



# SemanticGraph2Vec: Semantic graph embedding for text representation

Wael Etaiwi<sup>\*</sup>, Arafat Awajan

Princess Sumaya University for Technology, Amman, Jordan

## ARTICLE INFO

### Keywords:

DeepWalk  
Graph embedding  
Semantic graph  
Random walk

## ABSTRACT

Graph embedding is an important representational technique that aims to maintain the structure of a graph while learning low-dimensional representations of its vertices. Semantic relationships between vertices contain essential information regarding the meaning of the represented graph. However, most graph embedding methods do not consider the semantic relationships during the learning process. In this paper, we propose a novel semantic graph embedding approach, called SemanticGraph2Vec. SemanticGraph2Vec learns mappings of vertices into low-dimensional feature spaces that consider the most important semantic relationships between graph vertices. The proposed approach extends and enhances prior work based on a set of random walks of graph vertices by using semantic walks instead of random walks which provides more useful embeddings for text graphs. A set of experiments are conducted to evaluate the performance of SemanticGraph2Vec. SemanticGraph2Vec is employed on a part-of-speech tagging task. Experimental results demonstrate that SemanticGraph2Vec outperforms two state-of-the-art baselines methods in terms of precision and F1 score.

## 1. Introduction

A graph is a data structure that represent the relationships between different types of objects. One of the core goals of graph-based algorithms is to extract structural information from graphs, including summary graph statistics and local graph neighborhood structures [1]. The goal of graph embedding is to learn graph mappings into low-dimensional vector spaces in which the extracted features of vertices can be learned with respect to actual graph structures. Therefore, each vertex is represented as a vector.

A semantic graph is a network that represents the semantic relationships between different concepts [2]. Network vertices represent concepts and network edges represent the semantic relationships between concepts. A text semantic graph is used to encode plain text and represent its meaning in the form of a graph. An example of a semantic network is WordNet [3], which contains lexical databases of different natural languages, such as Arabic and English. In WordNet, words are grouped into sets of synonyms called “synsets” that represent the semantic relationships between synonyms.

Graph embedding has recently become an important and interesting research topic [4]. Graph embedding aims to learn low-dimensional representations of a graph or any of its components (e.g. vertices and edges), while preserving the structure of the graph and any other additional information, (e.g. vertex attributes). Such representations are used for machine learning models and Natural Language Processing (NLP) applications [5]. Many researchers have proposed graph embedding algorithms as components of dimensional reduction techniques,

such as Zhang et al. [6], Feng et al. [7], Isomap [8], Locally Linear Embedding (LLE) [9], and Laplacian Eigenmaps [10]. Graph embedding algorithms are used to calculate the similarity between pairwise data points to construct similar graphs that can be embedded into a new low-dimensional space.

Because graph embedding represents a graph as low-dimensional vectors while maintaining the graph’s structural integrity, it differs from graph representation learning. Graph representation learning, in contrast, does not rely for low-dimensional representations [11]. For instance, a vector representation of a graph’s vertex will have the same number of dimensions as the input graph’s vertex count.

Graph embedding techniques can be categorized into three main types [12]: factorization-based, random-walk-based, and deep-learning-based techniques. Factorization-based techniques represent the connections between vertices in the form of a matrix and the matrix is factorized to obtain a graph embedding. Factorization-based methods, such as LLE and Laplacian Eigenmaps, cannot learn the structural equivalence of a graph unless the corresponding information is explicitly included in their objective functions. Factorization-based approaches are incapable of learning specific functions, such as functions for describing network connectivity. Additionally, they cannot learn structural equivalences unless the corresponding information is explicitly included in their objective functions.

Random-walk-based techniques employs random graph walks to approximate various attributes in a graph, including node similarity

<sup>\*</sup> Corresponding author.

E-mail addresses: [w.etaiwi@psut.edu.jo](mailto:w.etaiwi@psut.edu.jo) (W. Etaiwi), [awajan@psut.edu.jo](mailto:awajan@psut.edu.jo) (A. Awajan).

