IEEE Open Journal of the
Computer Society

Received 26 May 2025; revised 19 September 2025; accepted 25 September 2025. Date of publication 1 October 2025;
date of current version 17 October 2025. The review of this article was arranged by Associate Editor Peng Li.

Digital Object Identifier 10.1109/OJCS.2025.3616329

An Unsupervised Model Based on Knowledge Graph and Concepts for Sentiment Analysis

YONATAN MAMANI-COAQUIRA AND EDWIN VILLANUEVA

Engineering Department, Pontificia Universidad Católica del Perú, Lima 15088, Peru

CORRESPONDING AUTHOR: YONATAN MAMANI-COAQUIRA (e-mail: ymamanic@pucp.edu.pe).

This work was supported by the Fondo Nacional de Desarrollo Científico, Tecnológico y de Innovación Tecnológica (Fondecyt) of Peru.

ABSTRACT Sentiment analysis encompasses various fields such as psychology, marketing, and education, with social media serving as a key platform for gauging public opinion. Recently, graph-based methods have proven to be very useful in representing structured data. This study presents an unsupervised, graph knowledge approach to sentiment analysis that vectorizes nodes representing words and their conceptual connections. Using VADER (Valence Aware Dictionary and sentiment Reasoner) alongside conceptual words such as WordNet and ConceptNet, the method builds a graph of words based on sentiment polarity, capturing both co-occurrence and conceptual relationships. Additionally, a novel Polarity-biased Random Walk algorithm creates polarity-sensitive graph walks, which are vectorized using the Skip-Gram technique. The findings indicate that increasing walk length and the number of node walks, with a bias of 0.95 and employing ConceptNet or WordNet, enhances sentiment classification compared to models like Node2Vec, GraphSAGE, Graph Attention, and Graph Convolutional Networks. Lastly, embeddings generated from the IMDB dataset demonstrate superior accuracy in domain-specific tasks when compared to models such as Word2Vec, FastText, GloVe, and BERT.

INDEX TERMS Word embedding, sentiment analysis, polarity classification, knowledge graph, node embedding.

I. INTRODUCTION

Verbal language is a common form of human language that allows the expression and communication of information, ideas, emotions, and feelings. Today, there are several social media platforms that make it easier for people to express and disseminate verbal language. Companies and institutions constantly collect opinions and recommendations about their products or services, which constitutes a rich source of information to improve their offerings. [1], [2], [3]. In this context, Sentiment Analysis (SA) is an area of study that investigates methods and algorithms to understand the feelings expressed in verbal language, facilitating the analysis of large textual corpus and the decision-making process in different application fields such as political opinion, business, marketing, video preference, etc. [4].

The numerical representation of words is an important process during the development and implementation of machine learning models in the area of Natural Language Processing (NLP). When a word or phrase is optimally represented numerically, model building is facilitated and evaluation metrics yield better results. Word vectorization techniques are widely used in the SA area. The first techniques that emerged for this purpose were the Bag of Word (Bow) [5] and the Term Frequency-Inverse Document Frequency (TF-IDF) [6]. More recently, more sophisticated techniques have appeared, such as Word2Vec [7], Glove [8], FastText [9], ELMO [10] and BERT [11]. These techniques improve word vectorization in large amounts of data by approximating words by their semantic proximity to enable context understanding of sentences and to achieve better classifications of feelings and emotions.

In addition to traditional and graph-based approaches, recent studies have proposed novel techniques that further enrich sentiment analysis. For instance, ensemble-based models leveraging Transformers have achieved robust performance across multiple datasets [12]. Furthermore, heterogeneous ensemble frameworks have been introduced for aspect-based sentiment analysis (ABSA), enabling a finer granularity in sentiment detection [13]. Beyond sentence-level classification, other methods extend polarity detection to social media events, capturing broader collective dynamics [14]. At the same time, new embedding strategies specifically designed for sentiment tasks continue to be developed [15]. Finally, approaches that incorporate contextual cues, such as discourse markers and pragmatic features, have been demonstrated to significantly improve polarity classification [16]. These recent contributions highlight the importance of integrating ensemble architectures, contextual information, and domain-specific embeddings.

The aforementioned techniques have shown multiple benefits and have been applied in different SA studies. However, it is still necessary to further improve these techniques to achieve more accurate and explicable results [17]. Most of these techniques have been trained with millions of general domain texts with the intention of being usable in a wide variety of tasks, such as sentiment analysis, text translation, text summarization, question-and-answers, etc. However, several scholars have mentioned that word vectorization with a general approach may not be optimal in specialized domains and tasks because they do not use domain and task-specific information [18]. Most of these techniques are capable of understanding words syntactically and contextually. However, for a better understanding of the message expressed in the text, it is necessary to understand the words of the sentence conceptually. In this sense, some authors have mentioned that incorporating conceptual analysis into word representation can be an effective alternative to improve word representation and understanding [18], [19], [20], [21].

On the other hand, there has been a growing body of work in Artificial Intelligence (AI) focused on representing and analyzing structured information using graphs and graph theory. Some classical contributions in this area include DeepWalk [22], GraphSAGE [23], Node2Vec [24], and Graph2Vec [25], along with more recent deep learning architectures such as Graph Convolutional Networks (GCNs) [26] and Graph Attention Networks (GATs) [27], [28]. One of the main advantages of incorporating graph structures into modeling is their ability to more accurately capture contextual and semantic relationships. In particular, within the field of Natural Language Processing (NLP), several approaches have integrated graph structures in the form of knowledge graphs, showing potential to enhance performance across various NLP tasks [29], [30], [31], [32], [33], [34]. Specifically for sentiment analysis, we hypothesize that the introduction of knowledge graphs could improve the understanding of the meaning and subtle interactions among words, leading to better identification of the sentiment or emotion expressed in the texts. In this regard, the following research questions guide our study:

1) *RQ1:* How can conceptual knowledge (WordNet, ConceptNet) enrich word embeddings to capture semantic and polarity relationships more effectively?
2) *RQ2:* Does a polarity-biased random walk algorithm provide better sentiment-specific embeddings than traditional graph-based approaches?
3) *RQ3:* How does the proposed method compare with widely used pre-trained embeddings (Word2Vec, GloVe, FastText, BERT) across domain-specific and general-domain datasets?

This work presents a novel unsupervised method to generate word embeddings by combining knowledge graphs with conceptual analysis. Our approach involves constructing a word graph and incorporating semantic relationships and polarity information through biased random walks. The proposed technique aims to capture meaningful semantic associations such as synonyms and hyponyms along with polarity related features, to enrich embeddings and enhance sentiment polarity classification. The main contributions of this work include:

1) Enrichment of word embeddings by integrating conceptual knowledge to better capture semantic structure, concepts, and relationships between words, thus enhancing text comprehension.
2) Use of polarity-biased random walks to effectively group words sharing similar sentiment polarity, improving sentiment-specific representations.
3) A methodology for generating domain-specific embeddings, integrating both structural and semantic polarity information.
4) Integration of conceptual information to improve sentiment polarity classification performance.

The remainder of this article is structured as follows: Section II describes the proposed methodology. Section III outlines the experimental setup. Section IV presents the results and Section V the discussion. Finally, Section VI provides conclusions and future research directions.

II. PROPOSED METHODOLOGY

Fig. 1 shows the general workflow of the proposed knowledge graph-based approach for sentiment analysis. The process begins with data collection and pre-processing, involving the reading and cleaning of the data. Following this, new sentences are generated using WordNet and ConceptNet to improve the corpus with additional semantic relations. Subsequently, dictionaries of words classified by polarity (positive, negative, and neutral) are constructed using VADER (Valence Aware Dictionary and sEntiment Reasoner) [35], along with conceptual dictionaries based on WordNet and ConceptNet. Utilizing this information, a co-occurrence graph is generated, and random walks biased by word polarity are applied to create embeddings of more representative words. These embeddings are then utilized by various machine learning algorithms, such as Support Vector Machine (SVM), Random Forest (RF), XGBoost (XGB), Logistic Regression (LR), and K-nearest Neighbor (KNN), to perform sentiment classification, thereby improving the accuracy of predictive models.

