An Improved SVM Noise Tolerance for Implicit Aspect Identification in Sentiment Analysis

Halima Benarafa [1,*], Mohammed Benkhalifa [1], and Moulay Akhloufi [2]

[1] Department of Computer Science, Algorithms, Networks, Intelligent Systems and Software Engineering (ANISSE), Faculty of Sciences, Mohammed V University in Rabat, Morocco

[2] Department of Computer Science, Perception, Robotics, and Intelligent Machines (PRIME), University of Moncton, Moncton, New Brunswick, Canada

Email: halima_benarafa@um5.ac.ma (H.B.); m.benkhalifa@um5r.ac.ma (M.B.); moulay.akhloufi@umoncton.ca (M.A.)

*Corresponding author

***Abstract*—Opinion mining or Sentiment Analysis (SA) is an essential component of e-commerce applications where consumers generate a large number of reviews. Opinions conveyed about a particular feature of a product have a significant impact on consumer decisions and companies' reputations. Aspect-based Sentiment Analysis (ABSA), is the process of classifying text according to different aspects and identifying the sentiment associated with each category. In this article, a method is suggested for enhancing the Support Vector Machines (SVM) model to improve its noise tolerance when dealing with the Implicit Aspect Identification (IAI) task which is a subtask of Aspect Based Sentiment Analysis. Using WordNet (WN) semantic relations, modification to the SVM kernel computation is proposed. This study evaluates SVM noise robustness using its classification performance with noisy datasets and multiple kernels. Experiments are conducted on three benchmark datasets of laptops, restaurants, and product reviews. Results are evaluated and analyzed based on the impact of the proposed approach on the performance of SVM for two types of noise (class noise and attribute noise) and two types of kernels (linear kernel and nonlinear kernels). According to the empirical results, the suggested method is shown to increase the noise tolerance of SVM for IAI.**

***Keywords*—implicit aspect-based sentiment analysis, support vector machines, wordnet, Lesk algorithm, equalized loss of accuracy, noise robustness, label noise, class noise**

I. Introduction

In recent years, Sentiment Analysis (SA) has grown in importance as a task in Natural Language Processing (NLP), particularly for data from online social media platforms, blogs, forums, and microblogs. The goal of SA is to determine the aspect, sentence, or document's sentiment rating or sentiment polarity [1]. SA is regarded as the crucial technology for achieving powerful artificial intelligence and creating machines that can fully comprehend human languages. Sentiment Analysis (SA) is also useful in a variety of application contexts, including financial, political, e-health, e-tourism, user profiles, user influence, community detection, and dialogue systems.

Aspect Sentiment Analysis is a fine-grained form of Sentiment Analysis and a subtask of NLP [2] that involves analyzing text to identify the aspects of a given entity and the sentiment expressed towards each aspect. Aspect-based Sentiment Analysis (ABSA) can be divided into two subtasks: Aspect Identification (AI) and Aspect Sentiment Classification (ASC). There are two types of aspects: explicit aspects and implicit aspects. Explicit aspects are directly mentioned in the text and can be easily identified. They are usually nouns or noun phrases that refer to specific features or attributes of the product, service, or entity. However, implicit aspects are not directly mentioned in the text and are usually inferred from the context. They can be adjectives, verbs, or other parts of speech that describe the product, service, or entity.

The existing Aspect-based Sentiment Analysis (ABSA) techniques have several limitations that need to be addressed, namely performance limitations, imbalanced data, lack of large annotated Datasets, and noisy data. Current ABSA mechanisms may not perform well on challenging tasks and robustness issues, which could restrict research on ABSA models. Unbalanced data [3], where sentiment labels are unevenly distributed, can lead to biased models that favor over-represented classes, lowering accuracy and performance. Moreover, the absence of large-scale and high-quality ABSA datasets restricts the development of accurate ABSA models, as they rely on well-annotated data for evaluation and training. Small-scale ABSA datasets may not adequately represent the complexity and diversity of real-world data. Finally, and more importantly, noisy datasets represent a major weakness for ABSA systems as they negatively affect their performances and make them face challenges in finding relevant aspects and accurately categorizing aspects and sentiments [4]. Therefore, handling noisy datasets is crucial for reliable and precise sentiment analysis, especially for machine learning techniques that are fundamentally very sensitive to noise like SVM classifier.

Manuscript received December 22, 2023; revised February 2, 2024; accepted February 18, 2024; published July 18, 2024.

In this work, the topic of interest is AI which refers to the process of automatically identifying the specific aspects or features of a given entity that are being discussed in a text. This process is a crucial part of ABSA [5], which involves analyzing text to identify the aspects of a given entity and the sentiment expressed towards each aspect. The AI task can be divided into two subtasks: Aspect Term Extraction (ATE) and Aspect Category Detection (ACD). ATE involves identifying the specific aspect terms, which are the words or phrases that refer to the aspects of an entity. However, ACD which is the main focus of this paper, looks to detect the aspect categories, which are broader groups of aspect terms that share similar meanings.

AI is a challenging task due to the complexity of natural language. Various AI techniques have been proposed to address this task, including rule-based methods, machine learning, and deep learning models. These techniques have shown promising results in identifying aspects and categories from various domains, such as product reviews, social media, and news articles.

With machine learning, AI is treated as a classification task that is a supervised model that learns from the training data to classify new data instances [6]. Since training data quality can severely suffer from noise, AI performance may be significantly impacted. A thorough analysis of the impact of noise in machine learning algorithms for ABSA categorizes noise into two groups: attribute noise and class noise. Noise in the input data points is known as 'attribute noise', and noise in the labels is known as 'class noise'. Attribute noise refers to corruption in the values of one or more attributes. Examples of attribute noise are erroneous attribute values, conflicting or contradictory data, and errors in data entry. Class noise refers to corruptions in the labels of one or more samples. Class noise has several sources, including subjectivity during the labeling process, data entry errors, or insufficient information used for labeling each sample.

Being one of the most popular learning classifiers, SVM has received increased interest and has been used in considerable studies addressing ABSA with its two subtasks [7–12]. Due to its sensitivity to noise, SVM may not appropriately function with noisy datasets. Noise could appear in the labels or the input data points during SVM classification. In this research, SVM tolerance to noise (class and attribute) is investigated using Linear, Gaussian, ANOVA, and Bessel kernels.

In this paper, an approach is proposed for enhancing the SVM algorithm by merging its fundamental kernel functions with similarity functions derived from the Lesk algorithm [13], to improve its noise tolerance when dealing with Implicit Aspect Identification (IAI). Lesk algorithm is applied to Word Sense Disambiguation (WSD) [14]. According to [15–17], WSD is the process of automatically giving ambiguous words meaning in a particular context. A word's acceptable meaning in a given context has the greatest amount of overlap between the given context and its definition.

In this work, the principle of the Lesk Algorithm for WSD is applied. However, the work's originality is demonstrated on two different levels:

- The theoretical foundation behind the idea: To create a Lesk algorithm-based similarity function between terms WordNet dictionary (WN), created in [18], is needed. Using this function, a novel SVM kernel is developed to assign semantically similar words greater weights in terms of their level of impact on categorizing new observations.
- Model creation: The proposed similarity function amplifies the similarity score between terms. For terms with similar semantic features, this new formulation provides a noticeably higher similarity rating. However, it keeps the same fundamental kernel value for terms with different meanings.

The structure of the paper is as follows. Section II highlights related works on ABSA. Section III describes our suggested method. Section IV provides the experimental setup followed by a section on the findings and discussion. Section V presents the conclusion.

II. Related Works

ABSA is a fine-grained sentiment analysis problem that seeks to evaluate and comprehend people's opinions at the aspect level. ABSA has attracted a lot of interest over the last decade due to the need for recognizing more fine-grained aspect-level opinions and sentiments. It involves analyzing different sentiment elements and their relations, including the aspect term, aspect category, opinion term, and sentiment polarity [19]. Numerous techniques are presented supporting and evaluating various sentiment components and their relationships to handle ABSA in various scenarios. However, ABSA faces several challenges such as noisy and ambiguous aspect terms, which affect the accuracy of ABSA models. In this section, we present related works in ABSA for IAI some of which focus on ABSA performance enhancement, while others on noise tolerance improvement for ABSA tasks. To provide a clear landscape of existing literature, related works are organized into two major groups. The first group concerns those focusing on ABSA performance enhancement without treating noise, and the second contains works dealing with noise tolerance improvement for ABSA models. Works in each group are classified according to two aspects of their approaches (i) the operating level and (ii) the use of semantic information. These latter are two key criteria that help better situate this present paper within existing related literature. In this section, the first group of studies is introduced then followed by the second one.

Several studies in the first group have suggested methods to improve ABSA tasks without coping with noise.

Benarafa *et al*. [20, 21] operated on the model level to bring enhancements to their underlying models using semantic information to boost ABSA tasks.

To deal with the IAI task in sentiment analysis, Benarafa *et al*. [20] proposed in suggests a method to enhance the K-Nearest Neighbors (KNN) algorithm. To support the IAI task, the proposed method enhances KNN distance computation using WordNet semantic relations.

Benarafa *et al.* [21] suggest a technique to enhance SVM algorithm kernel computation using WordNet semantic relations for IAI tasks. Pathak *et al.* [22], Datta and Chakrabarti [23], and Tubishat *et al.* [24] operated on model levels to improve their ABSA tasks without semantic calculations. In Ref. [22], A topic-level model for SA based on the LSTM network is suggested to conduct ACD and sentiment categorization. Datta and Chakrabarti [23] improved the Recurrent Neural Network (RNN) for ASC on demonetization tweets by creating a novel method called Fire Fly-oriented Multi-Verse Optimizer (FF-MVO) that optimizes polarity measurements used by RNN to categorize the extracted features. Tubishat *et al.* [24] suggested an approach to perform explicit ATE in SA using optimal rules combination. They combine 126 aspect extraction rules to extract explicit aspects from the input text and propose a novel technique for combining the aspect extraction rules to optimize the technique's performance. Soni and Rambola [25], Sivakumar and Reddy [26], and Khan *et al.* [27] operated on data level via data augmentation and used semantic calculations through similarities in their underlying models.

In a recent paper, a hybrid method, combining RNN with a spaCy-based similarity function and WordNet-based similarity metrics, is presented for identifying implicit aspects [25]. Sivakumar and Reddy [26] exploited the semantic similarity between the aspect term and the opinion sentence to enhance several machine-learning techniques for the sentiment classification task. Khan *et al.* [27] suggested a method for performing lifelong aspect extraction from big data via knowledge engineering. It extracts features from big data using rule-based and machine-learning-based methods. Different works, in the second group, have proposed methods to improve noise tolerance for ABSA tasks.