**Table 1**  
Graph representation learning methods.

Model	Year	Technique	Datasets	Evaluation measures
DeepWalk [13]	2014	Random walks	BlogCatalog, Flickr and YouTube	Macro-F1, Micro-F1
LINE [18]	2015	Random walks	a language network, two social networks, and two citation networks	Macro-F1, Micro-F1
GraRep [19]	2015	Random walks	A social network, a language network and a citation networks	Macro-F1, Micro-F1
Node2Vec [14]	2016	BFS and DFS	Wikipedia, Protein-Protein Interactions and BlogCatalog	Macro-F1, Micro-F1, AUC
DNNGR [17]	2016	Statistical	20-NewsGroup, Wine and Wikipedia Corpus	–
Attributed Random Walk [20]	2017	Random walks	–	AUC
DeepGL [21]	2017	Statistical	Generated labeled dataset	AUC
MetaGraph2Vec [22]	2018	Random walks	DBLP bibliographic	Accuracy, F score, NMI, Precision
GEMSEC [23]	2019	Random walks	Two social networks	Cluster modularity
KEC [24]	2019	Statistical	Freebase and WordNet	Mean Rank
Dyngraph2ven [25]	2020	Deep Learning	collaboration networks and social networks	Mean Average Precision

and centrality. A random walk from a vertex  $v_i$ , denoted as  $W_{vi}$ , consists of a set of vertices  $W_{vi}^1, W_{vi}^2, \dots, W_{vi}^k$  chosen randomly from the neighbors of vertex  $v_k$ . A set of random walks is used to extract structural information from a graph. Random-walk-based methods are useful in applications where graphs are only partially observed. DeepWalk [13] and Node2Vec [14] are two examples of random-walk-based techniques. The main advantage of Random-walk-based methods is that they accommodate small changes in graph structures without requiring global re-computation, and the learned model can be updated with new random walks in the modified region of a graph. According to Deng [15], deep learning is a branch of machine learning that makes use of numerous levels of representation (layers) that comprise information processing units (neurons). Deep learning offers improved data representation, which can improve machine learning approaches. Deep learning can be used to represent words in textual data and create word embeddings that can be applied to various machine learning algorithms. Instead of manually creating features, deep learning is used. The process of handcrafting features takes a lot of time and is frequently lacking. Deep learning, on the other hand, offers beneficial features and numerous levels of representation that raise the effectiveness of machine learning models. Deep autoencoders are used in deep learning techniques to reduce dimensionality based on their ability to model the nonlinear structures of data. SDNE [16] and DNNGR [17] are two models for deep learning graph embedding that use deep autoencoders to generate embedding models that capture nonlinearity in graphs. Deep-learning-based methods can learn combinations of community and structural equivalences from a graph. The weights of an autoencoder can be used as a structural representation of a graph in deep learning methods.

The remaining of this paper is organized as follows. The main research objectives and contributions are outlined in Section 3. In Section 2, the related work on graph embedding are briefly reviewed. Section 4 describes the proposed model. Experimental results are discussed in Section 5. The conclusions are summarized in Section 6.

## 2. Related work

Recently, eight different graph embedding approaches are used in the literature works, namely: DeepWalk, LINE [18], GraRep [19], Node2Vec, DNNGR-Deep, Attributed Random Walk [20], DeepGL [21], and MetaGraph2Vec [22].

As illustrated in Table 1, random walks are the most common technique used for graph embedding and unweighted graphs are the most commonly targeted graph type in experiments. DeepWalk is the first random walk approach developed for graph embedding, thus it is considered as a baseline algorithm in many recent studies [26]. DeepWalk consists of two components: a random walk generator and update process. Initially, parametrized number of random walks of a predefined length are generated. Each random walk samples from the neighbors of a visited vertex until the parametrized maximum length of random walks is reached. In the update process, the SkipGram [27] and hierarchical softmax [28] algorithms are used to update graph representations. Many enhancements and customized versions of DeepWalk are proposed in the literature works, such as LINE and Node2Vec. LINE [18] preserves local and global graph structures by capturing the first-order and second-order proximity of graph vertices. Grover et al. [14] proposed Node2Vec, which learns the continuous features of vertices. Node2Vec uses breadth-first sampling (BFS) and depth-first sampling (DFS) to configure the process of generating of random walks and reduce the number of random moves. GEMSEC, which was proposed by Rozemberczki et al. [23], aims to maintain social communities in social networks using a clustering machine learning approach. The key concept of GEMSEC is the creation of social clusters based on social properties, which are embedded and isolated from each other. Finally, dyngraph2ven [25] is a deep learning model that learns the temporal changes in a graph using a deep architecture composed of dense and recurrent layers. Dyngraph2vec uses multiple nonlinear layers to learn structural patterns in graphs and uses recurrent layers to learn temporal changes in graphs.