A. DATA COLLECTION & PRE-PROCESSING

In this work, we utilized the IMDB movie dataset, which is commonly used for sentiment analysis. The dataset consists of movie reviews labeled with their corresponding polarity

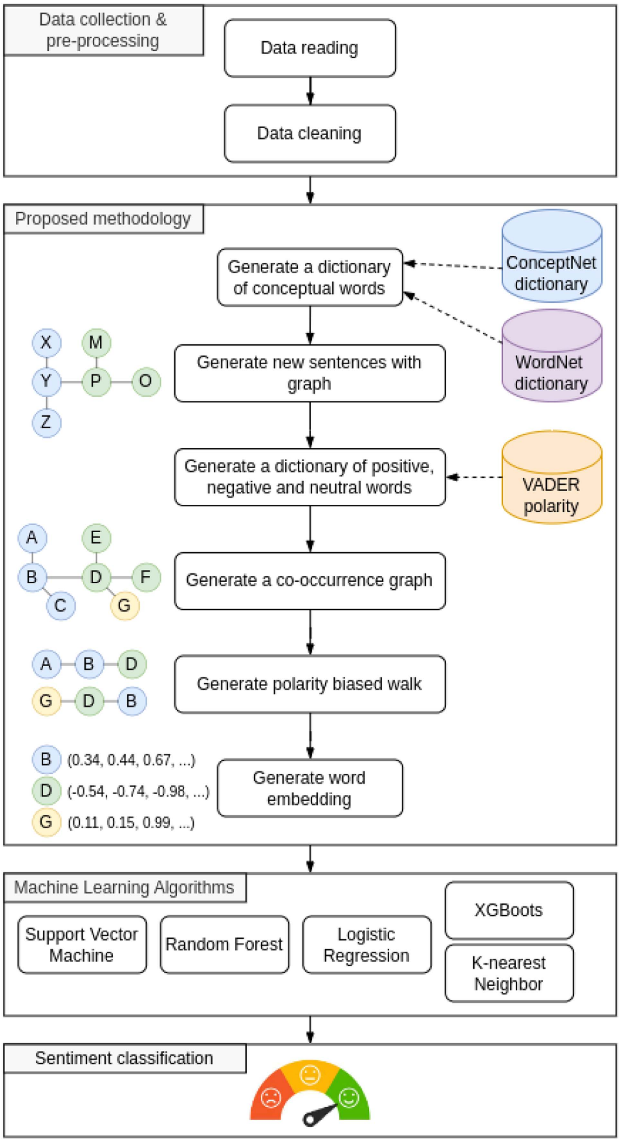

FIGURE 1. Overall workflow of the proposed methodology based on knowledge graph for sentiment analysis.

(positive or negative). To prepare the data for embedding generation, we implemented a data-cleaning algorithm to remove textual noise. We apply conventional proceses such as involved converting the text to lowercase, removing HTML tags, removing URLs, social media mentions, and special characters. We also remove square brackets and words longer than 20 characters and normalize white space. Following this, we applied lemmatization and removed a set of English stopwords to reduce the vocabulary size without losing the meaning essential for sentiment analysis [36].

To better understand the data, we clustered the IMDB reviews using TF-IDF and BERT embeddings followed by K-means. Fig. 2 shows a projection of the BERT embeddings onto the first two PCA components. This shows strong mixing of positive and negative reviews with no clear separation.

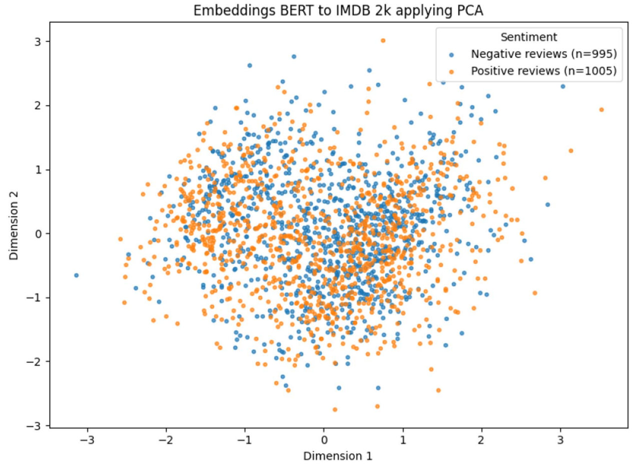

FIGURE 2. Scatter plot of the first two principal components (PCA) derived from BERT embeddings of the IMDB reviews. Colors indicate the sentiment labels.

Agreement between K-means clusters and sentiment labels is low: we found an Adjusted Rand Index (ARI) of 0.0217 and a Purity Score of 0.5765 using the TF-IDF embeddings, and an ARI of 0.0011 and Purity Score of 0.523 using the BERT embeddings [37]. These results indicate substantial lexical/semantic overlap between classes, underscoring the difficulty of sentiment classification in this domain.

B. GENERATE A DICTIONARY OF CONCEPTUAL WORDS

This phase consists of the creation of a dictionary that relates original words to their semantically similar concepts. Sources such as ConceptNet and WordNet are used to build this dictionary. By incorporating conceptual words, the corpus data is enhanced, resulting in a more varied representation of sentences.

Given a set of original words $O = \{o_1, o_2, \ldots, o_n\}$ that were extracted from the IMDB movie dataset and their associated concepts $C(o_i)$ obtained from sources such as ConceptNet or WordNet, the conceptual dictionary D is defined as:

$$D(o_i) = \{c_{i1}, c_{i2}, \ldots, c_{im}\} \quad \text{where} \quad c_{ij} \in C(o_i) \qquad (1)$$

C. GENERATE NEW SENTENCES WITH GRAPH

In this phase, the objective is to generate new sentences using the IMDB movie dataset. This will be done by using a graph knowledge-based approach and extracting concept words through the dictionary of conceptual words $D(o_i)$. Each word from the original sentence will be represented as a node in the graph, and additional nodes will be connected by edges to represent concept words.

1) GRAPH GENERATION

The graph $G_j = (V_j, E_j)$ is generated for each sentence o_j, where the nodes V_j represent both the original words of the sentence and their related concepts.

Given a sentence $o_j = \{w_1, w_2, \ldots, w_k\}$, the node set V_j of the graph consists of the original words w_i and their concepts $D(w_i) = \{c_{i1}, c_{i2}, \ldots, c_{im}\}$, where $c_{ij} \in C(w_i)$ is a concept associated with w_i. The node set V_j is defined as:

$$V_j = \{w_1, w_2, \ldots, w_k\} \cup D(w_i) \tag{2}$$

The edge set E_j contains two types of relationships:

- The connections between consecutive words in the sentence:

$$E_j = \{(w_i, w_{i+1}) \mid 1 \leq i < k\} \tag{3}$$

- The connections between each word w_i and its concepts $c_{ij} \in D(w_i)$:

$$E'_j = \{(w_i, c_{ij}) \mid c_{ij} \in D(w_i)\} \tag{4}$$

The resulting graph for each sentence is $G_j = (V_j, E_j \cup E'_j)$, representing the semantic structure of the sentence and its conceptual relationships.

2) GENERATION OF WORD-CONCEPT COMBINATIONS

The G_j represents a new combination of conceptual words for each sentence where for each word w_i, both the original word and its concepts $D(w_i)$ are considered, and all possible combinations of these words are generated.

The total set of possible combinations for a sentence o_j is defined as the cartesian product of the available options for each word w_i:

$$\mathcal{C}(o_j) = \prod_{i=1}^{k} (\{w_i\} \cup D(w_i)) \tag{5}$$

Each original word w_i can be replaced by any of its concepts $c_{ij} \in D(w_i)$. Likewise, the number of combinations is limited to a maximum $\mathcal{C}_{\text{selected}}$, which represents the total number of sentences selected from a random subset of combinations made in $\mathcal{C}(o_j)$.

3) GENERATION OF NEW SENTENCES

The selected combinations $\mathcal{C}_{\text{selected}}$ are used to construct new sentences. Each combination of $\mathcal{C}_{\text{selected}}$ results in a new sentence where some original words have been replaced by their concepts.

For each combination $c \in \mathcal{C}_{\text{selected}}$, a new sentence o_j^{new} is generated as a sequence of words w'_i, where $w'_i \in \{w_i\} \cup D(w_i)$. The new sentence is given by:

$$o_j^{\text{new}} = (w'_1, w'_2, \ldots, w'_k) \tag{6}$$

The total set of new sentences generated from the original sentence o_j is denoted as:

$$\mathcal{O}_j^{\text{new}} = \{o_j^{\text{new}} \mid o_j^{\text{new}} \in \mathcal{C}_{\text{selected}}\} \tag{7}$$

Finally, the set of all new sentences generated for the entire dataset is where N is the total number of sentences in the dataset.

$$\mathcal{O}^{\text{new}} = \bigcup_{j=1}^{N} \mathcal{O}_j^{\text{new}} \tag{8}$$

D. GENERATE A DICTIONARY OF POLARITY WORDS

In this phase, we create a word polarity dictionary using the VADER lexicon. This dictionary classifies words as positive, negative, or neutral by evaluating each word's score in the VADER lexicon based on three polarity categories. This dictionary facilitates the process of labeling words based on their sentiment in sentiment analysis.

For each word $w \in$ VADER lexicon, the polarity is assigned as follows:

$$W_{\text{polarity}} = \begin{cases} 1 & \text{if score(w)} \geq 0.05 \quad \text{(Positive)} \\ -1 & \text{if score(w)} \leq -0.05 \quad \text{(Negative)} \\ 0 & \text{if } -0.05 < \text{score(w)} < 0.05 \quad \text{(Neutral)} \end{cases} \tag{9}$$

The polarity dictionary $\mathcal{D}_{\text{polarity}}$ maps each word w to its corresponding polarity W_{polarity}.

E. GENERATE A CO-OCCURRENCE GRAPH

In this phase, a graph is created to represent co-occurrence relationships and conceptual word connections based on polarity. For every sequence of tokens in a sentence, the graph identifies the words present in the vocabulary (tokens) and establishes connections (edges) between words that co-occur within a specific context window. Furthermore, the conceptual dictionary adds new nodes and edges for vocabulary words only if these conceptual words are part of the polarity dictionary.