On the one hand, Ref. [28] is the only work that operates on model level and does not use any semantic information. Its authors put forward a novel neural network framework, namely the Gated Alternate Neural Network (GANN), for ABSA. Using Gate Truncation Layer (GTR) to handle noisy input data and learn useful sentiment clue representations, the proposed framework is designed to enhance the capability of ABSA models by incorporating gating mechanisms and alternate connections between the input and output layers. On the other hand, six other works [29–34] function on the data level without utilizing semantic computations.

Fei *et al.* [29] suggested enhancing ABSA robustness through the simultaneous integration of rich external syntactic dependencies and aspect labels with a universal-syntax graph convolutional network, inducing high-quality synthetic training data with various types to increase the diversity of the training set and therefore improving the model's ability to handle noise. Chen *et al.* [30] proposed a noise-aware BERT re-ranking algorithm to properly filter out noisy data. Their approach is founded on a novel loss function that can effectively handle textual noise in the input data and hence increase the robustness of BERT re-ranking. A three-step semi-supervised hybrid technique is put forth by Kumar *et al.* [31] for ABSA tasks in consumer reviews. The authors propose a noise-resistant loss function applied to the three phases (pre-processing, aspect prediction, and sentiment prediction) of a method called CASC performance is enhanced by lessening the effect of label noise in the training dataset. Wang *et al.* [32] introduced a Contrastive Cross-Channel Data Augmentation (CCDA), which incorporates contrastive learning into the data augmentation process. This approach aids in training the model to be more resilient to noisy data. Shi *et al.* [33] proposed a hybrid model for ABSA, integrating Local Position-POS Awareness (LPPA) and Global Dense Connection (GDC), to fuse the dependency features between the aspect terms and associated opinion words to obtain the final sentiment classification decision. Even if the study does not explicitly focus on noise tolerance, the proposed hybrid model integrates several features to enhance the robustness of the models to input variations, including noise in the dataset. Finally, Li *et al.* [34] introduced data augmentation strategies for ABSA to enhance the classification performance. The researchers investigate the effect of data augmentation on a hybrid approach for ABSA. They focus on techniques such as Easy Data Augmentation (EDA), back translation, and word mixup. While the noise tolerance of the models may vary depending on the specific implementation and dataset, data augmentation techniques (used in this study) help improve the noise tolerance of ABSA models by introducing additional variations and generating more diverse training samples.

Previously presented related works cope with noisy data for ABSA as it is done in this paper. However, it is worth mentioning that this article differs from related works in many regards:

- Approach principle: this work operates on the SVM model level by enhancing underlying kernel functions using semantic knowledge. Many related works (from the second group) operate on the data level by data augmentation, and no one of those operating on the model level brings changes to its underlying core model and uses semantic information for its approach. In this research, the SVM classification model is improved using semantic relations from WN so that it can better handle the classification of noisy datasets. To fulfill this purpose, a technique is proposed to suitably capture similarity information between two aspect terms and leverage this similarity to boost the degree of influence that these two aspect terms have on each other's classification. This method modifies the SVM kernel which is a fundamental component of the SVM model as it manages the degree of influence on classification between two aspect terms and therefore controls how each term in the training data affects the final SVM's classification results.
- Empirical validation: this work uses different experimental validation setups. Firstly, it uses two commonly used noise generation techniques (class and attribute noise) with different noise levels while

other works employ either context perturbation or artificially created noisy training data without using different noise quantities. Secondly, it uses SVM as the learning text classifier which is very popular in the sentiment analysis area and well known for its high sensitivity to noisy data, while other works use mainly deep learning models. Thirdly, besides the F1-score measure, it uses an Equalized Loss of Accuracy (ELA) metric that measures how noise robust a classifier is whereas other works mainly use accuracy and F1–Score measures.

- This work does not aim to do better than existing research nor complement it, but it intends to show how existing knowledge is exploited to provide additional understandings and insights into research problems in the large study space offered by the ABSA area. Even if it shares the same global goal with other works, it mainly seeks to propose a novel approach based on external semantic knowledge to change the SVM core model so that it becomes more performant and particularly more noise tolerant than the baseline SVM model.

III. Proposed Approach

Before presenting the proposed approach, the background notions are first introduced and their purpose is explained in this section.

SVM Classifier: Support Vector Machines (SVM) are a class of supervised learning methods for regression and classification [35]. Focusing on the classification task, SVM aims to build a hyperplane that separates examples into different classes while maximizing the margin with the nearest data points (i.e., support vectors). The following function must be minimized as part of SVM training:

$$\min_{w} \frac{1}{2} w^2 + C \sum_1^N \xi_i \quad (1)$$

where $y_i\,(w^T \phi(A_i) + b) \geq 1 - \xi_i$ and $\xi_i \geq 0$

With C being the error penalty, w the normal vector to the hyperplane, N the number of training cases, b a constant, and ξ_i is the slack variable measuring the degree of misclassification of aspect terms A_i. ϕ, also called K, represents the mapping function that transforms data from the input into the feature space. For simpler computations, the following equation represents how the minimization of Eq. (1) corresponds to the maximization of its dual problem through the use of Lagrange multipliers α_i:

$$\max_{\alpha} \sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i\, y_j K(A_i A_j) \quad (2)$$

where $\sum_i y_i \alpha_i = 0$ and $0 \leq \alpha_i \leq C$

K is defined as a mathematical trick that transforms data from a low-dimensional to a higher-dimensional space [36, 37]. When SVM cannot calculate its hyperplane for nonlinear classification problems, it uses the kernel that helps to transform the training data so that a non-linear decision surface can transform into a linear equation in a higher dimension. Thus, SVM succeeds in forming its hyperplane in the higher dimension and therefore creates its decision boundary.

Four kernels are employed in the proposed method:

Linear kernel [38]: Often referred to as “Non-kernel”, the Linear kernel is the simplest and most basic kind of kernel. When there are many features, it is considered to be the best kernel. For binary text classification tasks, the linear kernel is often chosen because the majority of these tasks are linearly separable. The equation of Linear kernel is presented as follows:

$$k(x, y) = x \cdot y \quad (3)$$

where x and y are two data points.

Gaussian RBF kernel [38]: One of the most popular kernels used by SVM is the Gaussian RBF kernel. This kernel is preferred in the case where no prior information on the data is available. The equation of Gaussian RBF is formulated as:

$$K(x, y) = exp(-\gamma \parallel x - y \parallel^2) \quad (4)$$

where $\parallel x - y \parallel^2$ is the Euclidean distance between the two data points, x and y. The shape of the Gaussian curve is controlled by the parameter γ , which also governs how each training sample influences the classification output.

Anova kernel [38]: Kernel-based algorithms like SVM commonly employ the ANOVA kernel [38], which is a radial basis function. The equation of the ANOVA kernel is given by:

$$K(x, y) = \sum_{k=1}^{n} e\,xp(-\sigma(x_k - y_k)^2)^d \quad (5)$$

where d stands for the degree of the ANOVA kernel and x and y are two data points. The classification problem’s border and the ANOVA kernel’s shape are both influenced by the parameter σ.

Bessel kernel [38]: The Bessel kernel is a radial basis function [38]. The radial basis function kernel, or RBF kernel, is a commonly used kernel function in machine learning that is employed in many kernelized learning techniques, especially in SVM classification. The equation of the Bessel kernel is presented as follows:

$$K(x, y) = J_0(\sigma \parallel x - y \parallel) \quad (6)$$

where $\parallel x - y \parallel$ is the Euclidean distance between the two data points x and y, and J_0 is the Bessel function of the first kind.

After introducing the key background notions of the approach, this latter will be outlined in this section. The main purpose of the proposed method, which is defined in Fig. 1, is to incorporate pertinent external knowledge, specifically semantic information from the WN lexical database, into the SVM Kernel calculation to improve its noise tolerance when dealing with IAI tasks. To attain this goal, a WN semantic information-based similarity for four SVM kernel functions is suggested.

Let Ai and Aj be two Implicit Aspect Terms (IAT) in the dataset, Def_{Ai} and Def_{Aj} are their corresponding WN definition sets. Following is a definition for Def_{Ai} and Def_{Aj}:

$$Def_{Ai} = \{subdef_{Ai1}, \ldots, subdef_{Ais}, \ldots, subdef_{AiN}\}, s \in [1, N] \quad (7)$$

$$Def_{Aj} = \{subdef_{Aj1}, \ldots, subdef_{Ajt}, \ldots, subdef_{AjM}\}, t \in [1, M] \quad (8)$$

where *N* and *M* are the number of definitions in *DefAi* and *DefAj*, respectively. $subdef_{Ais}$ and $subdef_{Ajt}$ are sets of words that correspond to the s^{th} and the definitions, respectively, in Def_{Ai} and Def_{Aj}.

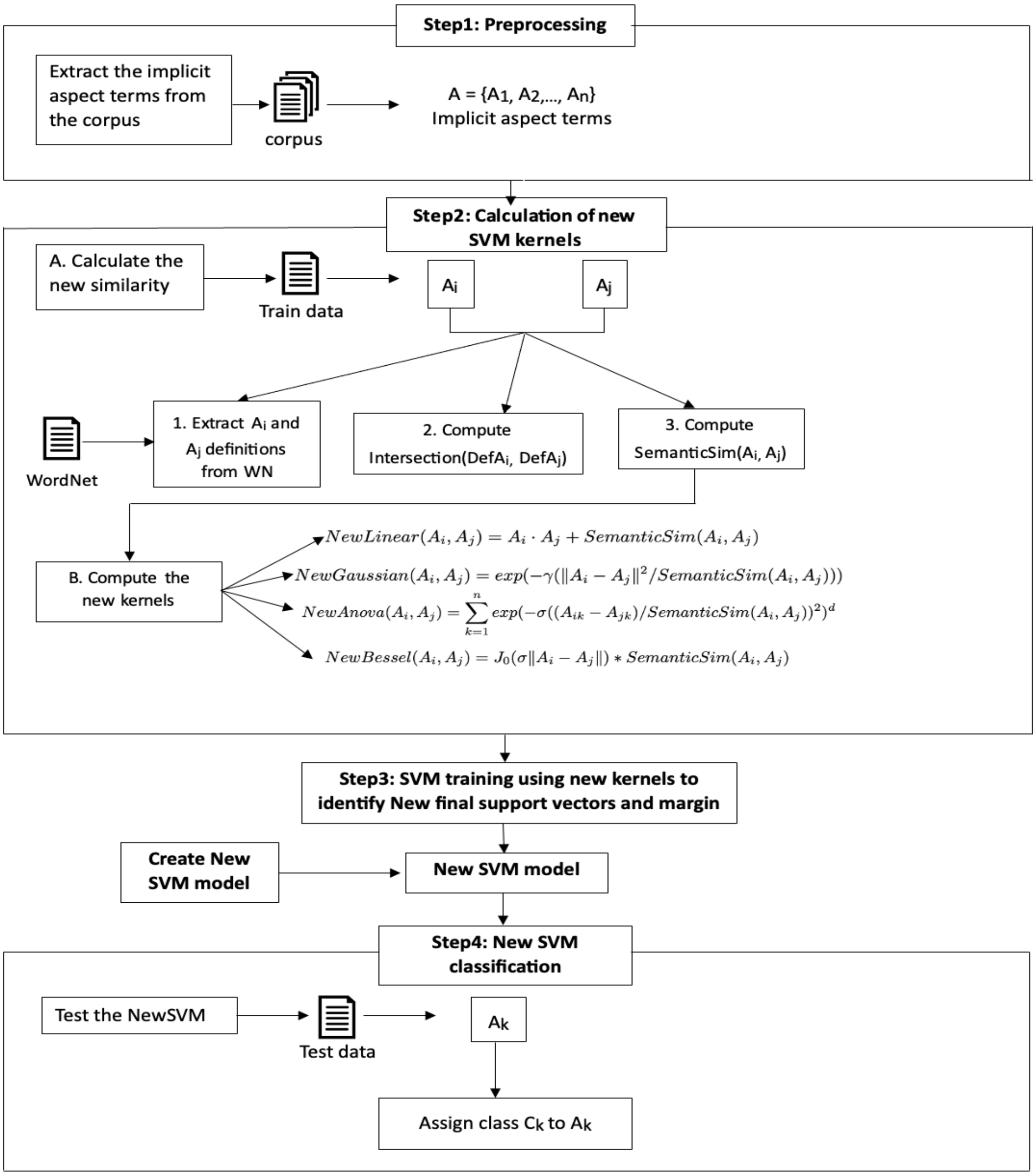


Fig. 1. Summary of our approach.