Over the past years, most proposed graph learning models have focused on preserving the structures of represented graphs, rather than on maintaining the semantic relationships between vertices. MetaGraph2Vec is the first model considering semantic relationships between vertices during the learning process. For semantic graph embedding, the MetaGraph2Vec model learns more informative embeddings by capturing rich semantic relationships between different types of vertices. It guides random walks in heterogeneous information networks to encode the semantic relationships between different types of vertices and generate heterogeneous walks through different types of vertices.

It should be noted that there are many different datasets used for evaluating various methods and there is no common benchmark dataset that can be used for comparison. This makes it difficult and ambiguous to compare the proposed method to other methods. The majority of

researchers use Macro-F1 and Micro-F1 as evaluation measures to evaluate their approaches and demonstrate improved results compared to previous approaches and baseline algorithms [26]. This is a clear indication that greater attention has been paid to this research area in recent years. Additionally, most existing models have enhanced the baseline algorithm (DeepWalk) in terms of how walks are generated. For example, LINE uses the BFS algorithm to select steps from a target graph, and node2vec uses both the BFS and DFS algorithms for the same purpose.

In this paper, a semantic graph embedding model is proposed to represent semantic graphs in a low-dimensional vector space. The proposed model guides random walks based on the semantic relationships between vertices. Semantic relationships are ordered according to their priority, where high-priority relationships are more likely to be included in generated paths. Therefore, the term “random” is no longer used in the proposed model.

### 3. Research objectives and contributions

There have been very few studies on graph embedding that have considered the semantic relationships between graph components (e.g., vertices or subgraphs). Additionally, most semantic graph embedding approaches focus on a specific application or specific type of graph (e.g., MetaGraph2Vec [22], which focuses on heterogeneous graphs). Therefore, the broad objective of our research is to develop a novel semantic graph embedding model in which the semantic relationships between words are considered during the walk generation process to enhance various NLP tasks, such as textual entailment and Part-of-Speech (POS) tagging. To summarize, the main contributions of this research are:

- A novel semantic graph embedding model that considers the semantic relationships between text components (words) called SemanticGraph2Vec.
- Enhance the performance of the DeepWalk model in terms of random walk generation process to achieve better results for NLP tasks.

It is worth to mention that POS tagging is considered as a case study to compare the performance and impact of baseline graph embedding with that of SemanticGraph2Vec. Therefore, the experiments conducted in the paper research do not seek to achieve the best POS tagging performance.

### 4. SemanticGraph2Vec: Semantic graph embedding

The SemanticGraph2Vec model is proposed to preserve the semantic relationships between graph vertices (documents, phrases, or words) in a text graph. Semantic preservation in text graph embedding is a difficult task because semantic relationships vary based on text language and are difficult to capture in some languages.

The SemanticGraph2Vec model has three main steps, As shown in Fig. 1: Extract semantic relationships, generate semantic walks, and learn vertex representations. The first step is to extract the semantic relationships from the semantic graph that represents the original text, and then to prioritize the semantic relationships. The second step involves extracting a set of walks from the semantic graph taking into consideration the semantic relation priorities. The final step is to learn the vertices representation and produce the output vectors.

#### 4.1. Extract semantic relationships

In SemanticGraph2Vec, provided the semantic graph that represents the original text, semantic relationships are dynamically ranked by the frequency. In order to rank the semantic relations, the semantic graph is built on the basis of the dataset used, and the semantic relations are extracted and sorted according to their presence in the text. For example,

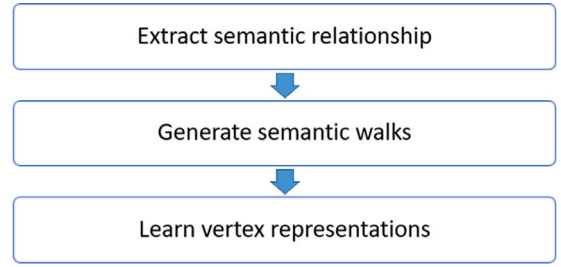


Fig. 1. SemanticGraph2Vec framework.