Given a document represented as a sequence of words $T = \{w_1, w_2, \ldots, w_n\}$, the set of nodes N consists of the words appearing in the document $N = \{w_1, w_2, \ldots, w_n\}$. Each word $w_i \in T$ is represented as a node in the graph.

For each word $w_i \in N$, a context window of size k is defined around w_i, meaning that w_i is connected to the words w_j that are within k positions to the left or right. The co-occurrence edges are defined as:

$$E_{\text{co-occurrence}} = \{(w_i, w_j) \mid |i - j| \leq k, i \neq j, w_i, w_j \in N\} \tag{10}$$

The conceptual dictionary $D(w_i)$ contains the concepts of each word w_i. For each word w_i, its concepts are retrieved, and if a concept $c \in C(w_i)$ belongs to the set of polarity words W_{polarity}, a semantic edge is added between w_i and c.

The semantic edges are defined as:

$$E_{\text{semantic}} = \{(w_i, c) \mid c \in D(w_i), c \in W_{\text{polarity}}\} \tag{11}$$

Graph result G is an undirected graph composed of the node set N and the edge set E, which includes both co-occurrence and semantic edges:

$$G = (N, E), \quad \text{where} \quad E = E_{\text{co-occurrence}} \cup E_{\text{semantic}} \tag{12}$$

F. GENERATE POLARITY-BIASED RANDOM WALKS

The polarity-biased random walk algorithm (see Algorithm 1) is created to navigate a graph of word co-occurrences. In this graph, nodes represent words or concepts, and edges represent their semantic or co-occurrence relationships. The algorithm prefers transitions between nodes that have the same sentiment polarity. It uses a bias parameter denoted as b to increase the likelihood of moving towards neighbouring nodes with the same polarity (positive, neutral, or negative). This approach helps to maintain polarity consistency in the generated walks, which is beneficial for developing polarity-sensitive representations in sentiment analysis tasks, such as creating word embeddings.

1) POLARITY-BIASED RANDOM WALK INITIALIZATION

Let $G = (V, E)$ be an undirected graph, where V is the set of nodes (representing words). E is the set of edges, representing relationships between nodes (such as co-occurrence or concept relationships). Each node $v \in V$ is associated with a polarity value $p(v) \in \{-1, 0, 1\}$, where:

- $p(v) = 1$ for positive words.
- $p(v) = -1$ for negative words.
- $p(v) = 0$ for neutral words or words without assigned polarity.

A polarity-biased random walk starting from node v_0 is represented as a sequence:

$$W = \{v_0, v_1, v_2, \ldots, v_{L-1}\} \tag{13}$$

where v_0 is the starting node, and L is the total length of the walk.

2) NODE NEIGHBORHOOD

For get node Neighborhood in each node v_i in the walk, the set of neighbors $N(v_i)$ is defined as:

$$N(v_i) = \{v_j \in V \mid (v_i, v_j) \in E\} \tag{14}$$

This set contains all nodes v_j that are connected to v_i by an edge in E.

3) TRANSITION PROBABILITIES

On the other hand, the transition probability from node v_i to a neighboring node $v_j \in N(v_i)$ is determined by their respective polarity values. The probability is defined as:

$$P(v_j \mid v_i) = \begin{cases} b & \text{if } p(v_i) = p(v_j) \text{ and } p(v_i) \neq 0 \\ 1 - b & \text{if } p(v_i) \neq p(v_j) \text{ or } p(v_i) = 0 \end{cases} \tag{15}$$

where b is a bias parameter that gives higher transition probabilities to nodes with the same polarity as v_i. If the polarities of v_i and v_j are equal and non-neutral, the transition probability is high (b). If the polarities differ or are neutral, the transition probability is lower $(1 - b)$.

After, transition probabilities are normalized to ensure they sum to 1. For each node v_i, the sum of the transition probabilities to its neighbors $v_j \in N(v_i)$ is:

$$\sum_{v_j \in N(v_i)} P(v_j \mid v_i) = 1 \tag{16}$$

4) SELECTION OF THE NEXT NODE

For selection of the next node in the walk, v_{i+1}, is selected from the set of neighbors $N(v_i)$ based on the normalized transition probabilities:

$$v_{i+1} = \arg\max_{v_j \in N(v_i)} P(v_j \mid v_i) \tag{17}$$

This selection process ensures that nodes with similar polarities are more likely to be chosen during the walk.

5) RANDOM WALK GENERATION

The walk $W = \{v_0, v_1, \ldots, v_{L-1}\}$ continues until it reaches the maximum length L(walk length), or terminates early if the current node has no neighbors. After, is necesaria the create multiple walks for each node $v \in V$, multiple independent polarity-biased random walks are generated. The number of steps is defined by n(num walks by node), leading to the generation of a set of steps:

$$\mathcal{W} = \bigcup_{v \in V} \{W_1(v), W_2(v), \ldots, W_n(v)\} \tag{18}$$

Where $W_i(v)$ represents the i step starting from node v.

G. GENERATE WORD EMBEDDING

To convert the network words into embeddings, we first transform each node of the walks into a text string. Then, we process these strings using the Skip-Gram [7] learning model, which uses the sequence of paths to create vector representations. The model (Skip-Gram) is presented as follows.

The objective is to predict context words given a central target word. The model aims to maximize the probability of correctly predicting nearby words w_{t+j} given a central word w_t, within a context window of size c. This technique is particularly useful for capturing semantic and co-occurrence relationships between words in a corpus.

Given a sequence of words $w_1, w_2, \ldots, w_T$, the objective of the Skip-gram model is to maximize the probability of predicting the nearby words w_{t+j} for a central word w_t, within a context window of size c. This is expressed as:

$$\prod_{t=1}^{T} \prod_{-c \leq j \leq c, j \neq 0} P(w_{t+j} \mid w_t) \tag{19}$$

To optimize this model, we maximize the log-likelihood of the joint probability:

$$\frac{1}{T} \sum_{t=1}^{T} \sum_{-c \leq j \leq c, j \neq 0} \log P(w_{t+j} \mid w_t) \tag{20}$$

Algorithm 1: Polarity-Biased Random Walk Algorithm.

Input: Graph $G = (V, E)$, Walks by node n, Walk length l, Bias b
Output: List of polarity-biased walks

LearnFeatures (Graph $G = (V, E)$, Walks by node n, Walk length l, Bias b):
Initialize walks as an empty list;
for *each node* $u \in V$ **do**
  for *iter = 1 to* n **do**
    walk = **PolarityBiasedWalk**(G, u, l, b);
    Append walk to walks;
return walks;

PolarityBiasedWalk (Graph $G = (V, E)$, Start node u, Length l, Bias b):
Initialize walk = $[u]$;
for *walk_iter = 1 to* $l - 1$ **do**
  curr_node = walk[-1];
  neighbors = **GetNeighbors**(curr_node, G);
  if *neighbors is empty* **then**
    break;
  Initialize probabilities as an empty list;
  for *each neighbor in neighbors* **do**
    curr_polarity = **GetPolarity**(curr_node);
    neighbor_polarity = **GetPolarity**(neighbor);
    if *curr_polarity = neighbor_polarity and curr_polarity* $\neq 0$ **then**
      probability = b;
    else
      probability = $1 - b$;
    Append probability to probabilities;
  Normalize probabilities so they sum to 1;

$$\sum_{w_j \in N(\text{curr\_node})} P(neighbor_i \mid curr\_node) = 1$$

  Select random next_node from neighbors based on the normalized probabilities:

$$\text{next\_node} = \arg \max_{w_j \in N(\text{curr\_node})} P(w_j \mid curr\_node)$$

  Append next_node to walk;
return walk;

The conditional probability $P(w_{t+j} \mid w_t)$ is modeled using the softmax function, which assigns a probability to each word in the vocabulary based on the vector representations v_{w_t} of the words. This is represented as:

$$P(w_{t+j} \mid w_t) = \frac{\exp(v_{w_{t+j}}^{\top} v_{w_t})}{\sum_{w_k \in V} \exp(v_{w_k}^{\top} v_{w_t})} \tag{21}$$

Where:
- v_{w_t} is the vector representation of the central word w_t.
- $v_{w_{t+j}}$ is the vector representation of the context word.
- V is the set of all words in the vocabulary.
- $\exp(\cdot)$ denotes the exponential function.

H. MACHINE LEARNING ALGORITHMS

In this phase, we apply several machine learning algorithms to classify sentiment using word embeddings generated in the previous stage. The algorithms we use include SVM, RF, LR, XGBoost, and KNN. We use word embeddings as input features to train classifiers that can predict sentiment polarity (positive, negative, or neutral). These models are trained to maximize accuracy and generalization in sentiment classification.

I. SENTIMENT CLASSIFICATION

The sentiment classification phase involves evaluating the performance of the machine learning models. Two key evaluation metrics used in this phase are accuracy and F1-score.

Accuracy measures the proportion of correct predictions (positive and negative) out of the total number of instances evaluated. It's defined as:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Where:
- TP (True Positives) are the correctly predicted positive instances.
- TN (True Negatives) are the correctly predicted negative instances.
- FP (False Positives) are the instances incorrectly predicted as positive.
- FN (False Negatives) are the instances incorrectly predicted as negative.