For calculating the new SVM kernels, the new semantic similarity between Aspects *Ai* and *Aj* is defined as follows:

$$SemanticSim(A_i, A_j) = Intersection^2(Def_{Ai}, Def_{Aj}) + 1 \quad (9)$$

where:

$$Intersection(Def_{Ai}, Def_{Aj}) = \max NbCT_{AiAj}(s, t),$$
$$s \in [1, N], t \in [1, M] \quad (10)$$

And *NbCTAiAj* (*s*, *t*) is the number of common words between $subdef_{Ais} \in DefAi$ and $subdef_{Ajt} \in DefAj$.

The new similarity between *Ai* and *Aj* is calculated by comparing word meanings extracted from the WN lexical database since the definitions of similar words frequently employ the same terms. The more similar terms two words have in their definitions, the more similar we might assume they are to one another. This assumption is inspired by our previous work which takes its inspiration from the Lesk algorithm to compute similarities between words. The Lesk algorithm proposes comparing concepts using the number of common terms that appear in their glosses [21].

As shown in Eq. (10), the intersection is computed as the maximum of all $NbCT_{AiAj}(s, t)$. The semantic similarity between A_i and A_j shown in Eq. (9) is then obtained by adding 1 to the square of the intersection's result.

The new SVM kernels are therefore defined as follows:

$$NewLinear(A_i, A_j) = A_i \times A_j + SemanticSim(A_i, A_j) \quad (11)$$

$$NewGaussian(A_i, A_j) = exp\left(-\gamma\left(\parallel A_i - A_j \parallel^2 / SemanticSim(A_i, A_j)\right)\right) \quad (12)$$

$$NewAnova(A_i, A_j) = \sum_{k=1}^{n} e\,xp\left(-\sigma\left((A_{ik} - A_{jk}) / SemanticSim(A_i, A_j)\right)^2\right)^d \quad (13)$$

$$NewBessel(A_i, A_j) = J_0(\sigma \parallel A_i - A_j \parallel) \times SemanticSim(A_i, A_j) \quad (14)$$

If *Ai* and *Aj* are not similar (Intersection (Def_{Ai}, Def_{Aj}) = 0), then the kernel between them is calculated as follows:

- Since the resulting value of SemanticSim (A_i, A_j) is equal to 1, the dot product between Ai and Aj for Linear is set to its basic value.
- Since the resulting value of SemanticSim (A_i, A_j) is equal to 1, the distance between *Ai* and Aj for Anova and Gaussian is set to the standard distance.
- Since the resulting value of SemanticSim (A_i, A_j) is equal to 1, the Bessel function of the first kind, J_0, is set to its basic value for the Bessel kernel.

The result of the intersection is squared to offer increased similarity of terms having more words in common between subsets of their definitions.

IV. Experiments and Results

The experiments conducted to evaluate the suggested approach are presented in this section. Below is a description of the pre-processing methods used, the classifier employed, the datasets utilized, the performance measures adopted, the implementation of the proposed SVM algorithm, and the experimental protocols executed.

A. Experimental Setup and Protocols

- Pre-processing: The first step in pre-processing is corpus parsing, which uses Part of Speech Tagger (POS) to extract a list of adjectives and verbs. The final list is then created by eliminating every stop word from the initial one.
- Classifier used: The SVM algorithm described in Section III is used to conduct experiments. SVM is used with four distinct kernels to evaluate the proposed method: Linear Kernel, Gaussian RBF Kernel, Anova Kernel, and Bessel Kernel.
- Datasets: Three well-known benchmark datasets in ABSA, Restaurant dataset, Products dataset, and Laptop dataset, are used to evaluate the proposed approach. The first corpus is the Restaurant[1] dataset which is distributed for SemEval-2014 ABSA task 4 [39]. It contains 3044 sentences from the English restaurant reviews of *Ganu et al*. [40]. The corpus uses the following preset implicit aspects: price, food, ambiance, service, and anecdotes/miscellaneous.

Cruz-Garcia *et al.* [41] created the Products[2] dataset by individually labeling each IAT. It was created using the customer review corpus from [42]. This dataset contains five corpora for various electronic products namely Apex AD2600, Progressive-scan DVD player, Canon G3, Creative Jukebox, Nikon Coolpix 4300, and Nokia 6610. The implicit aspects considered in this dataset are size, weight, quality, appearance, price, performance, and functionality.

The Laptop[3] corpus is a modified form of the SemEval-2015 ABSA dataset for the laptop domain provided by who updated the SemEval-2015 ABSA dataset with some corrections [43]. This corpus is utilized for ABSA SemEval-2016 task 5 [44]. This dataset contains reviews about laptops. The corpus uses the following preset implicit aspects: connectivity, portability, design features, quality, price, usability, and operation performance.

In every experiment, 10-fold cross-validation is performed to reduce the degree of uncertainty associated with data splitting between training and testing data. The k-fold cross-validation is a technique that involves randomly partitioning the dataset into *k* parts of equal size (where *K* can be any integer but 10 is a common value) and each partition is used for test and other *k*–1 partition are used for training the model.

- Evaluation measures: The following measures are used to evaluate the proposed model:
- F1–Score: the most frequently utilized evaluation metrics that measure the performance of the model are accuracy, precision, recall, and F1–Score. The percentage of successfully predicted samples is called accuracy. When the dataset is unbalanced, precision, recall, and F1–Score are used instead of accuracy because accuracy alone is insufficient. F1–Score is defined as the harmonic mean of recall and precision [45].

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (15)$$

- Equalized Loss of Accuracy (ELA): It was developed by Sáez *et al.* [46]. It is a metric that measures how noise-robust a classifier is. It calculates the accuracy loss by comparing the noise-free performance A_0% with the performance after adding noise A_l%. It is mathematically represented as follows:

$$ELA_l\% = \frac{100 - A_l\%}{A_0\%} \quad (16)$$

where Al% is the classifier's accuracy at the noise level l%. A small ELA value is better than a large ELA value. The higher the ELA value is the more important the

[1] Restaurant: http://metashare.ilsp.gr:8080/repository/browse/semeval-2014-absa-restaurant-reviews-train-data/479d18c0625011e38685842b2b6a04d72cb57ba6c07743b9879d1a04e72185b8/

[2] Product: http://www.gelbukh.com/resources/implicit-aspect-extraction-corpus

[3] Laptop: http://metashare.ilsp.gr:8080/repository/browse/semeval-2016-absa-laptop-reviews-english-train-data-subtask 1/0ec1d3b0563211e58a25842b2b6a04d77d2f0983ccfa4936a25ddb821d46e220/

accuracy loss is at the concerned noise level. Thus, small ELA values are better than large ELA values.

- Implementation of the proposed SVM algorithm: The following figure presents the core of the proposed NewSVM algorithm that is implemented using Scikit-learn machine learning library in Python language.
- Noise tolerance experimental protocol: Experimental protocol is prepared to evaluate the suggested method according to SVM noise tolerance.

It is described in the two subsections that follow, with a focus on each protocol's goal and how it is created to accomplish it (see Algorithm 1).

Algorithm 1: The core of the New_SVM implementation

```
Input:
  CV = KFold(n_splits=10)
  X_train, y_train, X_test, y_test from CrossValidation CV
  X_train_i X_train_j two training samples from X_train
  A_i,A_j the String form of X_train_i and X_train_j
  Extract definitions from Wordnet:
  Function get_defs(string):
    definitions = []
    for syn from wordnet.synsets(string):
      definitions. append (syn. definition())
    return definitions
  Compute the intersection between definitions:
  Function compare_defs (srting1, string2):
    return np.square(len(set(string1).Intersection(string2)))

  Calculate the new similarity:
  Function Similarity(srting1, string2):
    Scores = []
    defs_1=getdefs (string1)
    defs_2=getdefs(string2)
    for d1 in defs_1:
      for d2 in defs_2:
        Scores.append(compare_defs(d1, d2))
    return max (Scores)
  Define the new kernel K:
  Function New Kernel (X_train_i, X_train_j):
    A_i=inverse_transform(X_train_i)
    A_j=inverse_transform(X_train_j)
    sim = Similarity(A_i,A_j)+1
    return the new K formula using sim (Kernel formulas
are presented in the section Proposed Approach)
  Create the NewSVM model using K:
  NewSVM = SVC(kernel='K')
  Train the NewSVM model:
  NewSVM = NewSVM.fit(X_train,y_train)
```

SVM relies on support vectors, which are determined by considering every single instance, to derive the model. Consequently, by including or excluding a single noisy instance, the hyperplanes of SVM can be readily changed. Additionally, due to the implicit interdependence of input attributes, SVM performance may be impacted when noise is included in training data because relationships between attributes are compromised.

