pseudo-labels to train a supervised ML classifier and predict the semantic orientation of the low-confident reviews of LRSentiA. We demonstrate that significant improvement in sentiment classification can be achieved by employing our hybrid model in the scenario when labeled data are not available. Moreover, utilizing limited labeled data, SSentiA shows a comparable performance of a fully annotated dataset, thus reduce the time and labor needed for data annotation.

1.2. Contributions

The main contributions of our work can be summarized as follows:

- We present a hybrid sentiment analysis approach, SSentiA, for sentiment classification from unlabeled data. SSentiA utilizes pseudo-labels to train a supervised ML classifier for sentiment classification.
- We introduce a lexicon-based method LRSentiA to generate highly accurate pseudo-labels. LRSentiA provides sentiment orientation of the review as well as the confidence of predictions.
- We show an effective way to reduce the error-propagation from pseudo-label using the confidence score. We demonstrate that there is a correlation between the confidence score and the accuracy of the generated pseudo-label. Utilizing this information, we can balance between the size of the training set and error-propagation to the ML classifier.
- We provide a comparative performance evaluation of SSentiA with a number of lexicon-based, unsupervised, and self-supervised methods and demonstrate that significant performance gains can be attained by employing SSentiA.
- Besides, we show that by utilizing a small number of labeled samples, SSentiA can achieve comparable performances of the fully labeled dataset.

2. Related works

Most of the existing works related to sentiment analysis have been performed in the following two settings:

- Supervised setting: This approach utilizes labeled data to build a classification model and then infers the polarity of the unseen data.
### 2.1. Unsupervised (no labeled data) approaches

In (Taboada et al., 2011), the authors proposed lexicon-based Semantic Orientation CALCulator (SO-CAL) that uses dictionaries of sentiment annotated words along with intensification and negation. Turney (2002) utilized a lexicon-based method to identify the sentiment polarity from the reviews of automobiles, banks, movies, and travel destinations. A rule-based model, VADER (Gilbert & Hutto, 2014), was introduced for analyzing sentiments from social media data. The applications of lexicon-based approaches span various domains such as Twitter (Jurek, Mulvanna, & Bi, 2015), blog (Melville et al., 2009), product reviews (Ding, Liu, & Yu, 2008), tourism (García, Gaines, Linaza, et al., 2012), etc.

Zhou, Zhao, and Zeng (2014) proposed a graph-based algorithm called graph co-regularized non-negative matrix tri-factorization (GNTF). They assumed two words (or documents) sufficiently close to each other convey the same sentiment polarity. Utilizing the nearest neighbor graphs, they encoded the geometric information. Finally, they introduced an algorithm for learning the factorization, analyzed its complexity, and provided proof of convergence. Their experimental results on two datasets showed GNTF provided higher accuracy compared to the state-of-the-art methods. In (Fernández-Gavilanes, Álvarez-López, Juncal-Martínez, Costa-Montenegro, & González-Castaño, 2016), the authors proposed an unsupervised dependency parsing-based text classification method for predicting sentiment in online textual messages. They utilized various natural language processing tools and derived sentiment features from sentiment lexicons. The applied their method to Cornell movie review, Obama-McCain debate, and SemEval-2015 datasets and achieved competitive performance.

In (Sneath et al., 2012), the authors proposed an unsupervised dependency parsing-based text classification method for predicting sentiment in online textual messages. They utilized various natural language processing tools and derived sentiment features from sentiment lexicons. The applied their methods to Cornell Movie Review, Obama-McCain Debate, and SemEval-2015 datasets and achieved competitive performance.

SentiCircles (Saif, He, Fernandez, & Alani, 2016) is a lexicon-based method for classifying sentiment from Twitter data. SentiCircles considers the co-occurrence patterns of words in different contexts to capture their semantics and update the pre-assigned strength and polarity in sentiment lexicons accordingly. The authors evaluated SentiCircles on three Twitter datasets using three different sentiment lexicons. They found significant improvement over the baselines in terms of both accuracy and F-measure for tweet-level sentiment analysis. Among the three datasets, their approach performed better than the SentiStrength in two datasets.

SmartSA (Muhammad, Wiritunga, & Lothian, 2016) is a lexicon-based method that extracts the term polarity utilizing sentiment lexicons and aggregates such scores to predict the overall sentiment. As a general-purpose lexicon, the authors used SentiWordNet with genre-specific vocabulary and sentiment, as well as global and local context. When evaluated on the diverse social media data, their method showed improved performance compared to SentiStrength. Jiménez-Zafra, Martín-Valdivia, Martínez-Cámara, and Ureña-López (2016) proposed an unsupervised approach for aspect-based sentiment analysis (ABSA). They utilized a knowledge base to extract various aspects. Employing grammatical relationships and a lexicon-based approach, they classified the sentiments of various aspects. They presented a study of three well-known sentiment lexicons, Opinion Lexicon (Hu & Liu, 2004), MPQA (Wilson, Wiebe, & Hoffmann, 2005), and SentiWordNet (Baccianella et al., 2010) to determine the best combination for polarity classification at aspect-level.

Vilares, Gómez-Rodríguez, and Alonso (2017) proposed a framework for multilingual sentiment analysis utilizing compositional syntax-based rules. Their experiments showed improvement over both the existing unsupervised methods and state-of-the-art supervised models when evaluating outside their corpus of origin. Vashishtha and Susan (2019) determined the sentiments of social media posts exploiting a set of fuzzy rules and several lexicons. Their proposed fuzzy system integrates Word Sense Disambiguation (WSD) with nine fuzzy rule-based systems to classify posts into positive, negative, or neutral sentiment class. They applied their system into nine public twitter datasets, three sentiment lexicons and compared with four state-of-the-art unsupervised sentiment analysis approach, and one state-of-the-art supervised ML method.

### 2.2. Supervised or semi-supervised (labeled data) hybrid approaches

Supervised ML methods have been employed in many studies (Agarwal & Mittal, 2016; Liu et al., 2010). Researchers used ML classifiers in isolation using fully labeled dataset (Gamon, 2004; Go et al., 2009).
or combined them with the other approaches using a partially or fully labeled set (Appel et al., 2016; Malandrakis et al., 2013; Mudinas et al., 2012).

Appel et al. (2016) used a sentiment lexicon enhanced with Sentiment-WordNet (Baccianella et al., 2010) to classify sentiment in sentence-level. In (Ghiassi, Skinner, & Zimbra, 2013), the authors utilized fuzzy sets to assess both the semantic orientation and intensity to develop a Twitter-specific lexicon for sentiment analysis. SentBuk (Ortigosa, Martín, & Carro, 2014), an application for hybrid sentiment analysis in Facebook, reported accuracy of 83.27%. pSenti (Mudinas et al., 2012) is a concept-level sentiment analysis system that integrated both the lexicon-based and learning-based approaches for opinion mining. They achieved good accuracy in both sentiment polarity classification and strength detection for CNET and IMDB movie review datasets.