The F1-score is the harmonic mean of precision and recall, and it is especially useful when there is an imbalance between classes. It is defined as:

$$\text{F1-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{22}$$

Where precision and recall are represented as follows:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{23}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{24}$$

III. EXPERIMENTS

In this section, we conducted experiments using a dataset related to polarity classification to verify the performance of the proposed methodology. The experiment scripts to reproduce our results are publicly available at https://github.com/yonatanmc/sentinode3.

A. DATASET

The dataset chosen for building word embedding, generating pretraining models, and comparison includes IMDB movie reviews with both negative and positive polarity (see Table 1).

TABLE 1. Number of Sentences by Dataset

Dataset	Positive sentences	Negative sentences	Total sentences
IMDB movie	1005	995	2000
SST2 movie	4522	4219	8741
McDonal's	2693	2307	5000

TABLE 2. Number of Nodes and Edges of IMDB Movie

Dataset	Sentences	Concepts	Nodes	Edges
IMDB movie	2000	-	25007	332493
IMDB movie	2000	WordNet	25265	348563
IMDB movie	2000	ConceptNet	25007	369296

It will be compared with Node2Vec, GraphSAGE, GAT, and GCN. After using pretraining models to compare the specific domain SST2 dataset (movie) and general domain McDonald's review (opinion on fast food) (see Table 1), both with negative and positive polarities, For this comparison, apply models such as Word2vec, Glove, FastText, and BERT, which are commonly used for text encoding for sentiment analysis. On the other hand, to create a polarity and conceptual dictionary, we use VADER for polarity and WordNet and ConceptNet for conceptual relation (see Table 2).

B. EVALUATION METRICS

The proposed word embedding models were evaluated using various metrics to validate their effectiveness. Spearman and Pearson correlation coefficients, along with their respective p-values, were used to assess how well the embeddings captured human-annotated word similarity using the WordSim353 and SimLex999 datasets. Additionally, polarity-based word similarity metrics (Positive-Positive, Negative-Negative, and Positive-Negative Similarity) were used to evaluate how well the embeddings captured sentiment distinctions. For the sentiment classification task, t-Distributed Stochastic Neighbor Embedding (t-SNE) was used to visualize the embeddings, identifying clusters of words based on polarity (positive and negative). Finally, accuracy and F1-score were applied to measure the model's overall performance [38], [39], [40].

IV. RESULTS

A. ACCURACY AND F1-SCORE

To evaluate these results with the proposed methodology, we considered the parameters of walk length (wl), number of walks by node (nw), and bias (b). This allowed for random path exploration with a bias to find better representations of correlated words. Subsequently, the generated embeddings were used in sentiment polarity classification using SVM, RF, LR, XGB, and KNN algorithms. Using the metrics of accuracy and f1-score, we were able to verify that the parameters of the proposal demonstrated better performance in polarity classification.

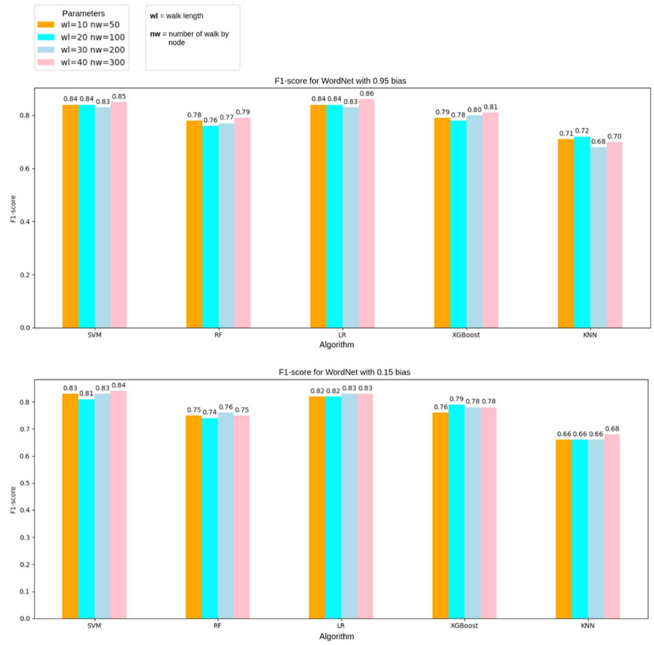

FIGURE 3. Results of F1-score using WordNet with bias 0.95 and 0.15.

1) RESULT OF CONCEPTNET AND WORDNET WITH F1-SCORE

Fig. 3 shows the impact of the depth of the walk and the number of walks per node when using WordNet with two different biases, 0.95 and 0.15. Fig. 3, which has a high bias towards the same polarity (0.95), shows superior performance. The best F1-scores are achieved with SVM (0.85) and LR (0.86) when using $wl = 40$ and $nw = 300$. On the other hand, the Fig. 3, with a low bias (0.15), shows slightly lower performance, with SVM and LR reaching approximately 0.84 and 0.83, respectively, under the same parameter conditions. Overall, the best results are obtained in WordNet with a bias of 0.95 using the SVM and LR algorithms with the greatest walk depth and the highest number of walks by node.

Fig. 4 shows the results of the *F1-score* according to the walk length and the number of walks by node using ConceptNet with two different biases of 0.95 and 0.15. In the Figure with a high bias of 0.95, the best *F1-score* is achieved with SVM (0.84) and LR (0.83) when using $wl = 40$ and $nw = 300$. On the other hand, in Fig. 4 with a low bias of 0.15, the results are slightly lower, with SVM reaching an *F1-score* of 0.81 and LR 0.82 under the same conditions. Similar to evaluating the high *bias*, the results improve as the number of paths and walk depth increases. The overall best performance is obtained in Fig. 4 with the SVM and LR algorithms using the greatest walk depth and the highest number of walks by node.

2) RESULTS FOR IMDB DATASET USING GRAPH KNOWLEDGE

Tables 3 and 4 shows the performance results of different classification algorithms when applying graph knowledge for sentiment classification on the IMDB movie dataset. Six models were evaluated: Node2Vec, GraphSAGE, GAT, GCN, and

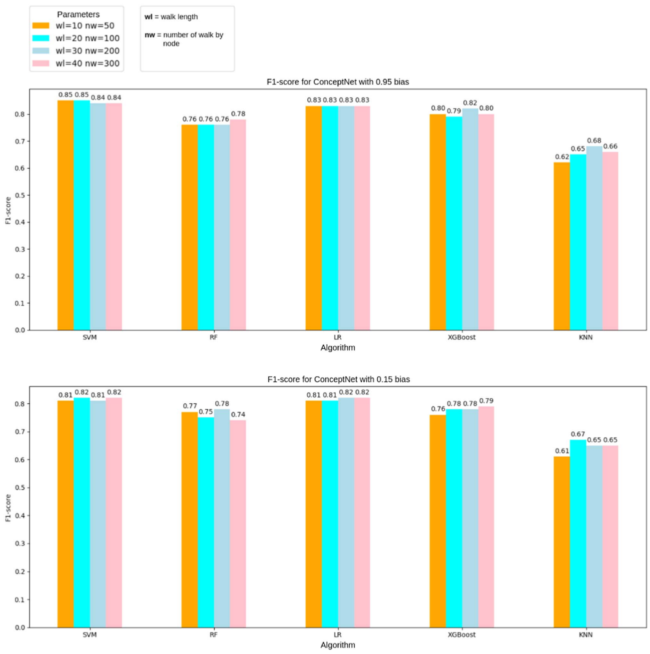

FIGURE 4. Results of F1-score using ConceptNet with bias 0.95 and 0.15.

TABLE 3. Results of Accuracy With Graphs for IMDB Movie Dataset

Model	SVM	RF	LR	XGBoots	KNN
Node2vec	0.81	0.71	0.79	0.77	0.63
GraphSAGE	0.60	0.69	0.65	0.68	0.61
GAT	0.70	0.71	0.71	0.72	**0.73**
GCN	0.68	0.69	0.69	0.67	0.67
Proposal (wordnet)	**0.85**	**0.79**	**0.85**	**0.81**	0.71
Proposal (conceptnet)	0.82	0.78	0.83	0.78	0.71

TABLE 4. Results of F1-Score With Graphs for IMDB Movie Dataset

Model	SVM	RF	LR	XGBoots	KNN
Node2vec	0.82	0.72	0.80	0.77	0.61
GraphSAGE	0.65	0.71	0.66	0.70	0.60
GAT	0.69	0.70	0.70	0.71	**0.73**
GCN	0.66	0.68	0.69	0.67	0.64
Proposal (wordnet)	**0.85**	**0.79**	**0.86**	**0.81**	0.69
Proposal (conceptnet)	0.82	0.78	0.83	0.78	0.68

the proposals using WordNet and ConceptNet. For the proposals, the optimal parameters identified earlier were used: walk length ($wl = 40$), number of paths by node ($nw = 300$), and a bias ($b = 0.95$), which had previously proven to be the most effective in evaluations.