A protocol is developed to evaluate and compare the impact of noisy data on the performance of NewSVM and BasicSVM with four kernels: Linear, Gaussian RBF, Anova, and Bessel. To achieve this goal, the proposed protocol should:

- Be based on a noisy data generation mechanism which is significant in terms of (1) where introducing generated noisy data and (2) how much noisy data should be generated. The quality of data heavily depends on one hand on the attributes that are supposed to efficiently characterize instances for classification purposes, and on the other hand on class labels that must reflect the right assignment of class to instances. Therefore, the adopted noisy data generation considers introducing noise in input attributes (attribute noise) and output class (class noise). For each of these two types of noise, different noise levels are generated (0%, 10%, 20%, 30%, 40%, 50%, 60%) where each one represents the quantity (in percentage) of corrupted data generated in each dataset:
- Attribute noise: As used in [47], this type of noise is introduced in the training dataset only, in the test dataset, and in both training and test datasets. For each dataset considered and depending on the desired noise percentage level the corresponding number of examples is modified for each attribute concerning the complete considered dataset. The noise is generated and introduced in the chosen dataset as follows: For a noise level l, l% attributes from the chosen dataset are randomly selected and replaced with a random value from the whole chosen dataset.
- Class noise: As used in [48], for this type of noise, a pairwise class noise is generated because only specific classes are likely to be mislabeled in real-world scenarios. The proportion of the entire training set that is corrupted will therefore be smaller than l% which is the theoretical level. This type of noise is generated in the training dataset as follows: to corrupt the class labels under a noise level l, l% of the instances class labels of the training dataset's majority class are randomly replaced with the second majority class's label. This way of generating noise induces not only mislabeled but also contradictory instances [49].
- Supplies a measure of the impact of noisy data on SVM performance for comparing performances of NewSVM and BasicSVM concerning how they behave towards noisy data classification. F1–Score, Improvement rate, and ELA-score measures are used to compare the noise tolerance of both techniques.

To experimentally study the impact of different types of noise on the performance of SVM, the protocol presented in Algorithm 2 is adopted.

Algorithm 2: Attribute Noise and Class noise generation algorithms

```
Input:
nb_samples is the number of samples in the dataset
noise level the selected quantity of noise
DS={Laptop, Products, Restaurant}
K={Linear, Gaussian RBF, Anova, Bessel}
Model={NewSVM, BasicSVM}
Attribute Noise:
For each dataset DS :
  For each Kernel K :
    For each Model :
```

If NoiseLocation = Noise in train or Noise in test:
for i = 0 to (nb_samples−1) * noise level :
Randomly choose a data value Vi from the set of attributes in training or test data depending on NoiseLocation.
Randomly generate an instance number J from 0 to nb_samples−1.
Replace the value of instance number J with Vi.
If NoiseLocation = Noise in both:
for i=0 to (nb_samples−1) * noise level/2:
Randomly choose a data value Vi_test and Vi_train respectively from the set of attributes in test data and train data.
Randomly generate two instance numbers J_test and J_train from 0 to nb_samples−1.
Replace the value of instance numbers J_test and J_train with Vi_test and Vi_train.
Class Noise:
For each dataset DS :
For each Kernel K :
For each Model :
Identify the two first majority classes C1 and C2
for i = 0 to (nb_samples−1) * noise level:
Randomly generate an instance number J from C1 instances.
Replace the class label of instance number J with C2.

B. Results and discussion

In this section, the experimental findings are presented and discussed considering SVM Noise tolerance.

To study the impact of different types of noise on the behavior and performance of SVM, Experiments are carried out according to the following noise generation schemas:

- Attribute noise in test data.
- Attribute noise in train data.
- Attribute noise in both train and test data.
- Class noise.

Experimental results are provided in Figs. 2–5. These figures summarize F1−Score performance results obtained by NewSVM and BasicSVM for: i) the three datasets (Restaurant, Products Laptop), (ii) the four noise generation schemas (attribute noise in test data, attribute noise in train data, attribute noise in both train and test data and class noise) and (iii) the different noise levels (0%, 10%, 20%, 30%, 40%, 50%, 60%).

From Figs. 2–5, it is noted that in general NewSVM model outperforms BasicSVM for almost the three datasets, the four kernels, and all noise levels. This is due to the new proposed kernel function that helps SVM to improve its noise tolerance. Indeed, the value of the kernel determines how much influence each term in the training data has on classifying new terms. Thus, semantic information (brought by the proposed similarity) is added to the kernel function, the new kernel value is amplified and this increases the degree of influence each term of training data has on its neighbors. This is what explains the performance improvement of NewSVM over BasicSVM.

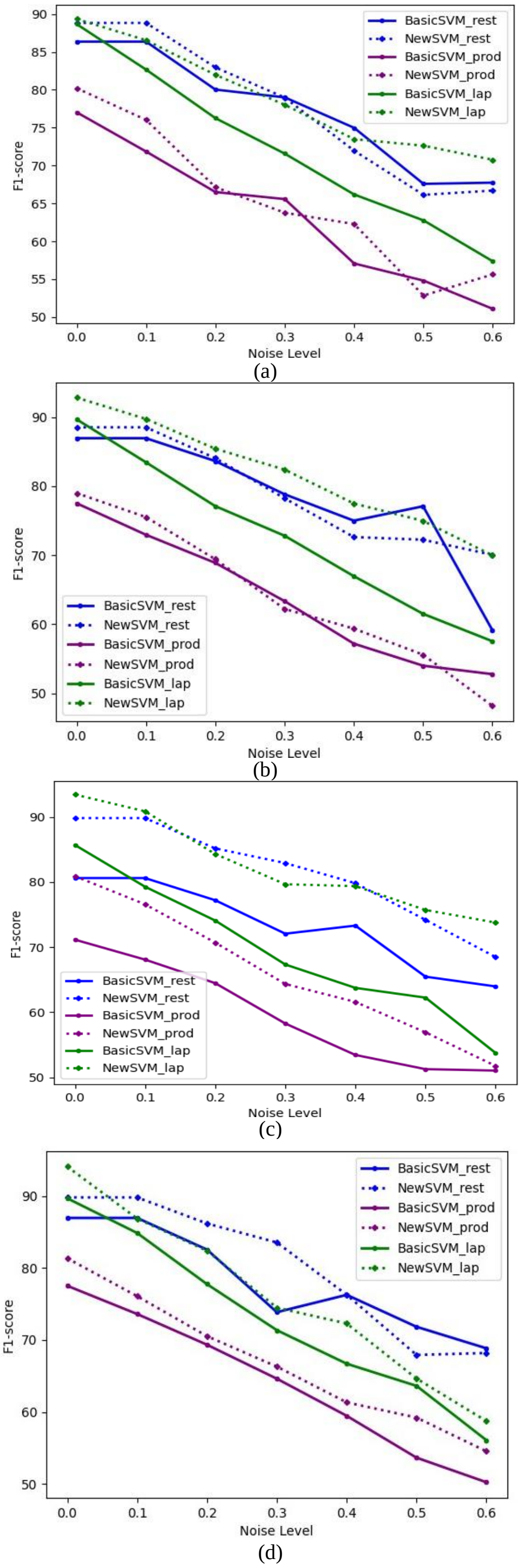
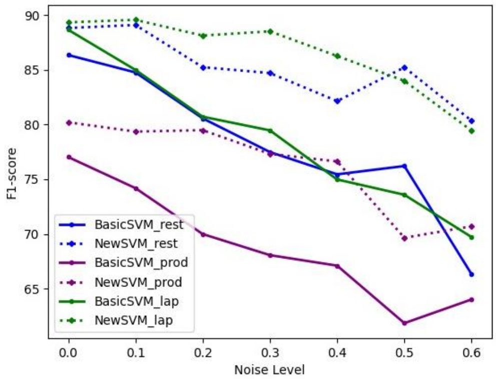
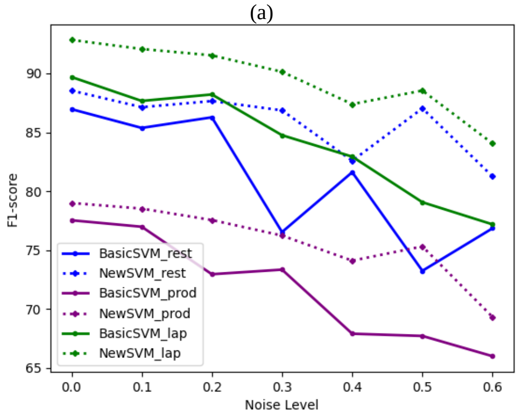


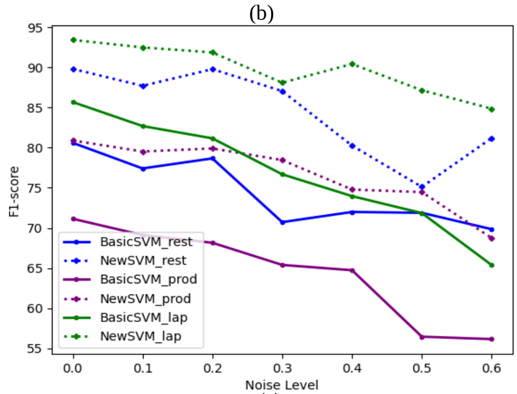
Fig. 2. Attribute noise in the test data, (a) F1−Score performance under attribute noise in test data for BasicSVM and NewSVM with Gaussian Kernel and for all datasets, (b) F1−Score performance under attribute noise in test data for BasicSVM and NewSVM with Anova Kernel and for all datasets, (c) F1−Score performance under attribute noise in test data for BasicSVM and NewSVM with Bessel Kernel and for all datasets, (d) F1−Score performance under Attribute noise in the test for BasicSVM and NewSVM with Linear Kernel and all datasets.



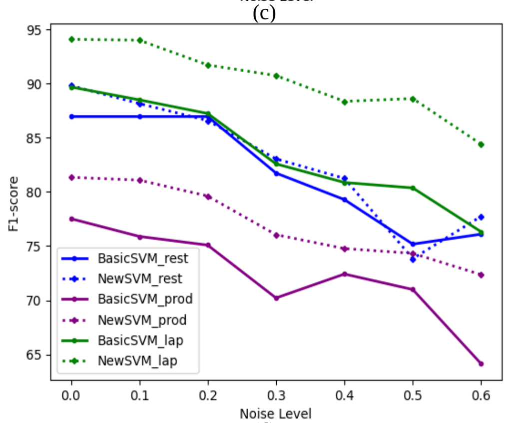
(a)



(b)

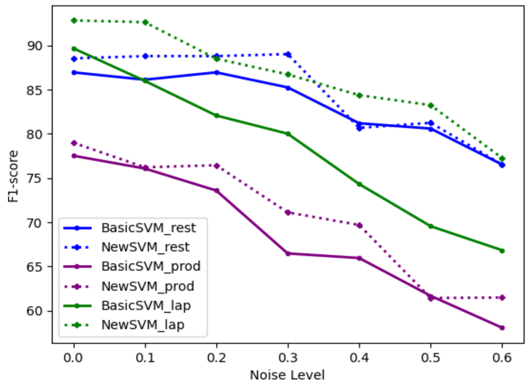


(c)

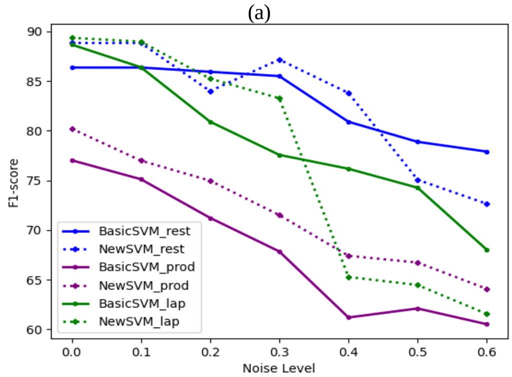


(d)

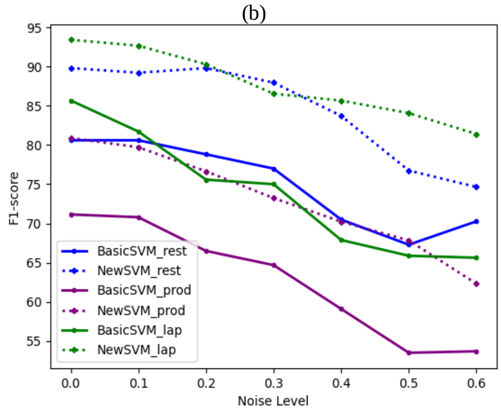
Fig. 3. Attribute noise in train data, (a) F1−Score performance under attribute noise in train data for BasicSVM and NewSVM with Gaussian Kernel and for all datasets, (b) F1−Score performance under attribute noise in train data for BasicSVM and NewSVM with Anova Kernel and for all datasets, (c) F1−Score performance under attribute noise in train data for BasicSVM and NewSVM with Bessel Kernel and for all datasets, (d) F1−Score performance under Attribute noise in train for BasicSVM and NewSVM with Linear Kernel and all datasets.