In (Malandrakis et al., 2013), the authors proposed a hybrid approach for sentiment analysis in Twitter data. Their model used a lexicon generated from a large corpus. They consolidated the lexicon-based model with a maximum entropy-based classifier trained on a large dataset. The two models are combined at the posterior level to generate the final output. Similarly, Xiang and Zhou (2014) proposed improvement over the Twitter sentiment analysis with the help of a topic-based mixture modeling approach with semi-supervised training. In (Prabowo & Thebwall, 2009), the authors introduced an approach that combines rule-based classification, supervised learning, and machine learning into a new hybrid model, and tested it on movie reviews, product reviews, and MySpace comments. Becker, Erhart, Skiba, and Matula (2013) proposed linear classifiers with a combination of lexical and syntactic features and automatically labeled a large Twitter dataset. They utilized the automatically labeled tweet to discover prior polarities of words and to provide additional training examples for self-training. Their approach in expanding the polarity lexicon and augmenting the training data with unlabeled tweets can yield performance improvement.

ALDONAr, a hybrid solution for sentence-level aspect-based sentiment analysis was proposed by Meškèle and Frasincar (2020). They used a lexicalized domain ontology and a neural attention model. Their manually created lexicalized domain ontology is integrated to utilize the domain-specific knowledge. ALDONAr uses BERT word embeddings, regularization, Adam optimizer, and different model initialization.

Schouten and Frasincar (2018) presented a hybrid approach for aspect-based sentiment analysis. They proposed a knowledge-driven approach that complements traditional machine learning methods. By using domain knowledge encoded in an ontology, they improved the sentiment analysis of a given aspect. Cai et al. (2019) constructed a domain-specific three-layered sentiment dictionary with entities, aspects, and sentiment words. They employed a stacking approach to combine SVM and GBDT and achieved a better performance than baseline single models. da Silva, Coletta, Hruschka, and Hruschka (2016) combined Support Vector Machines (SVM), constructed from labeled data, with the information provided by the pair-wise similarities between unlabeled data points. Their proposed framework is based on an iterative self-training approach. Their results show that the use of unlabeled tweets improves classification performance when a few labeled tweets are available.

Lee and Kim (2017) proposed a sentiment labeling approach with a joint sentiment/topic model (JST). Their semi-supervised sentiment classification framework adds pseudo-labeled instances to the training corpora by filtering confidently predicted instances. To exploit the sufficient number of unlabeled instances, they conducted self-training with a concatenated vector that complements the document and polarity vectors.

SAIL (Malandrakis et al., 2013) model uses a lexicon automatically generated from a very large web corpus. The word and bigram affective ratings were calculated and used as features of a Naïve Bayes (NB) tree model. In the unconstrained scenario, the authors combined the lexicon-based model with a classifier built on maximum entropy language models and trained on a large external dataset. The two models were fused at the posterior level to produce a final output. Their approach performed well in Twitter sentiment analysis in both constrained and unconstrained scenarios.

Giatsoglou et al. (2017) proposed a hybrid approach for the prediction of sentiment by combining the context-sensitive Word2Vec with the sentiment lexicon. The resulting hybrid representations are then used as inputs for the supervised training of a classifier. They tested their approach in four different text corpora in Greek and English, along with different coding schemes and different classifier models. They found the SVM model with a linear kernel achieved the best results in terms of efficiency in accuracy and the process times.

Araque, Corcuera-Platas, Sánchez-Rada, and Iglésias (2017) presented a methodology for sentiment analysis that performs surface and deep features integration. Their ensemble techniques combined several sentiment classifiers trained with different kinds of features. They utilized six datasets from two domains: Twitter and movie reviews.

Yu, Wang, Lai, and Zhang (2017) proposed a word vector refinement model to improve the existing word embeddings such as Word2vec and GloVe. Their proposed approach adjusts the vector representations of words so that semantically and sentimentally similar words come closer. They applied the refined word-embedding with CNN, LSTM, and DNN in Stanford Sentiment Treebank (SST) and achieved improvement over conventional word-embeddings in both binary and fine-grained sentiment classification.

### 2.3. Self-supervised approaches (pseudo-labeled data)

Zhang et al. (2011) presented an entity-level sentiment analysis method for Twitter. Their method first adopt a lexicon-based approach to perform sentiment analysis. They found that the lexicon-based method provided high precision, but low recall. To improve recall, they identified additional tweets that are likely to be opinionated by the lexicon-based method. A classifier is then trained to assign polarities to the entities in the newly identified tweets. Instead of being labeled manually, the training examples are given by the lexicon-based approach. Their experimental results showed that the proposed method improved the F-score and outperformed the state-of-the-art baselines.

He and Zhou (2011) proposed a framework where an initial classifier is learned by incorporating a sentiment lexicon and using generalized expectation criteria. They utilized documents classified with high confidence as pseudo-labeled examples for automatical domain-specific feature acquisition. The word-class distributions of self-learned features are estimated from the pseudo-labeled examples. They trained another classifier by constraining the model’s predictions on unlabeled instances. Their framework was evaluated on two small datasets, the movie-review (MR) dataset and the multi-domain sentiment (MDS) dataset (each contains 2000 reviews), and attained comparable performance with other weakly-supervised sentiment classification methods.

SESS (Self-Supervised and Syntax-Based method) was proposed by (Zhang et al., 2009) that consists of three phases. In the first phase, some documents are classified based on a sentiment dictionary, and then the sentiments of phrases and documents are iteratively revised. In the second phase, a machine learning model is trained with the labeled data from the first phase. In the third phase, for the final classification, the learned model is applied to the whole data set. Their datasets span four domains, where each domain contains 1000 positive and 1000 negative documents.

Zhang and He (2013) introduced a weakly-supervised approach for Chinese sentiment classification. They applied a variant of a self-training algorithm to train an initial classifier. Later they utilized a pseudo-labeled training set and adopted a standard self-learning cycle to obtain the overall classification results.

Qiu et al. (2009) proposed SELC Model for the sentiment classification in Chinese. The SELC Model is comprised of two phases; In the first
phase, a sentiment dictionary is used to classify reviews. In the second phase, a supervised classifier is trained by utilizing some of the reviews predicted in the first phase. Then the supervised classifier predicts other reviews and improves the results produced in the first phase. They applied the SELC model to a dataset of 7779 Chinese product reviews and achieved an improvement of 6.63% in F1 score over the previous best result.