During the evaluation of accuracy, the best results are achieved by the proposal based on ConceptNet, obtaining values close to 0.85 in SVM, RF, and LR, which suggests that the embeddings generated from ConceptNet effectively capture semantic and polarity relationships in the text. The WordNet-based proposal has values close to those of ConceptNet, also

TABLE 5. The Result of F1-Score for IMDB Using Pre-Trained Models

Model	SVM	RF	LR	XGBoots	KNN
Word2vec	0.83	0.79	0.80	0.79	**0.76**
Glove	0.80	0.79	0.81	0.78	0.75
FastText	0.84	**0.81**	0.80	**0.83**	0.72
Bert	0.79	0.74	0.78	0.75	0.62
Proposal (wordnet)	**0.85**	0.79	**0.86**	0.81	0.69
Proposal (conceptnet)	0.82	0.78	0.83	0.78	0.68

achieving good accuracy results across all algorithms. In comparison, the models utilizing Node2Vec, GraphSAGE, and GAT demonstrate lower performance, with accuracy ranging from 0.60 to 0.73, indicating that these methods fail to capture contextual information as efficiently as the proposals based on WordNet and ConceptNet. On the other hand, the F1-score reveals a similar pattern, with the ConceptNet proposal achieving the highest values in nearly all models, especially in SVM and RF, with values of 0.85 and 0.79, respectively. The WordNet proposal also demonstrates good performance, outperforming traditional methods such as Node2Vec, GraphSAGE, and GAT in all cases. This highlights that the use of additional semantic relationships provided by ConceptNet and WordNet is key to improving sentiment classification, as it reinforces connections between nodes of similar polarity, which is crucial in the context of sentiment analysis.

These results confirm that proposals based on WordNet and ConceptNet, utilizing random walks with $b = 0.95$, more accurately capture semantic relationships and polarity.

3) RESULT OF F1-SCORE FOR IMDB USING PRE-TRAINED MODELS

Table 5 shows the F1-score results on the IMDB dataset using pre-trained embedding models. Although models such as Word2vec, GloVe, FastText and BERT were trained with millions of domain-general words, the proposed models (WordNet and ConceptNet) show competitive performance. Specifically, the WordNet-based proposal achieves the best performance with SVM (0.85) and LR (0.86), surpassing models such as BERT and FastText in certain cases. Similarly, the RF and XGBoost algorithms with FastText outperform the proposal, but the difference is minimal. This suggests that, in some cases, incorporating domain-specific and graph knowledge can outperform models trained on general-domain data.

B. SPEARMAN AND PEARSON CORRELATION

Tables 6 and 7 show the correlation results of different embedding models on the WordSim-353 and SimLex-999 datasets. Traditional models like Word2vec, GloVe, and FastText demonstrate low correlations in both Spearman and Pearson, indicating that they don't effectively capture semantic similarities. Additionally, the high p-values suggest low statistical significance. On the other hand, BERT performs significantly better in both datasets, especially in

TABLE 6. Results of Correlation for WordSim-353

Model	Spearman corr.	Spearman p-value	Pearson corr.	Pearson p-value
Word2vec[1]	0.03	0.03	0.03	0.91
Glove[1]	0.07	0.37	0.12	0.12
FastText[1]	-0.01	0.83	0.02	0.79
Bert[1]	0.33	3.57E-08	0.33	3.82E-08
Proposal(wordnet)[2]	0.30	6.55E-07	0.43	2.85E-13
Proposal(conceptnet)[2]	0.43	7.54E-16	0.51	7.85E-22

TABLE 7. Results of Correlation for SimLex-999

Model	Spearman corr.	Spearman p-value	Pearson corr.	Pearson p-value
Word2vec[1]	0.03	0.40	0.08	0.04
Glove[1]	0.01	0.74	0.01	0.88
FastText[1]	0.06	0.09	0.04	0.18
Bert[1]	0.17	4.39E-07	0.15	1.00E-05
Proposal(wordnet)[2]	0.22	1.14E-11	0.27	7.74E-17
Proposal(conceptnet)[2]	0.41	4.20E-40	0.41	1.34E-40

TABLE 8. Results of Word Similarity by Polarity

Model	Positive-Positive	Negative-Negative	Positive-Negative
Word2vec[1]	0.91	0.91	0.90
Glove[1]	0.14	0.15	0.11
FastText[1]	0.86	0.96	0.81
Bert[1]	0.82	0.86	0.81
Proposal(wordnet)[2]	0.20	0.20	0.11
Proposal(conceptnet)[2]	0.16	0.16	0.11

WordSim-353, with Spearman and Pearson correlations of 0.33, showcasing its ability to capture semantic relationships. However, its performance decreases in SimLex-999. The WordNet and ConceptNet-based models, particularly the ConceptNet model, exhibit notable performance, achieving the highest correlations in both datasets. The ConceptNet model shows a Spearman correlation of 0.43 in WordSim-353 and 0.41 in SimLex-999, along with very low p-values, indicating a strong relationship between embeddings and human similarities. These results highlight the effectiveness of conceptual graph-based models in capturing semantic relations and similarity, which is beneficial for sentiment analysis tasks. To ensure comparability between models, the same dimensions were employed for training from scratch, as demonstrated in Tables 6, 7, and 8. Models referenced in (1) utilized these algorithms to generate our embeddings with the IMDB dataset at a dimension of 300. Proposals referenced in (2) for embeddings based on WordNet and ConceptNet were also generated with a dimension of 300, using optimal parameters of $wl = 40$, $nw = 300$, and $b = 0.95$ as identified previously.

TABLE 9. Results of Accuracy for SST2 Dataset

Model	SVM	RF	LR	XGBoots	KNN
Word2vec	0.62	0.64	0.66	0.63	0.58
Glove	0.59	0.67	0.68	0.67	0.59
FastText	0.69	0.66	0.68	0.65	0.60
Bert	0.73	0.67	0.73	0.70	0.62
Proposal (wordnet)	0.77	0.70	**0.76**	0.72	0.66
Proposal (conceptnet)	**0.78**	**0.71**	**0.76**	**0.74**	**0.68**

TABLE 10. Results of F1-Score for SST2 Dataset

Model	SVM	RF	LR	XGBoots	KNN
Word2vec	0.69	0.65	0.68	0.65	0.61
Glove	0.69	0.68	0.70	0.68	0.62
FastText	0.71	0.67	0.70	0.67	0.62
Bert	0.74	0.69	0.74	0.71	0.67
Proposal (wordnet)	0.78	0.71	**0.77**	0.73	**0.70**
Proposal (conceptnet)	**0.79**	**0.72**	**0.77**	**0.75**	**0.70**

C. POLARITY BASED WORD SIMILARITY

For the evaluation, the following group of words was considered.

- Positive words: happy, joy, love, good, excellent, well, beautiful, honestly, amazing, fantastic, wonderful, delightful, glad, great, awesome, positive, perfect, smile, exciting, friendly.
- Negative words: sad, anger, hate, bad, terrible, stupid, serious, afraid, horrible, awful, disappointing, disgusting, frustrated, nasty, pain, worst, miserable, annoyed, bored, foolish.

Table 8 evaluates word similarity based on polarity (Positive-Positive, Negative-Negative, and Positive-Negative) for various embedding models. Word2Vec, FastText, and BERT have achieved the best results, with FastText particularly excelling at capturing relationships within the same polarity (0.86 for Positive-Positive and 0.96 for Negative-Negative). BERT also performs well in the Positive-Negative category (0.81), demonstrating its ability to differentiate between opposing polarities. However, the proposed models based on WordNet and ConceptNet show lower performance across all categories, suggesting that while they are effective at capturing conceptual relationships, they are not as precise at distinguishing words based on polarity.

On the other hand, the model based on GloVe and the proposal with WordNet and ConceptNet shows better differentiation between words of opposite polarities (Positive-Negative), which is a good indication for finding better results for classification in sentiment analysis. However, both models have issues capturing relationships within the same polarities, especially the proposals using graphs.