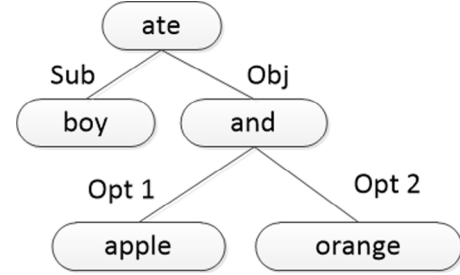


Fig. 2. Semantic graph.

the “subject” relationship has the lowest rank (highest priority) because it is the most frequently appearing relationship in text. Appendix lists semantic relationship ranks based on the semantic graph proposed by Etaiwi and Awajan [29].

#### 4.2. Generate semantic walks

In this subsection, the problem of semantic graph embedding is formalized and some preliminary definitions are presented.

**Definitions 1.** A semantic graph  $G$  is defined as a directed graph  $G = (V, E, W)$ , where  $V$  is a set of vertices (mainly representing concepts, words, or sentences),  $E$  is a set of edges (where  $E \subseteq V \times V$ ), and  $W$  is a set of semantic relationships, where  $\forall w \in W : w \in SR$  and  $SR$  is a set of semantic relationships.

**Examples 1.** A semantic graph that represent ‘the boy ate an apple and an orange’ statement is illustrated in Fig. 2. This graph consists of five vertices and four semantic edges, where  $SR$  is listed in Table A.4 in Appendix.

**Definitions 2.** Given a semantic graph  $G$ , we define a semantic walk as a sequence of vertices  $S = \{v_1, v_2, \dots, v_L\}$  of length  $L$ , in which, For each  $v_i$  ( $1 \leq i \leq L$ ) in  $S$ ,  $v_i \in V$  and for each  $v_i$  ( $1 < i \leq L$ ) in  $S$ ,  $(v_{i-1}, v_i) \in E$ .

**Examples 2.** For the semantic graph  $G$  in Fig. 2, a possible semantic walk of length  $L = 3$  is  $s = \{ate, and, orange\}$ .

**Definitions 3.** The semantic walk with the highest priority is the semantic walk that has the minimum summation of its edges ranks, therefore,  $S = \{v_1, v_2, \dots, v_L\}$  is the semantic path with the highest priority when for each  $v_i$  ( $1 \leq i \leq L$ ) in  $S$ , the rank of edge  $e(v_{i-1}, v_i)$  is the minimum rank among all possible edges  $e_1, e_2, \dots, e_n$ , where  $e_n(v_{i-1}, v_n)$  and  $e_n \in E$ .

#### 4.3. Learn vertex representations

The semantic walk generation process can explore semantic graphs according to semantic priorities. For each vertex in the semantic graph,

a set of semantic walks of length  $L$  is generated. The top- $n$  semantic walks with the lowest rank (highest priority) are selected during the sampling process. The sum of all semantic relationship ranks present in each semantic walk is calculated according to this selection procedure, the semantic walks are then ordered based on their semantic rank, and finally the most important semantic walks with the lowest ranks are considered in the vertex learning process. Inspired by the skip-gram model [30], a vertex representation is learned by optimizing the semantic neighborhood objective using Stochastic Gradient Descent with negative sampling [31] as per proposed by DeepWalk model.

## 5. Experiments and evaluation

The ability of SemanticGraph2Vec model to improve NLP tasks that use graph embedding in their techniques was evaluated. In the experiments, the POS tagging problem is used as a case study. The experiments' main purpose is to address other baseline graph embedding models that are used for improving NLP tasks, rather than achieving better POS tagging performance.

### 5.1. POS tagging: Case study

POS tagging is the process of tagging words in a given body of text automatically based on their context, role, or relationships with other words. Used tags (called the tag set) may include nouns, verbs, adjectives, and adverbs. POS tagging is considered to be a classification problem in which the output represents the targeted classes. Additionally, POS tagging is a fundamental NLP task that is used in the preprocessing phase of many NLP approaches, such as those proposed in [32–34].