2.4. Comparison of SSentiA and existing self-supervised methods

Table 1 shows the details of the existing self-supervised methods and SSentiA. SSentiA differs from the existing self-supervised methods in several perspectives. The key methodological differences exist in the following aspects: pseudo-label generation and selection, training-testing data splitting, the granularity of classification, and sentiment lexicon used. In the evaluation phase, the differences come from the assessment of SSentiA on large datasets and classification at both binary and ternary levels.

<table>
<thead>
<tr>
<th>Method</th>
<th>Language</th>
<th>#Class</th>
<th>Granularity</th>
<th># Samples</th>
<th>Lexicon</th>
<th>ML Classifier</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Zhang et al., 2011)</td>
<td>English</td>
<td>2/3</td>
<td>Document/Sentence</td>
<td>95150/54800</td>
<td>Opinion Lexicon</td>
<td>SVM/LR</td>
</tr>
<tr>
<td>(He &amp; Zhou, 2011)</td>
<td>English</td>
<td>2</td>
<td>Document</td>
<td>7,779</td>
<td>HowNet</td>
<td>SVM</td>
</tr>
<tr>
<td>(Zhang et al., 2009)</td>
<td>English</td>
<td>2</td>
<td>Document</td>
<td>23203</td>
<td>HowNet</td>
<td>SVM</td>
</tr>
<tr>
<td>(Tan et al., 2008)</td>
<td>Chinese</td>
<td>2</td>
<td>Document</td>
<td>7,779</td>
<td>HowNet</td>
<td>SVM</td>
</tr>
<tr>
<td>(Zhang &amp; He, 2013)</td>
<td>Chinese</td>
<td>2</td>
<td>Document</td>
<td>7,779</td>
<td>HowNet</td>
<td>SVM</td>
</tr>
<tr>
<td>(Qiu et al., 2009)</td>
<td>Chinese</td>
<td>2</td>
<td>Document</td>
<td>7,779</td>
<td>HowNet</td>
<td>SVM</td>
</tr>
</tbody>
</table>
| S. Sazzed and S. Jayarathna Machine Learning with Applications 4 (2021) 100026

Table 2 The description of 2-class datasets.

<table>
<thead>
<tr>
<th>Domain</th>
<th>Negative</th>
<th>Positive</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>TripAdvisor</td>
<td>9520</td>
<td>9520</td>
<td>19040</td>
</tr>
<tr>
<td>IMDB</td>
<td>12500</td>
<td>21987</td>
<td>34487</td>
</tr>
<tr>
<td>Amazon MP3</td>
<td>6482</td>
<td>21987</td>
<td>28469</td>
</tr>
<tr>
<td>Clothing</td>
<td>4101</td>
<td>18540</td>
<td>22641</td>
</tr>
<tr>
<td>Cornell</td>
<td>1000</td>
<td>1000</td>
<td>2000</td>
</tr>
</tbody>
</table>

Table 3 The description of 3-class datasets.

<table>
<thead>
<tr>
<th>Domain</th>
<th>Negative</th>
<th>Neutral</th>
<th>Positive</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>TripAdvisor</td>
<td>9520</td>
<td>4760</td>
<td>9520</td>
<td>23800</td>
</tr>
<tr>
<td>Amazon</td>
<td>6482</td>
<td>2531</td>
<td>21987</td>
<td>31000</td>
</tr>
</tbody>
</table>

8 & Zhai, 2011), Clothing and Cornell movie review. The TripAdvisor hotel dataset consists of 23600 reviews with 5 different ratings (−4, −2, 0, 2, 4). We consider reviews with a rating below 0 as negative, above 0 as positive, and 0 as neutral. At binary-level, the dataset uses positive and negative reviews, while the 3-class dataset includes neutral reviews as well. The IMDB dataset is originally a binary-level dataset with 12500 positive and 12500 negative reviews; therefore, we use it only for binary-level classification. The Amazon MP3 dataset consists of user reviews with ratings between 1 to 5. To categorize the reviews into the binary level, we consider rating 1–2 as a negative and 4–5 as positive (Maas et al., 2011; Pang & Lee, 2004, 2005). This binary dataset contains a total of 28469 customer reviews, 21987 positive and 6482 negative. The 3-class dataset includes additional 2531 neutral reviews with ratings of 3 (Mudinas et al., 2012; Pang & Lee, 2004, 2005). The binary-level Clothing dataset comprised of 18539 positive reviews and 4101 negative reviews, totaling 22640 reviews. Although The Cornell movie review dataset is very small, consists of only 1000 positive and 1000 negative reviews, we use it as some of the state-of-the-art methods reported their results on this dataset, therefore, allow us to compare the results. Among the five binary datasets, IMDB, TripAdvisor, and Cornell datasets are class-balanced, while other datasets contain mostly positive reviews. None of the 3-class datasets is class-balanced (see Table 2 and Table 3).

To convert the 5-point scale system (i.e., reviews with a rating between 1 and 5) to a ternary dataset, we follow the class-labeling procedure of existing literature. Maas et al. (2011), Mudinas et al. (2012), Pang and Lee (2004). Based on Mudinas et al. (2012), Pang and Lee (2004), in a 5-points scale system, a rating of 1 or 2 is considered as negative, 3 as neutral, and 4 or 5 as positive.

4. Baseline comparison

We compare the performance of SSentiA with the existing tools and classifiers in two different settings.

1. Without using any labeled data (which is the main focus of this work).

---

2. Utilizing limited labeled data

Although we show that SSentiA can be utilized as a semi-supervised tool using limited labeled data, the main focus of this study is to show the efficacy of SSentiA for classifying sentiment in unlabeled data.

4.1. Setting 1: Without using any labeled data

We compare both LRSentiA and SSentiA with a number of state-of-the-art lexicon-based, unsupervised and self-supervised methodologies that do not use any labeled data.

4.1.1. Comparison with fully lexicon-based methods

Some of the state-of-the-art lexicon-based sentiment analysis tools and lexicons: AFINN (Nielsen, 2011), SentiStrength (Thelwall et al., 2010), VADER (Gilbert & Hutto, 2014), TextBlob and Opinion-Finder (Wilson et al., 2005) are utilized. These lexicon-based tools are capable of classifying sentiments from unlabeled data. They are selected based on their performances on a number of benchmark comparisons (Ahmed Abbasi & Dhar, 2014; Ribeiro, Araújo, Gonçalves, & Benevenuto, 2016).