We can observe an apparent tension between word-level polarity similarity (see Table 8) and sentence-level classification performance (see Tables 3–4 and 9 –10). This is not a

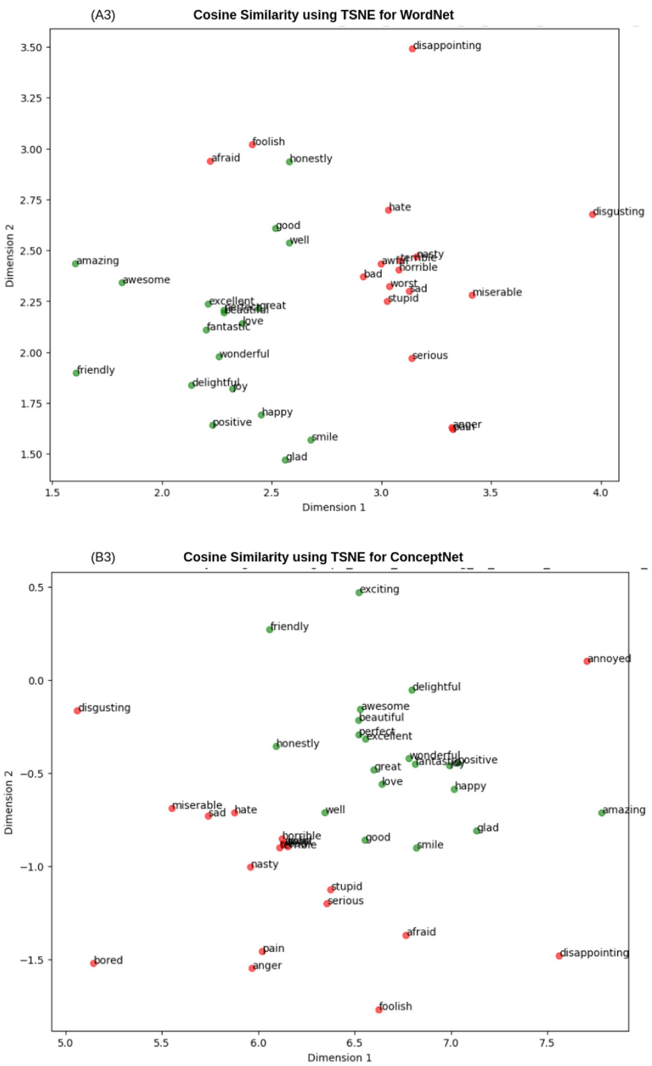

FIGURE 5. Cosine similarity using t-SNE for WordNet and ConceptNet. Red color (negative words) and green (positive words).

contradiction but a difference in evaluation objectives. Intrinsic metrics quantify local cosine proximity between isolated tokens, whereas our downstream task relies on compositional sentence representations and a supervised classifier that exploits multi-token patterns (e.g., negation, contrast, discourse markers). Consequently, a model may obtain modest scores on word-level polarity similarity yet perform strongly in sentence classification, because the latter benefits from multi-word interactions and sequence patterns induced by polarity-biased random walks.

D. T-SNE POLARITY VISUALIZATION

Fig. 5 show word representations using t-SNE with cosine similarity for embeddings generated from WordNet (A3) and ConceptNet (B3). In the WordNet (A3), we can observe better clustering of positive words (green) such as amazing, awesome, fantastic, and wonderful, which form coherent clusters. Additionally, negative words (red) such as hate, sad, and miserable also tend to cluster correctly, but are sparsely scattered.

TABLE 11. Results of Accuracy for McDonals Dataset

Model	SVM	RF	LR	XGBoots	KNN
Word2vec	0.81	0.84	**0.87**	0.85	**0.83**
Glove	**0.85**	**0.85**	0.86	0.86	0.82
FastText	**0.85**	**0.85**	0.85	**0.87**	0.82
Bert	0.84	0.83	0.84	0.84	**0.83**
Proposal (wordnet)	0.80	0.82	0.84	0.84	0.80
Proposal (conceptnet)	0.80	0.82	0.84	0.84	0.78

On the other hand, in ConceptNet (B3), the positive words maintain coherence, and only the negative word annoyed is highly dispersed. It seems that the WordNet model captures semantic relationships within the polarities better, while the ConceptNet model shows some scatter, indicating that it does not clearly capture the relationship with some words. In terms of polarity differentiation, the WordNet graph presents a more distinct separation between positive and negative words, making it more suitable compared to the ConceptNet graph.

E. RESULTS OF ACCURACY AND F1-SCORE FOR SPECIFIC AND GENERAL DOMAIN

To evaluate the SST2 dataset, the proposals from WordNet and ConceptNet were generated with optimal parameters of $wl = 40$, $nw = 300$, and $b = 0.95$, using a dimension of 300 for the embeddings, which allows for a fair comparison with the embedding models generated from the IMDB dataset using Word2Vec, GloVe, FastText, and BERT models with 300 dimensions. On the other hand, to evaluate the McDonald's dataset (food domain), general domain pre-trained models were used, except for the proposals with WordNet and ConceptNet, which pertain to the movie domain.

1) RESULTS FOR THE SST-2 DATASET IN A SPECIFIC DOMAIN

The results in Tables 9 and 10 show the Accuracy and F1-score values for the SST2 movie dataset (specific domain). The results are based on 300-dimensional embeddings created from the IMDB dataset using the Word2vec, GloVe, FastText, and BERT models. It's evident that the proposed models with WordNet and ConceptNet consistently outperform the pre-trained embeddings in terms of accuracy and F1-score across most algorithms. Specifically, the ConceptNet-based proposal achieves the best results with SVM (0.78) and LR (0.76) in accuracy and also excels in F1-score, reaching 0.79 with SVM and 0.77 with LR. While pre-trained models such as BERT and FastText deliver good results, particularly in accuracy, they don't surpass the proposed models. This indicates that embeddings based on graph knowledge, such as WordNet and ConceptNet, are better suited for capturing semantic and sentiment relationships in the specific domain.

2) RESULTS FROM THE MCDONALDS DATASET APPLIED TO A GENERAL DOMAIN

The accuracy and F1-score results on the McDonalds dataset are presented in Tables 11 and 12. It's important to note that the proposed models trained on the IMDB dataset WordNet

TABLE 12. Results of F1-Score for McDonals Dataset

Model	SVM	RF	LR	XGBoots	KNN
Word2vec	0.80	**0.85**	**0.87**	0.86	0.83
Glove	**0.85**	**0.85**	0.86	0.86	0.82
FastText	**0.85**	**0.85**	0.85	**0.87**	0.82
Bert	**0.85**	0.83	0.85	0.85	**0.84**
Proposal (wordnet)	0.79	0.82	0.84	0.84	0.79
Proposal (conceptnet)	0.78	0.82	0.84	0.84	0.78

and ConceptNet aren't from the same domain as McDonalds (fast food), while models such as Word2vec, GloVe, FastText, and BERT are domain-general. In terms of accuracy, GloVe and FastText demonstrate superior performance across most classifiers, with GloVe achieving 0.85 with SVM and FastText achieving 0.87 with XGBoost. Similarly, in terms of F1-score, GloVe and FastText also deliver the best performances, with GloVe reaching 0.85 with SVM and RF, and FastText reaching 0.87 with XGBoost and LR with 0.87 using Word2vec. On the other hand, models based on WordNet and ConceptNet, while achieving acceptable F1-score values, exhibit inferior performance, indicating that embeddings trained on a specific domain such as IMDB do not transfer optimally to another domain such as McDonalds. This underscores the importance of training embeddings with domain-specific data for improved results in sentiment analysis.

V. DISCUSSION

On domain-specific datasets (with IMDB used to induce embeddings and SST-2 for evaluation), the proposed concept-enriched embeddings built from WordNet/ConceptNet and trained on polarity-biased random walks consistently outperform general-purpose baselines (Word2Vec, GloVe, FastText, BERT) across most classifiers. In particular, linear models (e.g., SVM and Logistic Regression) benefit from representations that encode lexical-semantic relations alongside sentiment cues, yielding higher Accuracy and F1. These gains align with the experimental configuration that proved most effective for embedding induction (dimension $= 300$, walk length $wl = 40$, walks per node $nw = 300$, bias $b = 0.95$), and support the claim that integrating conceptual knowledge with polarity structure is advantageous when the downstream task and source domain are well aligned (see Section IV).

From a mechanistic standpoint, qualitative analyses (e.g., t-SNE) indicate that the lexical graph particularly when instantiated with WordNet organises polarity-aware neighbourhoods more distinctly, showing clearer separation between positive and negative regions and tighter intra-polarity clusters than the ConceptNet counterpart. This observation is consistent with the role of a high polarity bias (b) during random walks, which increases the likelihood of traversing sentiment homogeneous regions, and with the contribution of semantic edges (e.g., synonymy, hyponymy) that densify paths among semantically related nodes. Collectively, these factors help explain why the proposed embeddings better capture sentiment relevant structure for polarity classification in domain-constrained scenarios. On the other hand, our approach depends on lexical resources (WordNet/ConceptNet) and fixed VADER thresholds, which may fail to cover slang and neologisms and can mis estimate polarity across domains. Moreover, conceptual expansion and graph modelling can introduce semantic drift unless relation types are carefully filtered and edges are weighted by reliability or relation type. Finally, using type level word embeddings without sense disambiguation prevents the model from fully capturing polysemy and pragmatic cues (e.g., negation, intensification, irony), thereby limiting polarity detection accuracy.

VI. CONCLUSION

This study proposed a novel graph-based approach to sentiment analysis by integrating semantic and polarity relationships through lexical resources such as WordNet and ConceptNet. The developed methodology enables the generation of enriched vector representations using a polarity-biased random walk algorithm, combined with the Skip-Gram model to generate embeddings. These embeddings effectively capture both semantic and sentiment-related relationships, enhancing performance in sentiment classification tasks.

Results obtained on the IMDB dataset demonstrate that embeddings based on ConceptNet outperform widely-used pre-trained models such as Word2Vec, GloVe, FastText, and BERT in key metrics like accuracy and F1-score, particularly in domain-specific applications such as movie reviews. Moreover, the proposed approach significantly outperforms traditional graph-based models, including Node2Vec, GraphSAGE, GAT, and GCN, highlighting the added value of incorporating conceptual knowledge and polarity bias into the embedding generation process.

In general-domain contexts, such as the McDonald's review dataset, general-purpose pre-trained models achieved superior performance, emphasizing the importance of tailoring embeddings to the target domain to optimize results.