**Examples 3.** The statement “The boy plays football” is POS tagged according to the Stanford POS tagger<sup>1</sup> as “The/DT boy/NN plays/VBZ football/NN” where DT denotes a determiner, NN denotes a noun, and VBZ denotes a verb (third-person singular present).

POS taggers can be classified into three main types [35,36]: rule-based taggers, statistical taggers, and hybrid taggers. A set of predefined rules is used for assigning POS tags to each word in a rule-based tagger. In a statistical tagger, frequencies and probabilities are used to identify the most appropriate tag for each word. Finally, in hybrid approaches, a set of rules is applied based on various statistical calculations.

### 5.2. Dataset

To evaluate the ability of the proposed model to improve the task of POS tagging, a custom dataset extracted from news articles was used in the experiments. This dataset was compiled from an Arabic news website called AlJazeera News.<sup>2</sup> It consists of more than 3500 words that were annotated with their corresponding POS tags using the Farasa POS tagger [37]. Farasa is an open-source text-processing toolkit that offers many Arabic text processing libraries for lemmatization, POS tagging, dependency parsing, and name-entity recognition. The Farasa results are chosen as the reference tag because they provided high-quality POS tagging results with 98.1% accuracy [38].

**Table 2**

Sensitivity analysis of the proposed model parameters.

Walk length	Number of walks			
	10	30	50	70
10	66.54%	66.58%	66.93%	66.52%
30	66.56%	66.77%	66.36%	66.63%
50	66.45%	66.85%	66.32%	66.27%
70	66.45%	66.85%	66.32%	66.62%

**Table 3**

Experimental results.

Model	Precision	Recall	F1 score
DeepWalk	44.03%	65.32%	51.67%
Node2Vec	42.69%	65.34%	51.64%
SemanticGraph2vec	47.16%	65.30%	52.66%

### 5.3. Experiments

Experiments were performed to evaluate the ability of SemanticGraph2Vec to improve POS tagging tasks compare with other baseline graph embedding models. The evaluation metrics considered were precision, recall, and F-score. Precision is the proportion of correct decisions made over the total number of decisions for a given class. Recall refers to the fraction of correct decisions provided by the POS tagger over the total number of POS tags for a given class. The evaluation process is a conditional decision-making process in which the judgement of the proposed model is considered to be correct if and only if it is compatible with the ground truth labels in the data collection. Finally, the F-score is the harmonic mean of precision and recall.

The sensitivity of the semantic walk length and total number of semantic walks for each vertex was measured and analyzed in separate experiments. Table 2 lists the accuracy results of applying the proposed model to various parameters.

The sensitivity analysis focused on two parameters: the total number of semantic walks for each vertex and the semantic walk length  $L$ . The results of the experiments revealed that the greatest accuracy was obtained when the walk length was  $L = 10$  and the total number of walks was 50. In the experiments, the impact of the proposed SemanticGraph2Vec method was evaluated. The performance of the proposed model was measured based on the average results of 20 runs, which each divided the original dataset into a training dataset (70% of the original dataset) and testing dataset (30% of the original dataset). As indicated in the related work section (Section 2), none of the approaches listed in Table 1 are proposed to handle semantic graphs. As a result, the proposed model is compared with the baseline models that researchers employ as a benchmark. The results are compared to those of the main baseline graph embedding models, namely Deepwalk and Node2Vec. Deepwalk is selected for comparison because it is the first random-walk-based graph embedding model and it is used as a baseline method to compare with (see Table 1). Furthermore, since the proposed SemanticGraph2vec is a graph embedding model to represent vertices, node2vec is selected for comparison because, according to Goyal and Ferrara [12], it outperforms other graph embedding methods on the task of vertex classification. Results for these models (shown in Table 3) were obtained by applying them the same datasets as the proposed model. Three separate evaluation measures are considered: precision, recall and F-score. The results demonstrate that the proposed model outperforms the baseline models on most of the assessment metrics.