The AFINN (Nielsen, 2011) lexicon consists of a list of English terms manually rated for valence by Finn Årup Nielsen. Each term is represented by an integer between −5 (strongly negative) and +5 (strongly positive). We utilize the PyPI implementation of AFINN to compute the polarity of the review. For binary classification, a non-negative polarity score indicates a positive class. For 3-class, a polarity score above 0 refers to positive prediction, below 0 refers to negative prediction, and a 0 polarity score means neutral class prediction.

SentiStrength (Thelwall et al., 2010) is a sentiment analysis tool that employs multiple methods to simultaneously extract positive and negative sentiment strength from the short informal text. It uses a dictionary of sentiment words with associated polarity strength. SentiStrength utilizes a dual 5-point system for positive and negative sentiment and leverages machine learning methods to optimize sentiment term weightings. The predicted class of a review is determined following the same procedure of AFINN. We utilize the PyPI SentiStrength package.

VADER (Valence Aware Dictionary and sEntiment Reasoner) (Gilbert & Hutto, 2014) is a lexicon and rule-based sentiment analysis tool specifically attuned to sentiments expressed in social media. VADER combines the lexical features with five generalizable rules to classify reviews at the document level. For the binary classification, a non-negative compound score refers to a positive prediction. For 3-class prediction, we consider a compound score greater than 0.05 as positive, a score less than −0.05 as negative, and a score between −0.05 and 0.05 as neutral. The ranges of the compound score for different classes are selected based on the original VADER paper. We use the PyPI VADER package.

TextBlob is a Python library for processing textual data. It provides an API for various NLP tasks such as part-of-speech tagging, noun phrase extraction, and sentiment analysis. TextBlob calculates both the polarity and subjectivity score from the text reviews utilizing opinion word’s weighted scores. At the binary-level, a non-negative polarity score refers to positive prediction. When 3-class is considered, a polarity score above 0.05 indicates the positive class, below −0.05 refers to the negative class, and remaining attributes to the neutral class.

OpinionFinder is a framework that (Wilson et al., 2005) can identify the subjectivity and polarity of a text. The OpinionFinder polarity classifier utilizes a prior word-level polarity score of positive, negative, or neutral to classify text. The class prediction of a review is determined following the same procedure of AFINN and SentiStrength.

4.1.2. Unsupervised/self-supervised methods

Besides, we provide comparative performances with some of the state-of-the-art unsupervised methodologies Fernández-Gavilanes et al. (2016), Vashishtha and Susan (2020), Viñares et al. (2017), Zhou et al. (2014) and self-supervised He and Zhou (2011) in a benchmark dataset (i.e., Cornell movie review). Few other existing self-supervised methodologies do not have their source code or datasets publicly available. We compare both LRSentiA and SSentiA utilizing limited labeled data. In setting 2, we show that SSentiA utilizing limited labeled data can yield similar performance to a fully labeled training dataset. This comparison is performed to exhibit the efficacy of SSentiA for minimizing the necessity of labeled data. Two supervised ML classifiers, Logistic Regression (LR) and Support Vector Machine (SVM) are used for the comparison.

5. Methodology

5.1. LR-SentiA: Proposed Lexical Rule-based Approach

LR-SentiA is a lexicon and rule-based method that can classify sentiment without using any labeled data. The main purpose of introducing LR-SentiA is to generate accurate pseudo-labels so that supervised ML classifiers can be incorporated into SSentiA. LR-SentiA utilizes a binary-level sentiment lexicon and set of rules to predict the polarity of a review. Besides predicting the class, LR-SentiA also provides the confidence score of the prediction. The steps of LR-SentiA are shown in Fig. 2.

5.1.1. Opinion words extraction

Not every word of a sentence constitutes an opinion. It is imperative to use only the subjective words which convey sentiment. Identifying and excluding objective text from reviews could significantly improve sentiment detection performance (Liu, 2012). In our work, we exclude words that do not convey any opinion or do not have any influence on the polarity of the sentence. We apply a dependency parser to extract grammatical structure and to identify the relationships among words. In addition to the POS tagger, a dependency parser can be used to eliminate words that do not affect sentence polarity. We use the spaCy library (Honnibal & Montani, 2017) to determine both the POS tag and the relationship among words in a sentence.

5.1.2. Sentiment lexicon

To determine the polarity of a review, a sentiment lexicon composed of lexicon units such as words and their sentiment orientations is required. The efficiency of a lexicon-based method largely depends on the coverage of opinion words in the lexicon. Some of the popular lexica for sentiment analysis are opinion lexicon (Hu & Liu, 2004), MPQA (Wilson, Wiebe, & Hoffmann, 2009), SentiWordNet (Baccianella et al., 2010), and SentiNet (Cambria et al., 2010). We utilize the opinion lexicon curated by Hu and Liu (Hu & Liu, 2004), which consists of 4783 negative and 2006 positive words with the polarity score of either −1 (negative) or +1 (positive). This lexicon contains the following POS: nouns, verbs, adjectives, and adverbs. It also includes misspellings, morphological variants, slang, and social-media mark-up.
5.1.5. Aggregation and document-level class assignment

2-class dataset: As most of the reviews are comprised of multiple sentences, it is necessary to combine the polarity scores obtained from individual sentences. In the aggregation step, we combine the polarity scores of each sentence to determine the overall sentiment polarity of a review. The overall polarity of a review is determined by the cumulative sum of the polarity scores of each sentence. If the review \( r \) consists of \( n \) sentences, \( s_1, s_2, \ldots, s_n \), and corresponding polarity scores are \( P(s_1), P(s_2), \ldots, P(s_n) \), then the overall polarity score is calculated as \( P(r) = \sum_{i=1}^{n} P(s_i) \). The semantic orientation of review \( r \) is determined by:

\[
\text{Sentiment of review}(r) = \begin{cases} 
\text{positive}, & \text{if } P(r) > 0 \\
\text{negative}, & \text{if } P(r) < 0 \\
\text{tie}, & \text{Otherwise}
\end{cases}
\]

If the overall polarity score of a review is greater than 0, it is considered as a positive review. If it is less than 0, then it is considered as a negative review. If the review score is 0, we consider it as a tie.