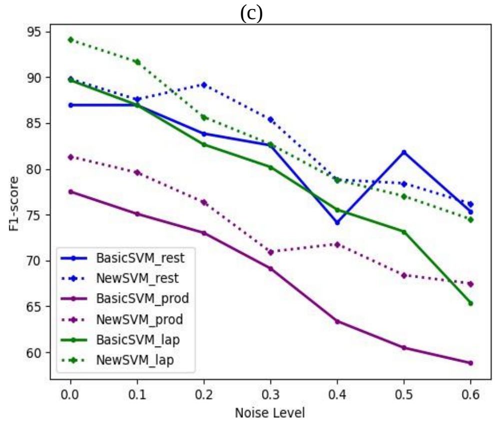
(a)



(b)

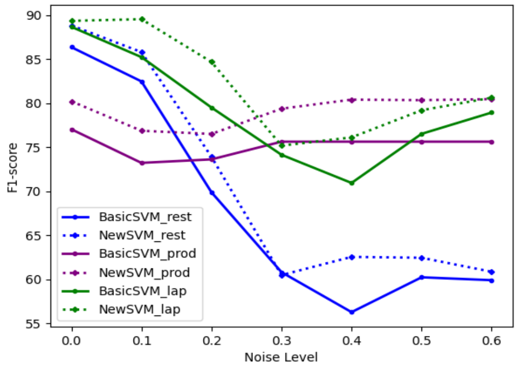


(c)

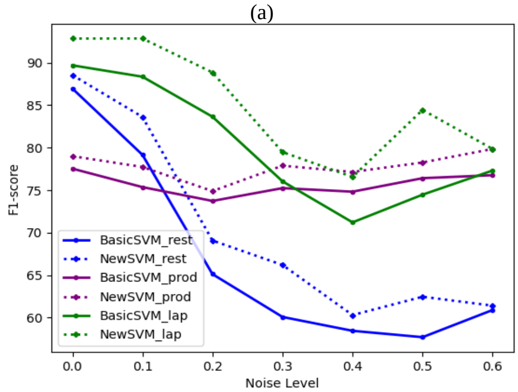


(d)

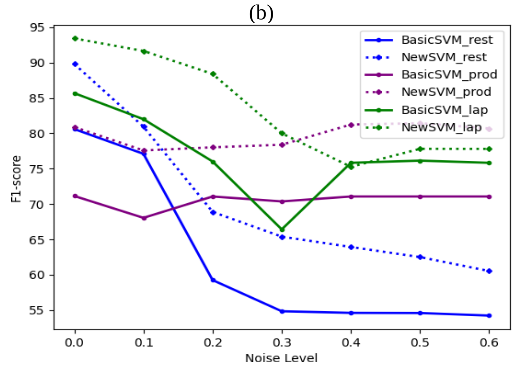
Fig. 4. Attribute noise in both train and test data, (a) F1−Score performance under attribute noise in both train and test data for BasicSVM and NewSVM with Gaussian Kernel and for all datasets, (b) F1−Score performance under attribute noise in both train and test data for BasicSVM and NewSVM with Anova Kernel and for all datasets, (c) F1−Score performance under attribute noise in both train and test data for BasicSVM and NewSVM with Bessel Kernel and for all datasets, (d) F1−Score performance under attribute noise in both train and test data for BasicSVM and NewSVM with Linear Kernel and for all datasets.



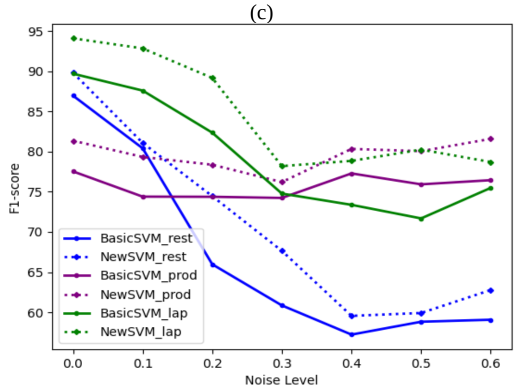
(a)



(b)



(c)



(d)

Fig. 5. class noise, F1–Score performance under class noise for BasicSVM and NewSVM with Gaussian Kernel and all datasets, (b) F1–Score performance under class noise for BasicSVM and NewSVM with Anova Kernel and all datasets, (d) F1–Score performance under class noise for BasicSVM and NewSVM with Linear Kernel and all datasets, (c) F1–Score performance under class noise for BasicSVM and NewSVM with Bessel Kernel and all datasets.

For deeper analysis, a closer look is taken, from different views, at the initial results represented by the above figures. Tuned results are summarized in Tables I–V where the NewSVM noise tolerance superiority measure is introduced. The definition of this superiority is prepared in such a way as to fulfill two conditions. First, taking a positive value means higher noise tolerance of NewSVM over BasicSVM whereas a negative value signifies the opposite. Second, having values that should be proportional to NewSVM superiority. The NewSVM noise tolerance superiority is assigned two definitions. It is defined on the one hand as the average over all noise levels of the F1-score improvement rates of NewSVM over BasicSVM in Tables I and II ($\sum_l \mathrm{IR}_l / \mathrm{nb}_{\mathrm{noiselevels}}$, where $\mathrm{IR}_l = \frac{\mathrm{F1\_score}_{\mathrm{NewSVM}} - \mathrm{F1\_score}_{\mathrm{BasicSVM}}}{\mathrm{F1\_score}_{\mathrm{BasicSVM}}}$, $\mathrm{nb}_{\mathrm{noiselevels}} = 6$ and l represents the noise levels) and on the other hand as the difference between BasicSVM Mean of ELA-scores and NewSVM Mean of ELA-scores in Tables III and V ($\mathrm{Mean\_ELA\_Scores}_{\mathrm{BasicSVM}} - \mathrm{Mean\_ELA\_Scores}_{\mathrm{NewSVM}}$).

These tuned results are used to highlight and discuss SVM noise tolerance superiority according to the following aspects:

A. In General Terms

The initial results show that NewSVM generally outperforms BasicSVM with noisy data. Even with the presence of two special cases (As shown in Table I, negative NewSVM superiority for Gaussian kernel, Restaurant and Laptop datasets, and train and test data noise generation) where NewSVM shows poorer performance than BasicSVM, Table I reveals that NewSVM has higher F1–Score performance than BasicSVM for all noise levels, all noise generation schemas, all datasets, and all kernels.

B. Noise Generation Mode

From initial F1–Score-based figures, NewSVM is shown to be generally more noise tolerant than BasicSVM. However, its noise tolerance superiority level changes with the noise generation schema. Table I shows that:

- For attribute noise generation and all datasets and kernels, the NewSVM noise tolerance superiority level reaches its maximum level with attribute noise in train data (shown in Table I by the Mean column representing NewSVM average noise tolerance superiority by noise generation schema). When attribute noise is introduced in train data only the train SVM model is corrupted which penalizes the BasicSVM (by impacting SVM support vectors and margins) where this penalization reaches its maximum. The proposed new similarity helps reduce the penalization for NewSVM. Thus, this latter shows higher noise tolerance superiority than BasicSVM. When test data is corrupted with fewer train data or without train data, BasicSVM penalization decreases which makes NewSVM noise tolerance superiority decrease too.
- For class noise generation and all datasets and kernels, NewSVM maintains higher noise tolerance superiority than BasicSVM. The class noise superiority is globally lower than attribute noise superiority (shown in Table I by Mean column). In fact, in class noise generation noise is injected in class labels without corrupting train data values.

When only class labels are corrupted and attribute values remain clean, the SVM train model is less penalized than when attribute values are corrupted. Thus, NewSVM noise tolerance superiority is lower in class noise generation than in attribute noise generation.

- According to ELA-score-based results, as noted earlier for F1–Score based results, ELA–Based average superiority of NewSVM reaches its highest level for train data noise generation mode as can be deduced from Tables III–V (NewSVM average superiority values for test data generation mode, train data generation mode, and test and train data generation mode are respectively: 0.33, 0.41, and 0.33, given by the formula: $\sum_{Dataset}(\sum_{Kernel}(\text{Mean\_ELA\_Scores}_{\text{BasicSVM}} - \text{Mean\_ELA\_Scores}_{\text{NewSVM}})/4)/3$. Indeed, when we corrupt test data with fewer train data or without train data, BasicSVM penalization decreases which makes NewSVM superiority decreases too.