The performance of the proposed model depends on a variety of factors, including the consistency of the input semantic graphs and the values of the semantic walk parameters. Because SemanticGraph2Vec considers the semantic relationships in input graphs, semantic walks provide richer knowledge regarding the relationships between words compared to the baseline models. This significant difference between

<sup>1</sup> <http://nlp.stanford.edu:8080/parser/index.jsp>

<sup>2</sup> [www.aljazeera.net](http://www.aljazeera.net)



**Table A.4**  
Semantic relationship ranks.

Rank	Semantic relationship name	Semantic relationship description
1	SBJ	Subject
2	OBJ	Object
3	ADJ	Adjective
4	IDF	Identifier
5	MOD	Modifier
6	OP1	Option 1
7	OP2	Option 2
8	OP3	Option 3
9	OP4	Option 4
10	OP5	Option 5
11	OP6	Option 6
12	LOCATION	Location
13	PLACE	Place
14	DIRECTION	Direction
15	SOURCE	Source location
16	DESTINATION	Destination location
17	START-LOC	Start location
18	DATE/TIME	Date or time
19	TIME	Time
20	EXACT	Exact date
21	START	Start date or start time
22	FINISH	Finish date or finish time
23	DURATION	Time duration
24	WEEKDAY	Week day
25	DATE	Date
26	MONTH	Month
27	YEAR	Year
28	DECADE	Decade
29	WEEK	Week
30	HOURL	Hour
31	MINUTE	Minute
32	SECOND	Second

SemanticGraph2Vec and the baseline models explains the efficiency improvement of NLP tasks. Additionally, DeepWalk can produce redundant pathways that yield smaller distinct sample sizes, which can affect the overall performance of the model.

## 6. Conclusion

In this paper, a semantic graph embedding model called SemanticGraph2Vec is proposed. SemanticGraph2Vec considers the semantic relationships in input semantic graphs during the process of generating semantic walks. Semantic relationships are ranked according to their frequency in the input text and the most commonly used semantic relationships are assigned lower ranks (higher priority). The final semantic walks consist of vertices with the highest cumulative priority of semantic edges. The SemanticGraph2Vec model was evaluated based on its ability to improve the task of POS tagging. The Arabic language was considered as a case study in our experiments. Experimental results demonstrated that SemanticGraph2Vec improves the efficiency of POS tagging compared to two baseline models: DeepWalk and Node2Vec.

## CRedit authorship contribution statement

**Wael Etaiwi:** Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Resources, Software, Validation, Visualization, Writing – original draft. **Arafat Awajan:** Project administration, Supervision, Writing – review & editing.

## Declaration of competing interest

No author associated with this paper has disclosed any potential or pertinent conflicts which may be perceived to have impending conflict with this work.

## Data availability

No data was used for the research described in the article.

## Appendix. Semantic relationship ranks based on the semantic graph proposed by Etaiwi et al. [29]

See Table A.4.