To determine the labels of tie cases, we utilize the cosine similarity score. To find the most similar reviews of review \( i \), we first compute its average similarity score \( \text{avgSim}(i) \) and standard deviation \( \text{std}(i) \) considering all the other reviews in the same dataset. Then the threshold similarity value is calculated by \( \text{simThreshold}(i) = \text{avgSim}(i) + a \times \text{std}(i) \), where \( \text{avgSim}(i) \) represents the average similarity score of review \( i \) and \( \text{std}(i) \) represents standard deviation. The value of \( a \) is set heuristically, we find \( a = 3 \) yields good results in all the datasets. We assign the class label of review \( i \) based on the similar reviews having a similarity score above the \( \text{simThreshold}(i) \). If there are \( m \) similar reviews, where \( j = 1, 2, \ldots, m \) and similarity score of between \( i \) and \( j \) is \( \text{sim}(i, j) \) and class label of \( j \) is \( c(j) \), then class score of \( i \) \( \text{classScore}(i) = \sum_{j=1}^{m} \text{sim}(i, j) \times c(j) \). If the \( \text{classScore}(i) \) is non-negative, then review \( i \) is predicted as positive, else it is assigned to the negative class.

3-class dataset: For 3-class datasets, we follow a very similar aggregation step of binary classification; only the class assignment procedure differs. If the polarity score of a review is positive with a confidence score (defined in the next section) above 0.10, we assign it to the positive class, while a negative polarity score with a confidence score above 0.10 is assigned to the negative class.

As stated above, in addition to the polarity score, the confidence score is utilized for a 3-class label assignment. A value of 0.10 is used as a threshold for the confidence score to distinguish between the positive or negative class and the neutral class. The threshold is selected based on the following assumptions — in the 3-class dataset, in a perfect case, reviews belong to the neutral class should have a polarity score of 0, a positive polarity score for the positive class, and a negative polarity score for reviews belong to negative class. However, this assumption is too stringent and not practical due to various reasons, such as lexicon coverage, binary polarity score (i.e., \(-1, +1\)) of opinion words, etc. Therefore, an additional constrain of confidence score (which actually refers to the polarity strength) of 0.10 is imposed to address the above-mentioned issue. We mark prediction as neutral when either \( P(r) \) is 0 or confidence score is less than 0.10.

5.1.6. Prediction confidence

In addition of predicting the class of a review, LRSentiA provides the confidence score of the prediction. We utilize the ratio of positive and negative polarity scores obtained from a review to determine the confidence score of the prediction. If the review \( r \) consists of \( n \) sentences, \( s_1, s_2, \ldots, s_n \) with positive polarity scores of \( P_{pos}(s_1), P_{pos}(s_2), \ldots, P_{pos}(s_n) \) and negative polarity scores of \( P_{neg}(s_1), P_{neg}(s_2), \ldots, P_{neg}(s_n) \), then overall positive polarity score of review \( r \) is calculated as \( P_{pos}(r) = \sum_{i=1}^{n} P_{pos}(s_i) \) and negative polarity score is...
calculated as \( P_{\text{neg}}(r) = \sum_{i=1}^{n} P_{\text{neg}}(t_i) \). The confidence score of the review \( r \) is determined by:

\[
\text{ConfScore}(r) = \frac{\text{abs}(P_{\text{pos}}(r) + P_{\text{neg}}(r))}{\text{abs}(P_{\text{pos}}(r) + \text{abs}(P_{\text{neg}}(r)))}
\]

5.2. SSentiA: Proposed self-supervised hybrid approach

The proposed self-supervised methodology, SSentiA, which is the main focus of this work, integrates supervised ML classifiers with the lexicon-based method LRSentiA. ML classifiers are capable of capturing the implicit pattern of data; therefore, they can enhance the classification performance on the complex and less polarized reviews that the lexicon-based methods like LRSentiA cannot distinguish well. However, the supervised ML classifier requires annotated reviews, which are not always available. Hence, we utilize LRSentiA to automatically generate pseudo-labels for the ML classifier of SSentiA.

SSentiA utilizes the confidence scores and predictions of LRSentiA to generate highly accurate pseudo-labels. By leveraging these pseudo-labels, a supervised ML algorithm is incorporated into SSentiA. We employ several ML classifiers: Logistic Regression (LR), Support Vector Machine (SVM), Random Forest (RF), and Extra Trees (ET) on reviews conveying weak or ambivalent opinion. We select the best performing ML classifiers and integrate them into SSentiA.

We utilize the scikit-learn (Pedregosa et al., 2011) implementation (ver.0.23.1) of the above-mentioned classifiers. For all the classifiers, the default parameter settings with the class-balanced weights are used. As an input feature for the ML classifiers, unigram, and bigram-based tf–idf (term frequency–inverse document frequency) scores are used. Fig. 3 shows various steps of SSentiA for sentiment classification.

5.2.1. Generation of pseudo-labels

We categorize the predictions of LRSentiA into multiple confidence groups based on their confidence scores ConfScore.

**2-class dataset:** In each dataset, we calculate the mean confidence score (mcs) and standard deviation (std) across all the predictions to find the threshold \( \text{thr} \) value. The \( \text{thr} \) value is used to determine various confidence groups, which is calculated as, \( \text{thr} = \text{mcs} + 0.5 \ast \text{std} \). If the \( \text{thr} \) value is less than 0.5, we use 0.5 as a \( \text{thr} \) value.

The confidence group of review \( r \), confGroup\((r) \) is determined as follows:

\[
\text{confGroup}(r) = \begin{cases} 
\text{very-high}, & \text{if ConfScore}(r) \geq \text{thr} \\
\text{high}, & \text{if ConfScore}(r) \geq \text{thr} - 0.5 \ast \text{std} \\
\text{low}, & \text{if ConfScore}(r) \geq \text{thr} - \text{std} \\
\text{very-low}, & \text{if ConfScore}(r) > 0 \\
\text{zero}, & \text{otherwise}
\end{cases}
\]

The predicted reviews with a confidence score above the \( \text{thr} \) fall into very-high confidence group. The next two categories (i.e., high and low) contain predictions whose confidence scores are 0.5 and 1.0 standard deviations (std) below the \( \text{thr} \) value, respectively. The very-low category contains predictions having pos confidence score and falls below low category.

Three criteria are considered while categorizing predictions into multiple groups that are described below:

a. Minimize the inclusion of wrong prediction (i.e., highly accurate pseudo-label) into few groups so that they can be used as training data for ML classifier with minimal error propagation.

b. Maximize the number of reviews (larger training set) utilized as pseudo-labels for ML classifier respecting criteria (a)

c. Show the correlation between confidence score and the accuracy (i.e., high confidence score implies high accuracy).

(a) and (b) both are important for having good performance from machine learning (ML) classifier, as (a) highly-accurate pseudo-label means less error-propagation to ML classifier and (b) higher number of pseudo-label means of the larger training set, that is needed to have good accuracy from machine learning model. (c) is important for group selection, (c) determines which groups should be used as training data and which ones to use as testing data.

We find discretizing the predictions of reviews into five categories fulfills the above criteria best; therefore, five groups are used.