TABLE I. AVERAGE F1–SCORE PERFORMANCES BASED SUPERIORITY OVER ALL NOISE LEVELS OF NEWSVM OVER BASICSVM FOR ALL NOISE GENERATION SCHEMAS FOR ALL KERNELS AND ALL DATASETS

Kernel	Noise-Generation	Restaurant			Products			Laptop			Mean[3]
		BasicSVM	NewSVM	Superirority[2]	BasicSVM	NewSVM	Superirority[2]	BasicSVM	NewSVM	Superirority[2]	
Gaussian	Train	78.16	85.09	9.28%	68.89	76.19	10.79%	78.87	86.47	9.99%	10.02%
	Train & Test	83.12	82.90	−0.37%	67.86	71.70	5.82%	78.85	76.88	−2.93%	0.84%
	Test	77.43	77.77	0.23%	63.41	65.41	3.23%	72.21	78.97	10.30%	5.58%
	Class	67.98	70.68	4.08%	75.19	79.16	5.28%	79.12	82.09	3.81%	4.39%
	Average[1]	76.67	79.11	3.31%	68.84	73.87	6.28%	77.26	81.10	5.29%	–
Anova	Train	80.97	85.86	6.39%	71.77	75.72	5.65%	84.21	89.51	6.42%	6.15%
	Train & Test	83.38	84.80	1.66%	68.48	70.77	3.45%	78.35	86.52	10.90%	5.33%
	Test	78.24	79.19	1.77%	63.82	64.19	0.35%	72.73	81.84	13.46%	5.19%
	Class	66.91	70.23	5.14%	75.70	77.82	2.81%	80.10	84.27	5.28%	4.41%
	Average[1]	77.37	80.02	3.74%	69.94	72.13	3.07%	78.85	85.54	9.02%	–
Bessel	Train	74.44	84.41	13.45%	64.44	76.68	19.40%	76.77	89.76	17.47%	16.77%
	Train & Test	75.00	84.55	12.77%	62.77	72.98	16.64%	73.91	87.72	19.30%	16.23%
	Test	73.71	81.45	11.07%	59.67	66.11	10.56%	69.44	82.43	19.87%	13.83%
	Class	62.17	70.28	13.58%	70.55	79.75	13.04%	76.84	83.48	8.81%	11.81%
	Average[1]	71.33	80.17	12.72%	64.36	73.88	14.91%	74.24	85.85	16.36%	–
Linear	Train	81.88	82.92	1.24%	72.32	77.08	6.7%	83.65	90.28	8.04%	5.33%
	Train & Test	81.66	83.64	2.45%	68.21	73.71	8.47%	79.09	83.5	5.8%	5.57%
	Test	78.18	80.25	2.53%	64.06	67.05	4.96%	72.87	76.22	4.62%	4.04%
	Class	67.04	70.74	5.76%	75.74	79.6	5.1%	79.28	84.58	6.77%	5.88%
	Average[1]	77.19	79.39	3%	70.08	74.36	6.31%	78.72	83.65	6.31%	–

[1] Average is defined, for each dataset and each kernel, as either (The average F1–Score) or (The average NewSVM superiority) overall noise generation schemas and all noise levels.
[2] Superiority is defined (for each dataset, each kernel, and each noise generation schema) as the average improvement rate of NewSVM over BasicSVM for all noise levels.
[3] Mean is defined, for each noise generation schema and each kernel, as the average NewSVM superiority over all datasets and all noise levels.

TABLE II. COMPARISON OF F1–SCORE AVERAGE IMPROVEMENT RATES OF NEWSVM AND BASICSVM FOR SMALL NOISE LEVELS, LARGE NOISE LEVELS, THE THREE DATASETS, AND THE THREE KERNELS USED

Noise-Generation	Superiority/ Kernel	Restaurant				Products				Laptop			
		Gaussian	Anova	Bessel	linear	Gaussian	Anova	Bessel	linear	Gaussian	Anova	Bessel	linear
Attribute-noise-Train	Superiority Rsmall[1]:	5.42%	4.49%	12.58%	0.71%	7.97%	3.11%	11.74%	5.19%	7.02%	4.37%	10.63%	6.05%
	Superiority Rlarge[2]:	9.92%	6.39%	16.94%	0.75%	8.01%	5.71%	13.55%	4.64%	10.48%	6.94%	17.08%	7.94%
Attribute-noise-Test	Superiority Rsmall[1]:	1.80%	0.47%	9.34%	5.39%	1.01%	0.65%	6.93%	1.78%	5.35%	8.06%	11.37%	3.26%
	Superiority Rlarge[2]:	1.84%	1.23%	6.58%	–1.51%	2.59%	–0.27%	7.84%	3.92%	10.19%	12.14%	16.36%	3.09%
Attribute-noise-Both	Superiority Rsmall[1]:	0.73%	2.76%	10.19%	2.93%	3.09%	2.55%	9.21%	3.22%	4.21%	6.59%	12.40%	3.40%
	Superiority Rlarge[2]:	2.06%	0.05%	9.02%	0.72%	4.79%	2.31%	11.36%	8.35%	9.04%	11.40%	17.25%	5.41%
Class-noise	Superiority Rsmall[1]:	2.35%	4.84%	8.01%	5.31%	3.43%	2.05%	8.17%	3.63%	3.51%	2.71%	11.89%	5.16%
	Superiority Rlarge[2]:	3.15%	2.37%	7.84%	2.37%	4.77%	2.42%	10.05%	4.10%	3.18%	5.95%	1.03%	5.75%

[1] the difference between the average F1–Score of NewSVM and the average F1–Score of BasicSVM in low noise area represented by noise levels [10%,20%,30%].
[2] the difference between the average F1–Score of NewSVM and the average F1–Score of BasicSVM in high noise area represented by noise levels [40%,50%,60%].

TABLE III. ELA–SCORES ACROSS ALL MODELS AND ALL NOISE LEVELS FOR RESTAURANT DATASET

Noise generation schema	ELA–Score with noise level/Model	ELA0%	ELA10%	ELA20%	ELA30%	ELA40%	ELA50%	ELA60%	Mean
Class-noise	BasicGaussian	0.17	0.21	0.33	0.44	0.57	0.63	0.61	0.42
	NewGaussian	0.13	0.14	0.22	*0.47*	*0.6*	*0.64*	*0.62*	0.40
	BasicAnova	0.15	0.23	0.38	0.48	0.61	0.61	0.65	0.44
	NewAnova	0.14	0.16	0.27	0.4	0.52	0.6	0.58	0.38
	BasicBessel	0.30	0.36	0.55	0.71	0.77	0.83	0.85	0.62
	NewBessel	0.13	0.19	0.29	0.38	0.43	0.51	0.52	0.35
	BasicLinear	0.15	0.21	0.35	0.48	0.55	0.60	0.61	0.42
	NewLinear	0.13	0.19	0.24	0.38	0.50	0.51	0.48	0.35
Attribute-noise-Train	BasicGaussian	0.17	0.18	0.21	0.25	0.25	0.27	0.33	0.24
	NewGaussian	0.13	0.13	0.14	0.15	0.22	0.18	0.23	0.17
	BasicAnova	0.15	0.17	0.19	0.29	0.23	0.30	0.29	0.23
	NewAnova	0.14	0.15	0.15	0.14	0.19	0.20	0.22	0.17
	BasicBessel	0.3	0.34	0.33	0.43	0.39	0.42	0.46	0.38
	NewBessel	0.13	0.13	0.13	0.15	0.21	0.31	0.23	0.18
	BasicLinear	0.15	0.15	0.15	0.21	0.25	0.28	0.32	0.21
	NewLinear	0.13	0.15	0.16	0.18	0.2	0.27	0.29	0.20
Attribute-noise-Test	BasicGaussian	0.17	0.17	0.24	0.24	0.33	0.37	0.41	0.27
	NewGaussian	0.13	0.13	0.19	0.24	0.32	0.38	0.36	0.25
	BasicAnova	0.15	0.15	0.19	0.25	0.31	0.27	0.45	0.25
	NewAnova	0.14	0.14	0.19	0.26	0.29	0.32	0.36	0.24
	BasicBessel	0.3	0.3	0.36	0.41	0.41	0.5	0.51	0.40
	NewBessel	0.13	0.13	0.19	0.21	0.24	0.32	0.35	0.22
	BasicLinear	0.15	0.15	0.2	0.29	0.31	0.33	0.37	0.26
	NewLinear	0.13	0.13	0.18	0.19	0.29	0.38	0.36	0.24
Attribute-noise-Both	BasicGaussian	0.17	0.17	0.19	0.17	0.22	0.26	0.29	0.21
	NewGaussian	0.13	0.13	0.16	0.14	0.19	0.27	0.28	0.18
	BasicAnova	0.15	0.16	0.15	0.18	0.24	0.25	0.3	0.20
	NewAnova	0.14	0.13	0.13	0.12	0.24	0.25	0.24	0.18
	BasicBessel	0.3	0.3	0.32	0.37	0.41	0.47	0.46	0.37
	NewBessel	0.13	0.14	0.13	0.17	0.21	0.24	0.27	0.18
	BasicLinear	0.15	0.15	0.17	0.18	0.28	0.26	0.28	0.21
	NewLinear	0.13	0.13	0.14	0.14	0.24	0.27	0.24	0.18

ELAl%: is defined as the ELA–Score value for noise level l.
Mean: represents NewSVM average ELA–Score values over all noise levels.

TABLE IV. ELA–SCORES ACROSS ALL MODELS AND ALL NOISE LEVELS FOR PRODUCTS DATASET

Noise generation schema	ELA–Score with noise level / Model	ELA0%	ELA10%	ELA20%	ELA30%	ELA40%	ELA50%	ELA60%	Mean
Class-noise	BasicGaussian	0.34	0.41	0.46	0.47	0.47	0.47	0.47	0.44
	NewGaussian	0.26	0.3	0.34	0.35	0.35	0.35	0.35	0.33
	BasicAnova	0.33	0.36	0.42	0.43	0.43	0.44	0.44	0.41
	NewAnova	0.28	0.31	0.34	0.36	0.37	0.37	0.38	0.34
	BasicBessel	0.5	0.59	0.62	0.62	0.62	0.62	0.62	0.60
	NewBessel	0.28	0.32	0.34	0.37	0.36	0.38	0.38	0.35
	BasicLinear	0.33	0.42	0.43	0.43	0.44	0.44	0.44	0.42
	NewLinear	0.27	0.3	0.32	0.36	0.35	0.36	0.37	0.33
Attribute-noise-Train	BasicGaussian	0.34	0.35	0.41	0.39	0.44	0.48	0.47	0.41
	NewGaussian	0.26	0.27	0.27	0.3	0.32	0.36	0.35	0.30
	BasicAnova	0.33	0.34	0.4	0.39	0.43	0.48	0.51	0.41
	NewAnova	0.28	0.29	0.31	0.31	0.31	0.34	0.38	0.32
	BasicBessel	0.5	0.53	0.57	0.57	0.61	0.74	0.73	0.61
	NewBessel	0.28	0.31	0.31	0.31	0.35	0.37	0.43	0.34
	BasicLinear	0.33	0.35	0.37	0.41	0.4	0.43	0.51	0.40
	NewLinear	0.27	0.28	0.28	0.34	0.36	0.36	0.36	0.32
Attribute-noise-Test	BasicGaussian	0.34	0.4	0.46	0.49	0.58	0.6	0.66	0.50
	NewGaussian	0.26	0.32	0.4	0.45	0.47	0.55	0.56	0.43
	BasicAnova	0.33	0.38	0.44	0.48	0.55	0.61	0.62	0.49
	NewAnova	0.28	0.33	0.39	0.47	0.5	0.55	*0.64*	0.45
	BasicBessel	0.5	0.54	0.61	0.67	0.72	0.75	0.75	0.65
	NewBessel	0.28	0.34	0.41	0.48	0.51	0.54	0.62	0.45
	BasicLinear	0.33	0.38	0.43	0.49	0.55	0.6	0.64	0.49
	NewLinear	0.27	0.32	0.4	0.44	0.51	0.53	0.58	0.43
Attribute-noise-Both	BasicGaussian	0.34	0.35	0.42	0.46	0.51	0.5	0.54	0.44
	NewGaussian	0.26	0.29	0.32	0.35	0.41	0.43	0.46	0.36
	BasicAnova	0.33	0.36	0.37	0.44	0.46	0.54	0.55	0.43
	NewAnova	0.28	0.3	0.34	0.38	0.42	0.47	0.48	0.38
	BasicBessel	0.5	0.52	0.56	0.59	0.62	0.72	0.73	0.60
	NewBessel	0.28	0.28	0.34	0.39	0.41	0.47	0.47	0.38

	BasicLinear	0.33	0.36	0.39	0.45	0.49	0.53	0.56	0.44
	NewLinear	0.27	0.29	0.33	0.39	0.37	0.41	0.45	0.36

ELAl%: is defined as the ELA–Score value for noise level l.
Mean: represents NewSVM average ELA–Score values over all noise levels.