## References

- [1] Hamilton WL, Ying R, Leskovec J. Representation learning on graphs: Methods and applications. *IEEE Data Eng Bull* 2017;40:52–74.
- [2] Borgida A, Sowa JF. Principles of semantic networks: Explorations in the representation of knowledge. Morgan Kaufmann Series in Representation and Reasoning, Morgan Kaufmann; 1991.
- [3] Miller GA. WordNet: a lexical database for English. *Commun ACM* 1995;38(11):39–41.
- [4] Xu M. Understanding graph embedding methods and their applications. *SIAM Rev* 2021;63(4):825–53.
- [5] Etaiwi W, Awajan A. Graph-based arabic NLP techniques: A survey. *Procedia Comput Sci* 2018;142:328–33. <http://dx.doi.org/10.1016/j.procs.2018.10.488>.
- [6] Zhang H, Gabbouj M. Feature dimensionality reduction with graph embedding and generalized hamming distance. In: 2018 25th IEEE international conference on image processing (ICIP). IEEE; 2018, p. 1083–7. <http://dx.doi.org/10.1109/icip.2018.8451089>.
- [7] Feng F, Li W, Du Q, Ran Q. Sparse graph embedding dimension reduction for hyperspectral image with a new spectral similarity metric. In: 2017 IEEE international geoscience and remote sensing symposium (IGARSS). IEEE; 2017, p. 13–6. <http://dx.doi.org/10.1109/igarss.2017.8126821>.
- [8] Tenenbaum JB, De Silva V, Langford JC. A global geometric framework for nonlinear dimensionality reduction. *Science* 2000;290(5500):2319–23.
- [9] Roweis ST, Saul LK. Nonlinear dimensionality reduction by locally linear embedding. *Science* 2000;290(5500):2323–6.
- [10] Belkin M, Niyogi P. Laplacian eigenmaps and spectral techniques for embedding and clustering. In: *Advances in neural information processing systems*. 2002, p. 585–91.
- [11] Cai H, Zheng VW, Chang KC-C. A comprehensive survey of graph embedding: Problems, techniques, and applications. *IEEE Trans Knowl Data Eng* 2018;30(9):1616–37. <http://dx.doi.org/10.1109/tkde.2018.2807452>.
- [12] Goyal P, Ferrara E. Graph embedding techniques, applications, and performance: A survey. *Knowl-Based Syst* 2018;151:78–94.
- [13] Perozzi B, Al-Rfou R, Skiena S. Deepwalk: Online learning of social representations. In: *Proceedings of the 20th ACM SIGKDD international conference on knowledge discovery and data mining*. ACM; 2014, p. 701–10.
- [14] Grover A, Leskovec J. node2vec: Scalable feature learning for networks. In: *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. ACM; 2016, p. 855–64.
- [15] Deng L. Deep learning: Methods and applications. *Found Trends Signal Process* 2014;7(3–4):197–387. <http://dx.doi.org/10.1561/20000000039>.
- [16] Wang D, Cui P, Zhu W. Structural deep network embedding. In: *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. ACM; 2016, p. 1225–34.
- [17] Cao S, Lu W, Xu Q. Deep neural networks for learning graph representations. In: *Proceedings of the thirtieth AAAI conference on artificial intelligence*. AAAI '16, AAAI Press; 2016, p. 1145–52. URL <http://dl.acm.org/citation.cfm?id=3015812.3015982>.
- [18] Tang J, Qu M, Wang M, Zhang M, Yan J, Mei Q. Line: Large-scale information network embedding. In: *Proceedings of the 24th international conference on world wide web*. International World Wide Web Conferences Steering Committee; 2015, p. 1067–77.
- [19] Cao S, Lu W, Xu Q. Grarep: Learning graph representations with global structural information. In: *Proceedings of the 24th ACM international conference on information and knowledge management*. ACM; 2015, p. 891–900.
- [20] Ahmed NK, Rossi RA, Zhou R, Lee JB, Kong X, Willke TL, Eldardiry H. Representation learning in large attributed graphs. *Stat* 2017;1050:25.
- [21] Rossi RA, Zhou R, Ahmed NK. Deep inductive network representation learning. In: *Companion proceedings of the the web conference 2018*. ACM Press; 2018, p. 953–60. <http://dx.doi.org/10.1145/3184558.3191524>.
- [22] Zhang D, Yin J, Zhu X, Zhang C. MetaGraph2Vec: complex semantic path augmented heterogeneous network embedding. In: *Pacific-Asia conference on knowledge discovery and data mining*. Springer; 2018, p. 196–208.
- [23] Rozenberczki B, Davies R, Sarkar R, Sutton C. GEMSEC. In: *Proceedings of the 2019 IEEE/ACM international conference on advances in social networks analysis and mining*. ACM; 2019, p. 65–72. <http://dx.doi.org/10.1145/3341161.3342890>.
- [24] Guan N, Song D, Liao L. Knowledge graph embedding with concepts. *Knowl-Based Syst* 2019;164:38–44. <http://dx.doi.org/10.1016/j.knosys.2018.10.008>.