**3-class dataset:** In 3-class sentiment analysis, predictions are grouped in a different way to incorporate the neutral class. The positive and negative predictions with a confidence score above 0.75 are placed to very-high confidence group, while confidence score above 0.5 are placed to high confidence groups. Predictions with a confidence score between 0.1 and 0.5 are considered low category predictions. The remaining reviews with confidence scores less than 0.1 are considered neutral class predictions.

The objective of distinguishing highly confident (i.e., very-high and high) predictions is to generate labels for supervised ML algorithms. When the rule-based classifier predicts the polarity of a review with high confidence (i.e., high positive/negative score), it is highly probable that the prediction is correct. As our rule-based method, LRSentiA, relies on the sentiment of individual opinion words, if the overall polarity score is very positive or very negative, then the review consists of mostly positive aspects (very positive score) or negative aspects (very negative score); thus the prediction is correct.

5.2.2. Utilizing pseudo-label

After identifying highly confident predictions (i.e., very-high and high confidence groups) of LRSentiA, we utilize them as pseudo-labeled training data for the supervised ML classifier.

**2-class dataset:**

As shown in Fig. 3, utilizing supervised ML algorithms and highly accurate pseudo-labeled training data (i.e., very-high and high confidence groups of LRSentiA), the low confidence prediction group is classified as either positive or negative. In the next phase, very-low and zero confidence groups are considered as testing data and predictions from very-high, high, low groups are employed as training data.

**3-class dataset:**

To train supervised ML classifiers for ternary classification, we use very-high and high confidence groups of positive and negative predictions as training data. As neutral class training data, we utilize predicted reviews having 0 polarity scores and contain at least 5 positive and negative terms.

As mentioned earlier, we want the pseudo-labels used in the training data as much accurate as possible. We observe that having a polarity score of 0 assigned by the lexicon-based method does not necessarily mean a review is neutral, as it could be due to the lexicon-coverage problem. For example, a review with 2 positive and 0 negative terms may yield a 0 polarity score if the positive terms do not exist in the lexicon and erroneously can be assigned to neutral class. Therefore, we impose an additional constraint to overcome the lexicon-coverage problem.

The minimum number of opinion terms required for a pseudo-labeled neutral class review is set to 5. The rationale behind that if the lexicon-based method identifies a high number (i.e., 5) of polarity terms in a review and still provides a total polarity score of 0 (due to the presence of both positive and negative terms), probably it is a neutral review. The remaining predictions from all the classes (positive, negative, and neutral) are considered as testing data for the supervised ML classifiers.

The remaining predictions from all the classes (positive, negative, and neutral) are considered as testing data for the supervised ML classifiers.
6. Results

To assess the performances of various classifiers, we calculate the accuracy, precision, recall, F1 score, and Matthews Correlation Coefficient (MCC) score. The accuracy score is biased towards the dominating classes. As most of our datasets are class-imbalanced, the MCC and F1 scores are better metrics to evaluate the performances of the classifiers.

Table 4 provides the mean confidence scores (mcs), standard deviations (std), and threshold (thr) scores of the predictions of LRSentiA in 4 binary datasets. We use these values to categorize predictions into various confidence groups.

Table 5 shows the accuracy and F1 score of four confidence categories of LRSentiA in various datasets. The results suggest a correlation between the confidence level of prediction and accuracy.

To show the correlation between the confidence score and accuracy of the predictions of LRSentiA, we conduct the Chi-squared test for independence, which indicates whether two categorical values are dependent. We consider four categories, very-high, high, low, and very-low. The null hypothesis assumes the confidences groups and their corresponding accuracy are independent, while the alternative hypothesis suggests they are not. The threshold value for the significant level is selected as 0.05. Using the Chi-squared test, we find the p-values < 0.05 for all datasets, which reject the null hypothesis and accept the alternative hypothesis.

Table 6 shows employing ML classifiers with pseudo-labels can enhance the accuracy of the low confidence groups of LRSentiA. In all the four datasets, we observe accuracy improvement between 25%–28% in low confidence groups.

Table 7 presents the comparative performance of proposed LRSentiA and SSentiA with the baseline sentiment analysis tools in four different datasets. In the IMDB dataset, LRSentiA provides an F1-score of 0.736, where VADER shows 0.69. The best F1-score is obtained using SSentiA with SVM, which is 0.807. Considering the accuracy, SSentiA with LR shows the highest accuracy of 80.63%, while LRSentiA and VADER show the accuracy of 73.62% and 67.6%, respectively. In the Amazon dataset, LRSentiA provides an F1 score of 0.719, while among the existing tools, VADER shows the highest F1 score of 0.724. The best F1 score is obtained using the proposed hybrid method using LR, which is 0.808. In the Clothing dataset, LRSentiA provides an F1 score of 0.665; Among the other existing lexicon-based tools, VADER shows the highest
Table 5

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Group</th>
<th>ConfScore</th>
<th>F1 Score</th>
<th>Accuracy</th>
<th>MCC</th>
<th>Acc. (%)</th>
<th>#Correct/Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>TripAdvisor</td>
<td>very-high</td>
<td>0.727 - 1.0</td>
<td>0.922</td>
<td>0.844</td>
<td>95.53</td>
<td>6391/6706</td>
<td></td>
</tr>
<tr>
<td></td>
<td>high</td>
<td>0.556 - 0.727</td>
<td>0.893</td>
<td>0.786</td>
<td>90.26</td>
<td>2345/2598</td>
<td></td>
</tr>
<tr>
<td></td>
<td>very-low</td>
<td>0.384 - 0.556</td>
<td>0.831</td>
<td>0.662</td>
<td>80.29</td>
<td>2144/2679</td>
<td></td>
</tr>
<tr>
<td>IMDB</td>
<td>low</td>
<td>0.500 - 1.0</td>
<td>0.757</td>
<td>88.04</td>
<td>6290/7144</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>high</td>
<td>0.367 - 0.500</td>
<td>0.826</td>
<td>0.653</td>
<td>82.71</td>
<td>2460/2974</td>
<td></td>
</tr>
<tr>
<td></td>
<td>very-low</td>
<td>(-0) - 0.384</td>
<td>0.679</td>
<td>0.353</td>
<td>61.67</td>
<td>3498/5672</td>
<td></td>
</tr>
</tbody>
</table>

Table 6

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Group</th>
<th>Method</th>
<th>F1 Score</th>
<th>Accuracy</th>
<th>Improvement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amazon</td>
<td>low</td>
<td>Lexicon OpinionFinder</td>
<td>0.568</td>
<td>0.557</td>
<td>0.561</td>
</tr>
<tr>
<td></td>
<td>high</td>
<td>TextBlob</td>
<td>0.771</td>
<td>0.622</td>
<td>0.712</td>
</tr>
<tr>
<td></td>
<td>SentiStrength</td>
<td>0.691</td>
<td>0.653</td>
<td>0.672</td>
<td>0.327</td>
</tr>
<tr>
<td></td>
<td>AFINN</td>
<td>0.723</td>
<td>0.696</td>
<td>0.707</td>
<td>0.417</td>
</tr>
<tr>
<td></td>
<td>VADER</td>
<td>0.735</td>
<td>0.695</td>
<td>0.723</td>
<td>0.445</td>
</tr>
</tbody>
</table>

F1-score of 0.666. The best F1-score is obtained using the SSentiA and LR, which is 0.774.