These findings validate the potential of concept-enriched graph-based representations as an effective alternative for sentiment analysis in specialized domains and highlight the opportunity to develop more interpretable and domain-aware models.

Future research will focus on developing new graph-based embedding models that dynamically integrate semantic and contextual polarity information by leveraging advanced architectures such as Graph Neural Networks (GNNs) and Graph Attention Networks (GATs). Additionally, we plan to extend this approach to cross-domain scenarios and explore its application to more complex tasks such as emotion analysis and bias detection in natural language.

APPENDIX A
ABLATION TESTS

Results using WordNet at $b = 0.95$ (Tables 13 and 14), performance improves as both walk length (wl) and number of walks (nw) increase. SVM and LR remain the most consistent, reaching around 0.85in accuracy and F1-score,

TABLE 13. Accuracy Results for the WordNet Proposal With a Bias of 0.95

Test	Parameters	SVM	RF	LR	XGB	KNN
Test 1	wl=10, nw=50, b=0.95	0.84	0.79	0.84	0.78	0.73
Test 2	wl=20, nw=100, b=0.95	0.84	0.76	0.84	0.78	0.74
Test 3	wl=30, nw=200, b=0.95	0.84	0.78	0.84	0.80	0.68
Test 4	wl=40, nw=300, b=0.95	0.85	0.79	0.85	0.81	0.71

TABLE 14. F1-Score Results for the WordNet Proposal With a Bias of 0.95

Test	Parameters	SVM	RF	LR	XGB	KNN
Test 1	wl=10, nw=50, b=0.95	0.84	0.78	0.84	0.79	0.71
Test 2	wl=20, nw=100, b=0.95	0.84	0.76	0.84	0.78	0.72
Test 3	wl=30, nw=200, b=0.95	0.83	0.77	0.83	0.80	0.68
Test 4	wl=40, nw=300, b=0.95	0.85	0.79	0.86	0.81	0.70

TABLE 15. Accuracy Results for the WordNet Proposal With a Bias of 0.15

Test	Parameters	SVM	RF	LR	XGB	KNN
Test 1	wl=10, nw=50, b=0.15	0.82	0.74	0.82	0.76	0.70
Test 2	wl=20, nw=100, b=0.15	0.81	0.73	0.82	0.76	0.69
Test 3	wl=30, nw=200, b=0.15	0.83	0.76	0.84	0.77	0.68
Test 4	wl=40, nw=300, b=0.15	0.83	0.75	0.83	0.77	0.71

TABLE 16. F1-Score Results for the WordNet Proposal With a Bias of 0.15

Test	Parameters	SVM	RF	LR	XGB	KNN
Test 1	wl=10, nw=50, b=0.15	0.83	0.75	0.82	0.76	0.66
Test 2	wl=20, nw=100, b=0.15	0.81	0.74	0.82	0.79	0.66
Test 3	wl=30, nw=200, b=0.15	0.83	0.76	0.83	0.78	0.66
Test 4	wl=40, nw=300, b=0.15	0.84	0.75	0.83	0.78	0.68

TABLE 17. Accuracy Results for the ConceptNet Proposal With a Bias of 0.95

Test	Parameters	SVM	RF	LR	XGB	KNN
Test 1	wl=10, nw=50, b=0.95	0.84	0.76	0.83	0.80	0.65
Test 2	wl=20, nw=100, b=0.95	0.84	0.78	0.81	0.79	0.69
Test 3	wl=30, nw=200, b=0.95	0.84	0.76	0.83	0.81	0.73
Test 4	wl=40, nw=300, b=0.95	0.83	0.78	0.82	0.80	0.71

while XGB benefits from deeper walks. RF and KNN show weaker results, particularly with lower nw. Overall, longer and denser walks yield richer embeddings that favour linear classifiers.

Results using WordNet at $b = 0.15$ (Tables 15 and 16), performance slightly declines compared to $b = 0.95$, particularly for RF and KNN. SVM and LR remain the most stable, while the reduced bias weakens polarity-based neighbourhood affinity, lowering the precision of models more dependent on local structures. This shows that tuning b is crucial to maximise classifier performance.

Results using ConceptNet ConceptNet at $b = 0.95$ (Tables 17 and 18), SVM and LR remain the most stable, reaching accuracy and F1-scores around 0.84–0.85. XGB also benefits from deeper walks, while RF and KNN show consistently weaker results. Overall, a high bias reinforces polarity-based structures in the embeddings, favouring linear models and XGB.

Results using ConceptNet at $b = 0.15$ (Tables 19 and 20), results are slightly lower than with $b = 0.95$. SVM and LR remain stable, while RF reaches moderate values and KNN and XGB show less robustness. The lower bias weakens polarity relations, reducing the advantage of neighbourhood- and ensemble-based models.

TABLE 18. F1-Score Results for the ConceptNet Proposal With a Bias of 0.95

Test	Parameters	SVM	RF	LR	XGB	KNN
Test 1	wl=10, nw=50, b=0.95	0.85	0.76	0.83	0.80	0.62
Test 2	wl=20, nw=100, b=0.95	0.85	0.76	0.83	0.79	0.65
Test 3	wl=30, nw=200, b=0.95	0.84	0.76	0.83	0.82	0.68
Test 4	wl=40, nw=300, b=0.95	0.84	0.78	0.83	0.80	0.66

TABLE 19. Accuracy Results for the ConceptNet Proposal With a Bias of 0.15

Test	Parameters	SVM	RF	LR	XGB	KNN
Test 1	wl=10, nw=50, b=0.15	0.80	0.77	0.81	0.76	0.65
Test 2	wl=20, nw=100, b=0.15	0.82	0.75	0.81	0.78	0.70
Test 3	wl=30, nw=200, b=0.15	0.81	0.78	0.82	0.77	0.68
Test 4	wl=40, nw=300, b=0.15	0.82	0.74	0.82	0.78	0.68

TABLE 20. F1-Score Results for the ConceptNet Proposal With a Bias of 0.15

Test	Parameters	SVM	RF	LR	XGB	KNN
Test 1	wl=10, nw=50, b=0.15	0.81	0.77	0.81	0.76	0.61
Test 2	wl=20, nw=100, b=0.15	0.82	0.75	0.81	0.78	0.67
Test 3	wl=30, nw=200, b=0.15	0.81	0.78	0.82	0.78	0.65
Test 4	wl=40, nw=300, b=0.15	0.82	0.74	0.82	0.79	0.65

APPENDIX B
K-FOLD CROSS-VALIDATION

To assess the generalisation ability of the models, we applied 5-Fold Cross-Validation. As shown in Table 21, LR and SVM achieved the best performance across both ConceptNet and WordNet embeddings, with WordNet providing slightly higher stability. These results confirm that both knowledge sources contribute effectively to the robustness of the models.

TABLE 21. Performance of Cross-Validation With ConceptNet and WordNet (B = 0.95, Wl = 40, Nw = 300)

Model	Cross Val. (ConceptNet)	Cross Val. (WordNet)
SVM	0.83	0.84
LR	0.83	0.85
XGB	0.75	0.76
RF	0.76	0.77
KNN	0.69	0.68

APPENDIX C
TIME PERFORMANCE

Training was conducted on a workstation equipped with an AMD Ryzen Threadripper PRO 5955WX (16 cores / 32 threads), 252 GB of RAM, and 32 logical CPUs. As shown in Table 22, shallow embeddings such as Word2Vec and FastText complete in seconds, while BERT requires several minutes due to the transformer's inference cost. Graph-based approaches are significantly slower, especially with WordNet, reflecting the higher density and complexity of lexical graph exploration.

TABLE 22. Time Performance With B = 0.95, Wl = 40, Nw = 300

Model	Time
Word2vec	2 seconds
Glove	12 seconds
FastText	4 seconds
Bert	5 minutes
Proposal (wordnet)	121.3 minutes
Proposal (conceptnet)	79.54 minutes

ACKNOWLEDGMENT

The authors would like to express their gratitude to the Artificial Intelligence Laboratory of the Pontifical Catholic University of Peru for providing us with access to their facilities and servers. Additionally, the authors like to acknowledge the financial support provided by the Fondo Nacional de Desarrollo Científico, Tecnológico y de Innovación Tecnológica (Fondecyt) of Concytec, which funded the doctoral studies.