TABLE V. ELA–SCORES ACROSS ALL MODELS AND ALL NOISE LEVELS FOR LAPTOP DATASET

Noise generation schema	ELA-score with noise level / Model	ELA0%	ELA10%	ELA20%	ELA30%	ELA40%	ELA50%	ELA60%	Mean%
Class-noise	BasicGaussian	0.12	0.18	0.28	0.38	0.46	0.48	0.48	0.34
	NewGaussian	0.1	0.11	0.21	0.39	0.45	0.45	0.45	0.31
	BasicAnova	0.11	0.13	0.24	0.37	0.43	0.44	0.45	0.31
	NewAnova	0.08	0.08	0.17	0.36	0.41	0.43	0.42	0.28
	BasicBessel	0.17	0.22	0.34	0.5	0.55	0.54	0.55	0.41
	NewBessel	0.06	0.1	0.17	0.32	0.39	0.4	0.4	0.26
	BasicLinear	0.11	0.15	0.26	0.39	0.42	0.44	0.46	0.32
	NewLinear	0.06	0.08	0.15	0.34	0.39	0.4	0.41	0.26
Attribute-noise-Train	BasicGaussian	0.12	0.13	0.15	0.16	0.2	0.22	0.23	0.17
	NewGaussian	0.1	0.1	0.11	0.12	0.13	0.17	0.2	0.13
	BasicAnova	0.11	0.13	0.13	0.16	0.17	0.22	0.24	0.16
	NewAnova	0.08	0.08	0.09	0.1	0.13	0.14	0.16	0.11
	BasicBessel	0.17	0.2	0.21	0.25	0.27	0.29	0.34	0.25
	NewBessel	0.06	0.07	0.07	0.09	0.09	0.12	0.14	0.09
	BasicLinear	0.11	0.12	0.13	0.18	0.2	0.2	0.24	0.17
	NewLinear	0.06	0.06	0.08	0.09	0.09	0.11	0.15	0.09
Attribute-noise-Test	BasicGaussian	0.12	0.18	0.24	0.3	0.35	0.38	0.45	0.29
	NewGaussian	0.1	0.13	0.18	0.21	0.26	0.27	0.3	0.21
	BasicAnova	0.11	0.17	0.23	0.28	0.34	0.39	0.44	0.28
	NewAnova	0.08	0.11	0.15	0.18	0.23	0.26	0.29	0.18
	BasicBessel	0.17	0.23	0.29	0.35	0.39	0.43	0.47	0.33
	NewBessel	0.06	0.09	0.15	0.18	0.2	0.24	0.27	0.17
	BasicLinear	0.11	0.16	0.22	0.29	0.34	0.37	0.45	0.28
	NewLinear	0.06	0.12	0.18	0.25	0.27	0.34	0.4	0.23
Attribute-noise-Both	BasicGaussian	0.12	0.15	0.2	0.23	0.27	0.29	0.33	0.227
	NewGaussian	0.1	0.1	0.13	0.16	0.36	0.38	0.41	0.234
	BasicAnova	0.11	0.14	0.18	0.21	0.24	0.29	0.33	0.21
	NewAnova	0.08	0.07	0.12	0.14	0.16	0.17	0.23	0.14
	BasicBessel	0.17	0.21	0.26	0.28	0.33	0.37	0.37	0.28
	NewBessel	0.06	0.07	0.09	0.13	0.14	0.16	0.18	0.12
	BasicLinear	0.11	0.15	0.18	0.21	0.26	0.29	0.35	0.22
	NewLinear	0.06	0.08	0.13	0.16	0.2	0.22	0.26	0.16

ELAl%: is defined as the ELA–Score value for noise level l.
Mean: represents NewSVM average ELA–Score values overall noise levels

C. Used Kernels

Concerning used kernels, NewSVM noise tolerance superiority (over all datasets and all noise generation schemas) reaches its maximum level with Bessel kernel for both F1–Score based and ELA–Score based evaluations. This is shown by the Mean column in Table I and by the difference between BasicSVM Mean of ELA-scores d NewSVM Mean of ELA-scores ($\text{Mean\_ELA\_Scores}_{\text{BasicSVM}} - \text{Mean\_ELA\_Scores}_{\text{NewSVM}}$) in Tables III–V. This is mainly due to the fact that BasicSVM performs very poorly with Bessel kernel which makes NewSVM's superiority much higher.

According to ELA–Based results, NewSVM shows smaller ELA–Scores and higher superiority than BasicSVM for all noise levels, all noise generation schemas, all datasets, and all kernels. This means that NewSVM presents less accuracy loss with noisy data and therefore shows higher noise tolerance than BasicSVM.

D. Noise Quantity

Table II is derived, from initial F1–Score-based figures, to show the change of NewSVM noise tolerance superiority level concerning the quantity of noise introduced in data. Table II shows that NewSVM is generally more performant than BasicSVM for both Rsmall and Rlarge noise quantity ranges. However, the average Improvement Rate is generally higher in high noise areas (shown by Superiority Rlarge in Table II) than in low noise areas (shown by Superiority Rsmall in Table II). Low noise area is represented by the noise levels (10%, 20%, 30%), whereas high noise area is represented by the noise levels (40%, 50%, 60%). This implies that NewSVM noise tolerance superiority concerning BasicSVM is higher for big noise quantity than for small noise quantity. Tables III–V are introduced to show the change of NewSVM noise tolerance (measured by ELA–Score) with respect to the level of noise introduced in data. They show that NewSVM is generally less affected with noise for all noise levels. When closer look is taken at ELA–score values for all noise generation schemas and models, it is noted that ELA-score increases as noise levels get higher. This means that for all kernels, all noise generation schemas, and all datasets both BasicSVM and NewSVM models are less tolerant to noise for higher noise levels than for smaller noise levels. There are two special cases where NewSVM is less noise tolerant than BasicSVM (shown by higher ELA–score of NewSVM, on one hand for Gaussian kernel and noise levels [30%, 60%] for class noise generation schema in Restaurant dataset and on the other hand for Anova kernel and noise level 60% for test data generation schema in Product dataset). Even with the presence of these two special cases, ELA−score mean values over all noise levels are generally smaller for NewSVM than for BasicSVM which proves the higher noise tolerance of NewSVM over all noise levels, all noise generation schemas, all datasets, and all kernels.

V. CONCLUSION

In this article a technique is proposed to improve SVM's ability to handle noise in Implicit Aspect Identification. By the use of WordNet semantic relations namely 'definition relation', SVM kernel computation is improved to better deal with noisy data when addressing the IAI task. The experiments are performed on three benchmark datasets, Restaurant, Products, and Laptop. The results of the proposed approach are evaluated and analyzed based on its impact on the performance of SVM for two types of noise (class noise and attribute noise) and two types of kernels (one linear kernel and three nonlinear kernels). The following is a summary of the main findings of our study:

- The suggested technique helps SVM better deal with various types of noise, by reducing noise impacts on SVM and therefore enhancing its noise tolerance for all kernels used.
- In general, SVM noise tolerance and NewSVM noise tolerance superiority level reach their maximum with train data noise generation schema.
- SVM noise tolerance decreases proportionally with the quantity of noise included in data, and NewSVM noise tolerance superiority is higher for large noise quantity than for small noise quantity.

Future work will consider investigating other Wordnet semantic relations, namely synonym and antonym relations, to improve SVM for IAI task. In fact, synonyms and antonyms are pertinent knowledge that can be exploited to enrich either machine learning models or their training data for IAI tasks. Additionally, the proposed method will be also considered to enhance other classifiers for ABSA, especially supervised learning classifiers.

CONFLICT OF INTEREST

The authors declare no conflicts of interest statement.

AUTHOR CONTRIBUTIONS

All authors contributed to the study conception and design. Experimental protocol design is prepared by Benarafa Halima, Mohammed Benkhalifa, and Moulay Akhloufi. Experiments are conducted by Benarafa Halima. Results analysis and discussion were performed by Benarafa Halima, Mohammed Benkhalifa, and Moulay Akhloufi. The first draft of the manuscript was written by Benarafa Halima. Mohammed Benkhalifa, and Moulay Akhloufi commented on previous versions of the manuscript. All authors read and approved the final manuscript.

REFERENCES

[1] B. Liu, *Sentiment Analysis and Opinion Mining*, *Synthesis Lectures on Human Language Technologies*, Springer Cham, 2012.

[2] K. R. Chowdhary, *Fundamentals of Artificial Intelligence*, Springer New Delhi, 2020. https://doi.org/10.1007/978-81-322-3972-7

[3] W. D. Alnatara and M. L. Khodra, "Imbalanced data handling in multi-label aspect categorization using oversampling and ensemble learning," in *Proc. 2020 International Conference on Advanced Computer Science and Information Systems (ICACSIS)*, Depok, Indonesia, 2020.

[4] A. Nazir, Y. Rao, L. Wu, and L. Sun, "Issues and challenges of aspect-based sentiment analysis: A comprehensive Survey," *IEEE Transactions on Affective Computing*, 2020.

[5] K. Schouten and F. Frasincar, "Survey on aspect-level sentiment analysis," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 3, pp. 813–830, Mar. 2016.

[6] P. K. Soni and R. Rambola, "A survey on implicit aspect detection for sentiment analysis: Terminology, issues, and scope," *IEEE Access*, vol. 10, pp. 63932–63957, 2022.

[7] H. X. Li, J. L. Yang, G. Zhang, and B. Fan, "Probabilistic support vector machines for classification of noise affected data," *Information Sciences*, vol. 221, pp. 60–71, Feb. 2013.

[8] S. A. Mahtab, N. Islam, and M. M. Rahaman, "Sentiment analysis on bangladesh cricket with support vector machine," in *Proc. 2018 International Conference on Bangla Speech and Language Processing (ICBSLP)*, Sylhet, Bangladesh, 2018.