We conduct the McNemar test (McNemar, 1947) to determine whether there exists a significant difference between the performances of the lexicon-based methods and SSentiA. The McNemar test determines whether two experimental results disagree with each other by comparing their sensitivity and specificity on the same data. The null hypothesis claims that the two results are the same, while the alternative hypothesis indicates the opposite.

For each dataset, the best performing lexicon-based method (prediction-1) is compared with SSentiA (prediction-2) using a 2 X 2 contingency table. A value of 0.05 is used as a level of significance. For all the four binary datasets, we find p-values less than 0.005, which infer significant differences between the performance of the best lexicon-based method and SSentiA.

Table 8 shows the comparison between SSentiA and several unsupervised and self-supervised methodologies in the Cornell movie review dataset. The results indicate that SSentiA performs better than all of them by some margin. When SVM is integrated into SSentiA, it performs best with an accuracy of 77.3%. All the other methods except Vashishtha and Susan (2020) show accuracy between 73%-75%.

In the 3-class Amazon dataset, as shown in Table 9, among the lexicon-based methods, LRSentia obtains the highest F1 score of 0.495; SentiStrength achieves the highest F1 score in the 3-class TripAdvisordataset, while VADER provides the highest accuracy for the Amazon dataset. When SSentiA is employed, both the accuracy and F1 score are improved significantly over the lexicon-based methods. In the TripAdvisordataset, SSentiA utilizing SVM achieves an F1-score of 0.740, which is approximately 32% enhancement over the best-performing lexicon-based classifier. In terms of accuracy, an improvement of 30% is observed, from 61.27% to 79.98%. In the Amazon dataset, SSentiA escalates the F1 score to 0.587 from 0.495 (LRSentia), an improvement of over 20%.
6.1. **SSentiA as a semi-supervised tool (using limited labeled data)**

We further show that SSentiA can also be utilized as a semi-supervised tool. We demonstrate that it yields similar accuracy to a fully labeled (FL) training dataset utilizing only a portion of manually labeled data. The comparison is shown with the fully labeled (FL) dataset that utilizes two supervised ML classifiers, LR and SVM.

SSentiA leverages predictions from high and very-high confidence groups of LRSSentiA as pseudo-labeled data (i.e., no manual labels are required for reviews belonging to these groups). Only for the reviews that belong to low, very-low, zero confidences groups of LRSSentiA, manually annotated data are used, therefore, we refer to it as SSentiA-PL (partially labeled). In a fully Labeled (FL) dataset, all labels are manually annotated.

For both the supervised ML classifiers, default parameter settings are used. As an input feature, unigram and bigram-based term frequency—inverse document frequency (tf-idf) scores are used. We perform 5-fold cross-validation (90% train data and 20% testing data) and compute averaged F1 score and accuracy. The assessment of the F1 score and accuracy of the 20% test split is performed considering the true label, not the pseudo-label.

As shown in **Table 4**, SSentiA-PL (partially labeled) reaches similar accuracy to the fully labeled (FL) dataset for binary classification using only 30%–60% of the labeled data. In the Clothing dataset, we find only 32% of labeled data are required for SSentiA-PL to attain a similar accuracy and F1 score of the fully labeled (FL) dataset. For the Amazon, TripAdvisor, and IMDb datasets, SSentiA-PL requires around 41.06%, 51.11%, and 59.60% of labeled data, respectively.

### Table 8
The performance comparison of SSentiA with existing lexicon-based methods in Cornell movie review dataset.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Method</th>
<th>P</th>
<th>R</th>
<th>F1 Score</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Movie</td>
<td>Dependency Parsing</td>
<td>0.749</td>
<td>0.748</td>
<td>0.748</td>
<td>74.80%</td>
</tr>
<tr>
<td>Movie</td>
<td>SentiStrength</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>71.10%</td>
</tr>
<tr>
<td>Movie</td>
<td>TextBlob</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>71.24%</td>
</tr>
<tr>
<td>Movie</td>
<td>AFINN</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>46.47%</td>
</tr>
<tr>
<td>Movie</td>
<td>VADER</td>
<td>0.539</td>
<td>0.502</td>
<td>0.520</td>
<td>60.09%</td>
</tr>
<tr>
<td>TripAdv</td>
<td>LRSSentiA</td>
<td>0.563</td>
<td>0.537</td>
<td>0.551</td>
<td>61.76%</td>
</tr>
<tr>
<td>TripAdv</td>
<td>SentiStrength</td>
<td>0.569</td>
<td>0.552</td>
<td>0.561</td>
<td>60.66%</td>
</tr>
<tr>
<td>Hybrid</td>
<td>SSentiA (LR)</td>
<td>0.717</td>
<td>0.705</td>
<td>0.711</td>
<td>76.75%</td>
</tr>
<tr>
<td>Hybrid</td>
<td>SSentiA (SVM)</td>
<td>0.742</td>
<td>0.738</td>
<td>0.740</td>
<td>79.98%</td>
</tr>
</tbody>
</table>