- [25] Goyal P, Chhetri SR, Canedo A. dyngraph2vec: Capturing network dynamics using dynamic graph representation learning. *Knowl-Based Syst* 2020;187:104816. <http://dx.doi.org/10.1016/j.knosys.2019.06.024>.
- [26] Etaiwi WA, Awajan A. Learning graph representation: A comparative study. In: 2018 international arab conference on information technology (ACIT). IEEE; 2018, p. 1–6. <http://dx.doi.org/10.1109/acit.2018.8672689>.
- [27] Mikolov T, Chen K, Corrado GS, Dean J. Efficient estimation of word representations in vector space. 2013. *CoRR* arXiv:1301.3781.
- [28] Morin F, Bengio Y. Hierarchical probabilistic neural network language model. In: *Aistats*, vol. 5. Citeseer; 2005, p. 246–52.
- [29] al Etaiwi WM, Awajan A. Arabic text semantic graph representation. In: 2019 2nd international conference on new trends in computing sciences. ICTCS, 2019, p. 1–6. <http://dx.doi.org/10.1109/ICTCS.2019.8923115>.
- [30] Mikolov T, Chen K, Corrado GS, Dean J. Efficient estimation of word representations in vector space. 2013. *CoRR* arXiv:1301.3781.
- [31] Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J. Distributed representations of words and phrases and their compositionality. In: *Advances in neural information processing systems*. 2013, p. 3111–9.
- [32] Lei J, Rao Y, Li Q, Quan X, Wenxin L. Towards building a social emotion detection system for online news. *Future Gener Comput Syst* 2014;37:438–48. <http://dx.doi.org/10.1016/j.future.2013.09.024>.
- [33] Vairetti C, Martínez-Cámara E, Maldonado S, Luzón V, Herrera F. Enhancing the classification of social media opinions by optimizing the structural information. *Future Gener Comput Syst* 2020;102:838–46. <http://dx.doi.org/10.1016/j.future.2019.09.023>.
- [34] Pandya A, Oussalah M, Monachesi P, Kostakos P. On the use of distributed semantics of tweet metadata for user age prediction. *Future Gener Comput Syst* 2020;102:437–52. <http://dx.doi.org/10.1016/j.future.2019.08.018>.
- [35] Kumawat D, Jain V. POS tagging approaches: A comparison. *Int J Comput Appl* 2015;118(6):32–8. <http://dx.doi.org/10.5120/20752-3148>.
- [36] Singh J, Joshi N, Mathur I. Development of marathi part of speech tagger using statistical approach. In: 2013 international conference on advances in computing, communications and informatics (ICACCI). IEEE; 2013, p. 1554–9. <http://dx.doi.org/10.1109/icacci.2013.6637411>.
- [37] Zhang Y, Li C, Barzilay R, Darwish K. Randomized greedy inference for joint segmentation, POS tagging and dependency parsing. In: *Proceedings of the 2015 conference of the North American chapter of the association for computational linguistics: Human language technologies*. Association for Computational Linguistics; 2015, p. 42–52. <http://dx.doi.org/10.3115/v1/n15-1005>.
- [38] Darwish K, Abdelali A, Mubarak H. Using stem-templates to improve arabic POS and gender/number tagging. In: *Proceedings of the ninth international conference on language resources and evaluation (LREC'14)*. Reykjavik, Iceland: European Language Resources Association (ELRA); 2014, p. 2926–31.

**Dr. Wael Etaiwi** is an assistant professor in the Department of Business Information Technology at Princess Sumaya University for Technology, Jordan. He received his B.Sc. degree in Computer Information Systems from the Hashemite University in 2007, his M.Sc. Degree in Computer Science in 2011 from Al Balqaa Applied University, and his Ph.D. in Computer Science from Princess Sumaya University for Technology in 2020. Dr. Al Etaiwi has 13 years of experience in software development and system analyst. His research interests include, but are not limited to, Artificial intelligence, Data mining, and Natural Language Processing.

**Prof. Arafat Awajan** is a Full Professor at Princess Sumaya University for Technology (PSUT). He received his Ph.D. degree in Computer Science from the University of Franche-Comte, France in 1987. He has held various administrative and academic positions at the Royal Scientific Society and Princess Sumaya University for Technology. Head of the Department of Computer Science (2000–2003) Head of the Department of Computer Graphics and Animation (2005–2006) Dean of the King Hussein School for Information Technology (2004–2007) Director of the Information Technology Center, RSS (2008–2010) Dean of Student Affairs (2011–2014) Dean of the King Hussein School for Computing Sciences (2014–2017) He is currently the vice president of the university (PSUT). His research interests include •Natural Language Processing •Arabic Text Mining •Digital Image Processing.