[9] R. Uma, A. H. Sana, P. Jawahar, and B. V. rishitha, "Support vector machine and convolutional neural Network approach to customer review sentiment analysis," in *Proc. 2022 1st International Conference on Computational Science and Technology (ICCST)*, Nov. 2022.

[10] I. Hidayah, A. E. Permanasari, and N. W. Wijayanti, "Sentiment analysis on product review using Support Vector Machine (SVM)," in *Proc. 2019 5th International Conference on Science and Technology*, pp. 30–31, 2019.

[11] A. Smadi, M. A. Smadi, O. Qawasmeh, M. A. Ayyoub, Y. Jararweh, and B. Gupta, "Deep recurrent neural network vs. support vector machine for aspect-based sentiment analysis of Arabic hotels' reviews," *Journal of Computational Science*, vol. 27, pp. 386–393, Jul. 2018. https://doi.org/10.1016/j.jocs.2017.11.006

[12] I. T. Imam, K. M. Ting, and J. Kamruzzaman, "z-SVM: An SVM for improved classification of imbalanced data," *Lecture Notes in Computer Science*, pp. 264–273, Jan. 2006. https://doi.org/10.1007/11941439_30

[13] M. Lesk, "Automatic sense disambiguation using machine readable dictionaries," in *Proc. International Conference on Systems*, Jun. 1986. https://doi.org/10.1145/318723.318728

[14] J. Rothstein, "Machine translation of languages." *Massachusetts Institute of Technology*, pp. 768–769, Oct. 1955. https://doi.org/10.1126/science.122.3173.768-b

[15] N. Ide and J. Véronis, "Introduction to the special issue on word sense disambiguation: The state of the art," *Computational Linguistics*, vol. 24, no. 1, pp. 2–40, Mar. 1998.

[16] M. Maru, S. Conia, M. Bevilacqua, and R. Navigli, "Nibbling at the hard core of word sense disambiguation," in *Proc. 60th Annual Meeting of THE Association for Computational Linguistics*, Jan. 2022. https://doi.org/10.18653/v1/2022.acl-long.324

[17] S. Conia and R. Navigli, "Framing word sense disambiguation as a multi-label problem for model-agnostic knowledge integration," in *Proc. 16th Conference of the European Chapter of the Association for Computational Linguistics*, 2021, pp. 3269–3275. https://doi.org/10.18653/v1/2021.eacl-main.286

[18] G. A. Miller, "WordNet: A lexical database for English," *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, Nov. 1995. https://doi.org/10.1145/219717.219748

[19] W. Zhang, X. Li, Y. Deng, L. Bing, and W. Lam, "A survey on aspect-based sentiment analysis: Tasks, methods, and challenges," *IEEE Transactions on Knowledge and Data Engineering*, pp. 1–20, 2022. https://doi.org/10.1109/tkde.2022.3230975

[20] H. Benarafa, M. Benkhalifa, and M. Akhloufi, "WordNet Semantic relations-based enhancement of KNN model for implicit aspect identification in sentiment analysis," *International Journal of Computational Intelligence Systems*, vol. 16, no. 1, Jan. 2023. https://doi.org/10.1007/s44196-022-00164-8

[21] H. Benarafa, M. Benkhalifa, and M. A. Akhloufi, "An enhanced SVM model for implicit aspect identification in sentiment Analysis," *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 5, Jan. 2023. https://doi.org/10.14569/ijacsa.2023.0140505

[22] A. R. Pathak, M. Pandey, and S. Rautaray, "Topic-level sentiment analysis of social media data using deep learning," *Applied Soft Computing*, 107440, Apr. 2021. https://doi.org/10.1016/j.asoc.2021.107440

[23] S. Datta and S. Chakrabarti, "Aspect based sentiment analysis for demonetization tweets by optimized recurrent neural network using fire fly-oriented multi-verse optimizer," *Sadhana*, vol. 46, no. 2, Apr. 2021. doi: https://doi.org/10.1007/s12046-021-01608-1

[24] M. Tubishat, N. Idris, and M. Abushariah, "Explicit aspects extraction in sentiment analysis using optimal rules combination," *Future Generation Computer Systems*, vol. 114, pp. 448–480, Jan. 2021. https://doi.org/10.1016/j.future.2020.08.019

[25] P. K. Soni and R. K. Rambola, "Deep learning, WordNet, and spaCy based hybrid method for detection of implicit aspects for sentiment analysis," in *Proc. 2021 International Conference on Intelligent Technologies (CONIT)*, Jun. 2021, pp. 25–27. https://doi.org/10.1109/conit51480.2021.9498372

[26] M. Sivakumar and U. S. Reddy, "Aspect based sentiment analysis of students opinion using machine learning techniques," in *Proc 2017 International Conference on Inventive Computing and Informatics (ICICI)*, Nov. 2017, pp. 23–24. https://doi.org/10.1109/icici.2017.8365231

[27] M. T. Khan, M. Durrani, S. Khalid, and F. Aziz, "Lifelong aspect extraction from big data: Knowledge engineering," *Complex Adaptive Systems Modeling*, vol. 4, no. 1, Mar. 2016. https://doi.org/10.1186/s40294-016-0018-7

[28] N. Liu and B. Shen, "Aspect-based sentiment analysis with gated alternate neural network," *Knowledge-Based Systems*, vol. 188, 105010, Jan. 2020. https://doi.org/10.1016/j.knosys.2019.105010

[29] H. Fei, T.-S. Chua, C. Li, D. Ji, M. Zhang, and Y. Ren, "On the robustness of aspect-based sentiment analysis: Rethinking model, data, and training," *ACM Transactions on Information Systems*, vol. 41, no. 2, pp. 1–32, Dec. 2022. https://doi.org/10.1145/3564281

[30] X. Chen, B. He, Kai Lung Hui, L. Sun, and Y. Sun, "Dealing with textual noise for robust and effective BERT re-ranking," *Information Processing and Management*, vol. 60, no. 1, pp. 103135–103135, 2023. https://doi.org/10.1016/j.ipm.2022.103135

[31] A. Kumar, P. Gupta, R. Balan, L. B. M. Neti, and A. Malapati, "BERT based semi-supervised hybrid approach for aspect and sentiment classification," *Neural Processing Letters*, vol. 30, Jul. 2021. https://doi.org/10.1007/s11063-021-10596-6

[32] B. Wang, L. Ding, Q. Zhong, X. Li, and D. Tao, "A contrastive cross-channel data augmentation framework for aspect-based sentiment analysis," arXiv preprint, arXiv:2204.07832, Apr. 2022.

[33] W. Shi and J. Zhang, "Integration of local position-POS awareness and global dense connection for ABSA," *Journal of Experimental and Theoretical Artificial Intelligence*, pp. 1–21, May 2023. https://doi.org/10.1080/0952813x.2023.2217811

[34] G. Li, H. Wang, Y. Ding, K. Zhou, and X. Yan, "Data augmentation for aspect-based sentiment analysis," *International Journal of Machine Learning and Cybernetics*, vol. 14, no. 1, pp. 125–133, May 2022. https://doi.org/10.1007/s13042-022-01535-5

[35] V. N. Vapnik, "The nature of statistical learning theory," *IEEE Transactions on Neural Networks*, vol. 10, 1995. https://doi.org/10.1007/978-1-4757-2440-0

[36] T. Hofmann, B. Schölkopf, and A. J. Smola, "Kernel methods in machine learning," *The Annals of Statistics*, vol. 36, no. 3, pp. 1171–1220, Jun. 2008. https://doi.org/10.1214/009053607000000677

[37] A. Patle and D. S. Chouhan, "SVM kernel functions for classification," in *Proc. 2013 International Conference on Advances in Technology and Engineering (ICATE)*, Mumbai, India, 2013. doi: 10.1109/ICAdTE.2013.6524743

[38] K. Adhikary, S. Bhushan, S. Kumar, and K. Dutta, "Evaluating the performance of various SVM kernel functions based on basic features extracted from KDDCUP'99 dataset by random forest method for detecting DDoS attacks," *Wireless Personal Communications*, vol. 123, Oct. 2021. https://doi.org/10.1007/s11277-021-09280-8

[39] M. Pontiki, D. Galanis, J. Pavlopoulos, H. Papageorgiou, I. Androutsopoulos, and S. Manandhar, "SemEval-2014 task 4: Aspect based sentiment analysis," *ACLWeb*, vol. 1, 2014. https://www.aclweb.org/anthology/S14-2004/

[40] G. Ganu, N. Elhadad, and A. Marian, "Beyond the stars: Improving rating predictions using review Text content," in *Proc. Twelfth International Workshop on the Web and Databases (WebDB 2009)*, Jan. 2009.

[41] C. I. Cruz, A. F. Gelbukh, and G. Sidorov, "Implicit aspect Indicator extraction for aspect-based opinion mining," *Int. J. Comput. Linguistics Appl.*, vol. 5, pp. 135–152, Jan. 2014.

[42] M. Hu and B. Liu, "Mining and summarizing customer reviews," in *Proc. 2004 ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2004, pp. 168–177. https://doi.org/10.1145/1014052.1014073

[43] M. Pontiki, D. Galanis, H. Papageorgiou, S. Manandhar, and I. Androutsopoulos, "Semeval-2015 task 12: Aspect based sentiment analysis," in *Proc. 9th International Workshop on Semantic Evaluation*, 2015.

[44] M. Pontiki *et al*., "SemEval-2016 task 5: Aspect based sentiment analysis," *HAL Archives Ouvertes*, pp. 19–30, 2016.

[45] B. C. Vickery, "Reviews: Van Rijsbergen, C. J. information retrieval. 2nd edn. London, Butterworths, I978. 208pp," *Journal of Librarianship*, vol. 11, no. 3, pp. 237–237, Jul. 1979. https://doi.org/10.1177/096100067901100306

[46] J. A. Sáez, J. Luengo, and F. Herrera, "Evaluating the classifier behavior with noisy data considering performance and robustness: The equalized loss of accuracy measure," *Neurocomputing*, vol. 176, pp. 26–35, Feb. 2016.

[47] D. F. Nettleton, A. O. Puig, and A. Fornells, "A study of the effect of different types of noise on the precision of supervised learning techniques," *Artificial Intelligence Review*, vol. 33, no. 4, pp. 275–306, Jan. 2010. https://doi.org/10.1007/s10462-010-9156-z

[48] X. Zhu and X. Wu, "Class noise vs. attribute noise: A quantitative study," *Artificial Intelligence Review*, vol. 22, no. 3, pp. 177–210, Nov. 2004. https://doi.org/10.1007/s10462-004-0751-8

[49] S. Gupta and A. Gupta, "Dealing with noise problem in machine learning data-sets: A systematic review," *Procedia Computer Science*, vol. 161, pp. 466–474, 2019.