### Table 9
The performance comparison of SSentiA with existing lexicon-based methods in 3-class datasets.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Model</th>
<th>Precision</th>
<th>Recall</th>
<th>F1 Score</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lexicon</td>
<td>OpinionFinder</td>
<td>0.437</td>
<td>0.414</td>
<td>0.425</td>
<td>41.71%</td>
</tr>
<tr>
<td>Lexicon</td>
<td>TextBlob</td>
<td>0.540</td>
<td>0.468</td>
<td>0.501</td>
<td>53.45%</td>
</tr>
<tr>
<td>Lexicon</td>
<td>AFINN</td>
<td>0.539</td>
<td>0.497</td>
<td>0.518</td>
<td>59.02%</td>
</tr>
<tr>
<td>Lexicon</td>
<td>VADER</td>
<td>0.539</td>
<td>0.502</td>
<td>0.520</td>
<td>60.09%</td>
</tr>
<tr>
<td>TripAdvisor</td>
<td>LRSSentiA</td>
<td>0.563</td>
<td>0.537</td>
<td>0.551</td>
<td>61.76%</td>
</tr>
<tr>
<td>TripAdvisor</td>
<td>SentiStrength</td>
<td>0.569</td>
<td>0.552</td>
<td>0.561</td>
<td>60.66%</td>
</tr>
<tr>
<td>Hybrid</td>
<td>SSentiA (LR)</td>
<td>0.717</td>
<td>0.705</td>
<td>0.711</td>
<td>76.75%</td>
</tr>
<tr>
<td>Hybrid</td>
<td>SSentiA (SVM)</td>
<td>0.742</td>
<td>0.738</td>
<td>0.740</td>
<td>79.98%</td>
</tr>
</tbody>
</table>

### Table 10
The performance of SSentiA using partial labeled dataset.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Classifier</th>
<th>F1 Score</th>
<th>Accuracy (%)</th>
<th>(% of Annotated data)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clothing</td>
<td>LR</td>
<td>0.828/0.819</td>
<td>88.4/88.6</td>
<td>100%/32.32%</td>
</tr>
<tr>
<td>Amazon</td>
<td>SVM</td>
<td>0.817/0.797</td>
<td>87.54/87.34</td>
<td></td>
</tr>
<tr>
<td>TripAdvisor</td>
<td>LR</td>
<td>0.885/0.878</td>
<td>91.50/90.90</td>
<td>100%/41.06%</td>
</tr>
<tr>
<td>IMDb</td>
<td>SVM</td>
<td>0.886/0.876</td>
<td>91.77/91.14</td>
<td>100%/51.11%</td>
</tr>
</tbody>
</table>

7. **Discussion**

The experimental results reveal that leveraging binary-level sentiment lexicon (i.e., +1, -1), LRSSentiA, performs similarly or better than other lexicon-based tools that utilize intensity-based polarity lexicon. In all the four binary datasets, LRSSentiA outperforms SentiStrength and TextBlob and yields similar performance to VADER. Among the two 3-class datasets, LRSSentiA achieves the highest F1 score in TripAdvisor and highest accuracy in the Amazon dataset.

From **Table 5**, it is evident that LRSSentiA classifies fairly accurately when the review polarity is easily distinguishable (i.e., high confidence score) as implied by the higher accuracy and F1 score in very-high and high confidence groups. The Chi-squared test indicates that there exists a correlation between the prediction confidence score and accuracy. Therefore, if the user review is comprised of mixed opinions, relying only on the explicit rules or polarity lexicon is often ineffective. Also, assigning weights to individual opinion words does not solve this issue, as seen by the inferior performance of SentiStrength and TextBlob in **Table 7**.

As **Table 5** indicates, LRSSentiA can generate highly accurate pseudo-labels (very-high and high confidence groups) when labeled data are unavailable. The resultant pseudo-labels can be utilized to incorporate supervised ML algorithms to improve the performance of sentiment classification for the hard-to-distinguish reviews. **Tables 7** and **9** suggest that integrating supervised ML algorithms can be highly effective as SSentiA outperforms baseline methods consistently and significantly in binary as well as 3-class sentiment analysis.

**Fig. 4** shows examples of reviews which lexicon-based methods, such as LRSSentiA, VADER, AFINN, and TextBlob misclassify but SSentiA by incorporating ML classifier predicts correctly. For example, the review: "This skirt looks exactly as pictured and fits great. I purchased it a few weeks ago and got lots of compliments on it. However, on the third wear, the side zipper split wide open, needless to say, it was returned." comprised of positive and negative words. Therefore, all the lexicon-based methods predict it as positive.

The following positive review, "This player is great. Forget about the iPod Shuffle. This has a much longer battery life, an fm tuner, and an incredible OLED screen. A previous reviewer said that it does not support mp3. Well they are wrong. I only use mp3s on it. The only down side is the terrible Songlist software that you use with it. This is the worst software I have ever used. Thankfully there is an alternative. Get MP3 File Manager. It is drag and drop and works great." contains both positive and negative words, but all the lexicon-based methods predict it as negative. Reliance
on word-level polarity without considering the contextual meaning, as well as dependence on lexicon coverage, makes the lexical rule-based methods ineffective for complex cases.

7.1. Findings and implications

- We find that the lexicon-based systems yield lower accuracy compared to ML methods, particularly for the reviews with mixed sentiments. Hence, it is essential to incorporate the ML method to enhance the performance of sentiment classification.
- We notice a correlation between the presence of sentiment terms/phrases in a review and the correctness of the prediction using a lexicon-based method. If a review is strongly positive or negative, lexicon-based methods are quite effective to classify them correctly. Therefore, lexicon-based methods are most suitable for classifying highly polarized reviews.
- We observe that utilizing a binary polarity lexicon, LRSentiA exhibits the better or similar performance of Sentistrength, VADER, TextBlob, and AFINN, which suggests that word-level polarity strength does not influence the performance of a lexicon-based method much. Thus, to improve the accuracy of the lexicon-based method, it is necessary to focus on other aspects.
- We find that it is important to select pseudo-labeled training data carefully to minimize error propagation. Therefore, it is crucial to improve the accuracy of lexicon-based methods so that highly accurate pseudo-labels can be generated.
- Finally, we learn that coupling the ML classifier with the lexicon-based method is essential to building an effective sentiment classifier for the unlabeled dataset.

8. Conclusion

In this work, we study both binary and multi-class sentiment classification problems from the unlabeled data using a hybrid approach. As data labeling often requires domain expertise and manual-effort, our study focuses on identifying sentiments from unlabeled data. We show that the lexicon-based methods, including LRSentiA, are not robust enough to deal with the ambiguity and variations of the natural language as they rely on word-level polarity and simple linguistic rules. Therefore, we integrate the ML algorithm to improve the performance of complicated subsets. Leveraging highly accurate pseudo-labels generated from LRSentiA, our proposed hybrid method SSentiA employs a supervised ML classifier to predict the sentiments of complex reviews. Combining the best of both approaches, interpretability of the rule-based approach, and implicit pattern learning capabilities of ML algorithms, SSentiA significantly enhances the performance of sentiment classification. Besides, we show that SSentiA utilizing around 30%-60% labeled data can achieve similar performance to a fully labeled dataset.

CRediT authorship contribution statement

Salim Sazzed: Designed the methodology, Wrote the paper, Implemented the methodology. Sampath Jayarathna: Designed the methodology, Wrote the paper.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data and code availability


References