REFERENCES

[1] B. Liu, *Sentiment Analysis: Mining Opinions, Sentiments, and Emotions*. New York, NY, USA: Cambridge Univ. Press, 2015.
[2] S. Minaee, N. Kalchbrenner, E. Cambria, N. Nikzad, M. Chenaghlu, and J. Gao, "Deep learning–based text classification: A comprehensive review," *ACM Comput. Surv.*, vol. 54, pp. 1–40, Apr. 2021.
[3] A. Yadav and D. K. Vishwakarma, "Sentiment analysis using deep learning architectures: A review," *Artif. Intell. Rev.*, vol. 53, pp. 4335–4385, Aug. 2020.
[4] L. Zhang, S. Wang, and B. Liu, "Deep learning for sentiment analysis: A survey," *Wiley Interdiscipl. Rev.: Data Mining Knowl. Discov.*, vol. 8, no. 4, 2018, Art. no.e1253.
[5] Z. S. Harris, "Distributional structure," *Word*, vol. 10, no. 2/3, pp. 146–162, 1954.
[6] K. Sparck Jones, "A statistical interpretation of term specificity and its application in retrieval," *J. Documentation*, vol. 28, pp. 11–21, Jan. 1972.
[7] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in *Proc. 1st Int. Conf. Learn. Representations*, Y. Bengio and Y. LeCun, Eds., Scottsdale, AZ, USA, 2013, pp. 1–12.
[8] J. Pennington, R. Socher, and C. Manning, "GloVe: Global vectors for word representation," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Doha, Qatar, 2014, pp. 1532–1543.
[9] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *Trans. Assoc. Comput. Linguistics*, vol. 5, pp. 135–146, 06 2017.
[10] M. E. Peters et al., "Deep contextualized word representations," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, M. A. Walker, H. Ji, and A. Stent, Eds., 2018, pp. 2227–2237.
[11] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, 2019, J. Burstein, C. Doran, and T. Solorio, Eds., pp. 4171–4186.
[12] A. Petrescu, C.-O. Truică, E.-S. Apostol, and A. Paschke, "EDSA-ensemble: An event detection sentiment analysis ensemble architecture," *IEEE Trans. Affect. Comput.*, vol. 16, no. 2, pp. 555–572, Apr.–Jun., 2025.
[13] E.-S. Apostol, A.-G. Pisică, and C.-O. Truică, "ATESA-BÆRT: A heterogeneous ensemble learning model for aspect-based sentiment analysis," *Knowl.-Based Syst.*, vol. 326, 2025, Art. no. 113987.
[14] A. Petrescu, C.-O. Truică, and E.-S. Apostol, "Sentiment analysis of events in social media," in *Proc. IEEE 15th Int. Conf. Intell. Comput. Commun. Process.*, 2019, pp. 143–149.
[15] M. Mitroi, C.-O. Truică, E.-S. Apostol, and A. M. Florea, "Sentiment analysis using topic-document embeddings," in *Proc. IEEE 16th Int. Conf. Intell. Comput. Commun. Process.* 2020, pp. 75–82.
[16] C.-O. Truică, E.-S. Apostol, M.-L. Şerban, and A. Paschke, "Topic-based document-level sentiment analysis using contextual cues," *Mathematics*, vol. 9, no. 21, 2021, Art. no. 2722.
[17] F. A. Lovera, Y. C. Cardinale, and M. N. Homsi, "Sentiment analysis in twitter based on knowledge graph and deep learning classification," *Electronics*, vol. 10, no. 22, 2021, Art. no. 2739.
[18] X. Han et al., "Pre-trained models: Past, present and future," *AI Open*, vol. 2, pp. 225–250, 2021.
[19] Y. Ma, "Concept-based embeddings for natural language processing," Phd thesis, Nanyang Technological Univ., Singapore, Jun. 2018. [Online]. Available: https://hdl.handle.net/10356/75838.
[20] A. R. Asokan, N. Kumar, A. V. Ragam, and S. S. Shylaja, "Interpretability for multimodal emotion recognition using concept activation vectors," in *Proc. Int. Joint Conf. Neural Netw.*, 2022, pp. 1–8.
[21] G. Schreiber et al., *Knowledge Engineering and Management: The CommonKADS Methodology*. Cambridge, MA, USA: MIT Press, 1999.
[22] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: Online learning of social representations," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, New York, NY, USA, 2014, pp. 701–710.
[23] W. L. Hamilton, R. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, Red Hook, NY, USA, 2017, pp. 1025–1035.
[24] A. Grover and J. Leskovec, "Node2Vec: Scalable feature learning for networks," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, New York, NY, USA, 2016, pp. 855–864.
[25] A. Narayanan, M. Chandramohan, R. Venkatesan, L. Chen, Y. Liu, and S. Jaiswal, "Graph2vec: Learning distributed representations of graphs," in *Proc. 13th Int. Workshop Mining Learn. Graphs*, 2017, pp. 1–8.
[26] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. 5th Int. Conf. Learn. Representations*, Toulon, France, 2017, pp. 1–14.
[27] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," *Int. Conf. Learn. Representations*, 2018, pp. 1–12.
[28] V. S. Anoop, C. S. Krishna, and U. H. Govindarajan, "Graph embedding approaches for social media sentiment analysis with model explanation," *Int. J. Inf. Manage. Data Insights*, vol. 4, no. 1, 2024, Art. no. 100221.
[29] H. T. Phan, N. T. Nguyen, and D. Hwang, "Aspect-level sentiment analysis: A survey of graph convolutional network methods," *Inf. Fusion*, vol. 91, pp. 149–172, 2023.
[30] S. Vashishth, M. Bhandari, P. Yadav, P. Rai, C. Bhattacharyya, and P. Talukdar, "Incorporating syntactic and semantic information in word embeddings using graph convolutional networks," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*, Florence, Italy, 2019, pp. 3308–3318.
[31] L. Huang, D. Ma, S. Li, X. Zhang, and H. Wang, "Text level graph neural network for text classification," in *Proc. Conf. Empirical Methods Natural Lang. Process., 9th Int. Joint Conf. Natural Lang. Process.*, K. Inui, J. Jiang, V. Ng, and X. Wan, Eds., Hong Kong, 2019, pp. 3444–3450.
[32] L. Yao, C. Mao, and Y. Luo, "Graph convolutional networks for text classification," *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 7370–7377.
[33] T.-J. Fu, P.-H. Li, and W.-Y. Ma, "GraphRel: Modeling text as relational graphs for joint entity and relation extraction," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*, Florence, Italy, 2019, pp. 1409–1418.

[34] Y. Zhang, P. Qi, and C. D. Manning, “Graph convolution over pruned dependency trees improves relation extraction,” in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Brussels, Belgium, 2018, pp. 2205–2215.

[35] C. Hutto and E. Gilbert, “VADER: A parsimonious rule-based model for sentiment analysis of social media text,” in *Proc. Int. AAAI Conf. Web Social Media*, 2014, pp. 216–225.

[36] C.-O. Truică, J. Darmont, and J. Velcin, “A scalable document-based architecture for text analysis,” in *Proc. 12th Int. Conf. Adv. Data Mining Appl.*, J. Li, X. Li, S. Wang, J. Li, and Q. Z. Sheng, Eds., 2016, pp. 481–494.

[37] C.-O. Truica, F. Radulescu, and A. Boicea, “Comparing different term weighting schemas for topic modeling,” in *Proc. 18th Int. Symp. Symbolic Numeric Algorithms Sci. Comput.*, 2016, pp. 307–310.

[38] C.-O. Truică and C. A. Leordeanu, “Classication of an imbalanced data set using decision tree algorithms,” *Univ. Politehnica Bucharest Sci. Bull. Ser. C - Elect. Eng. Comput. Sci.*, vol. 79, 2017, pp. 69–84.

[39] L. A. Jeni, J. F. Cohn, F. De, and La Torre, “Facing imbalanced data–recommendations for the use of performance metrics,” in *Proc. Humaine Assoc. Conf. Affect. Comput. Intell. Interaction*, 2013, pp. 245–251.

[40] M. Abdelhamid and A. Desai, “Balancing the scales: A comprehensive study on tackling class imbalance in binary classification,” *arXiv:2409.19751*, 2024.

YONATAN MAMANI-COAQUIRA received the bachelor’s degree in computer and systems engineering from the National University Micaela Bastidas of Apurimac, Abancay, Peru, in 2013, and the master’s degree in computer science from the National University of San Antonio Abad of Cusco, Cusco, Peru, in 2021. He is currently working toward the Ph.D. degree in engineering, with a specialization in computer science with the Pontifical Catholic University of Peru (PUCP), Lima, Peru. He was an Intern and collaborated as a Researcher with Data Analytics Science and Engineering Laboratory, Pablo de Olavide University, Seville, Spain. He is also a Conducting Researcher with Artificial Intelligence Group, PUCP. His research interests include machine learning, deep learning, data analytics, computer vision, and natural language processing.

EDWIN VILLANUEVA received the degree in electrical engineering from the National University of San Agustin of Arequipa, Arequipa, Peru, in 2000, the master’s degree in electrical engineering from the University of São Paulo, São Paulo, Brazil, in 2007, and the Ph.D. degree in science, in 2012. He was a Postdoctoral with the Institute of Mathematical and Computer Sciences, University of São Paulo, and the Institute of Functional Genomics, Universität Regensburg, Germany. He was an Engineer in several public and private institutions in Peru, for more than six years, accumulating experience in the telecommunications and software development sectors. He was also a Postdoctoral Researcher with the Institute of Mathematical and Computer Sciences, University of São Paulo, and Universität Regensburg. Since 2017, he has been a Senior Researcher with the Pontifical Catholic University of Peru. His research focuses on the areas of computer science, specifically artificial intelligence, machine learning, probabilistic graphical models, signal processing, and bioinformatics. He had been leading research and technological development projects in multidisciplinary fields, such as modeling and prediction of atmospheric phenomena, air quality monitoring and modeling, and modeling and prediction of financial markets.