Graph-based Aspect Representation Learning for Entity Resolution

Zhenqi Zhao[1]*, Yuchen Guo[2]*, Dingxian Wang[1]†
Yufan Huang[1], Xiangnan He[3], Bin Gu[4]

[1]**eBay Inc.** {zhenqzhao, diwang, yufhuang}@ebay.com
[2]**Nanjing University** mf1814036@smail.nju.edu.cn
[3]**University of Science and Technology of China** xiangnanhe@gmail.com
[4]**Mohamed bin Zayed University of Artificial Intelligence** jsgubin@gmail.com

Abstract

Entity Resolution (ER) identifies records that refer to the same real-world entity. Deep learning approaches improved the generalization ability of entity matching models, but hardly overcame the impact of noisy or incomplete data sources. In real scenes, an entity usually consists of multiple semantic facets, called aspects. In this paper, we focus on entity augmentation, namely retrieving the values of missing aspects. The relationship between aspects is naturally suitable to be represented by a knowledge graph, where entity augmentation can be modeled as a link prediction problem. Our paper proposes a novel graph-based approach to solve entity augmentation. Specifically, we apply a dedicated random walk algorithm, which uses node types to limit the traversal length, and encodes graph structure into low-dimensional embeddings. Thus, the missing aspects could be retrieved by a link prediction model. Furthermore, the augmented aspects with fixed orders are served as the input of a deep Siamese BiLSTM network for entity matching. We compared our method with state-of-the-art methods through extensive experiments on downstream ER tasks. According to the experiment results, our model outperforms other methods on evaluation metrics (accuracy, precision, recall, and f1-score) to a large extent, which demonstrates the effectiveness of our method.

1 Introduction

Entity resolution has a tremendous impact on applications and research, such as deduplication, record linkage and canonicalization. It is a common challenge in various domains including digital libraries, E-commerce, natural language understanding, etc. Applying deep learning methods to solve ER problems has become a current research hotspot. These kinds of approaches have good generalization capability to improve the accuracy of prediction values on unseen data. One of the remaining challenges in tackling ER tasks is the poor quality of data, such as missing values and ambiguity. This makes pairwise distance measures approaches less effective with noisy content and context. In real-world applications, different types of aspects often interact with each other to form heterogeneous relations (Shi et al., 2018) in almost all networks. Another challenge of ER lies in how to express the relationships between the heterogeneous nature of aspects by proper data structure.

Advanced Graph Representation Learning (GRL), also called graph embedding, aiming to learn low-dimensional representations of nodes in networks, has attracted considerable attention in many real applications of networks (Perozzi et al., 2014). The universal pattern of these learning approaches is employing various types of random walk to generate node sequences and applying language models to map nodes into the same semantic vector space. In Knowledge Graph (KG), several related nodes often jointly represent a structural identity. An example of graph embedding in an aspect-based KG is shown in Figure 1.

[*] The first two authors contributed equally to this paper.

[†] Dingxian Wang is the corresponding author.

This work is licensed under a Creative Commons Attribution 4.0 International License. License details: http://creativecommons.org/licenses/by/4.0/.

Proceedings of the Graph-based Methods for Natural Language Processing (TextGraphs), pages 15–23
Barcelona, Spain (Online), December 13, 2020

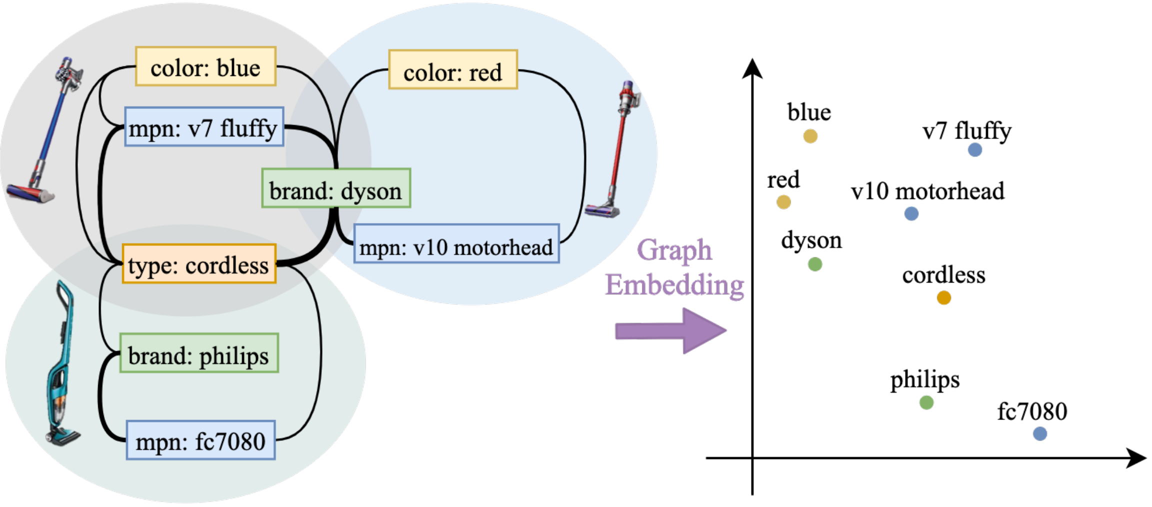


Figure 1: Schema of graph embedding in an aspect-based KG. The edges between aspects represent their co-occurrence in entities. Node colors in the graph represent different aspects types, and the thickness of the edge represents its weight. This method learns a latent space representation of aspects, which can be applied by downstream machine learning tasks.

An entity is composed of a set of aspects, and the relationship of aspects is easy to be represented in the form of graphs. In this paper, GRL is introduced to resolve entity augmentation in ER problems. We apply a heuristic feedback mechanism to the GRL field, which has long been proven successful in handling combinatorial optimization problems. This mechanism can significantly reduce the aggregation phenomenon caused by the long tail distribution of aspects, and generate more diverse and reasonable traversal sequences. We develop an algorithm (ASPECT2VEC) that learns the latent representation of aspects in a KG, by modeling a stream of random walks. ASPECT2VEC applies neural language models to process a special language composed of a set of heuristically-generated walks. The latent space representation of aspects would capture neighborhood similarity.

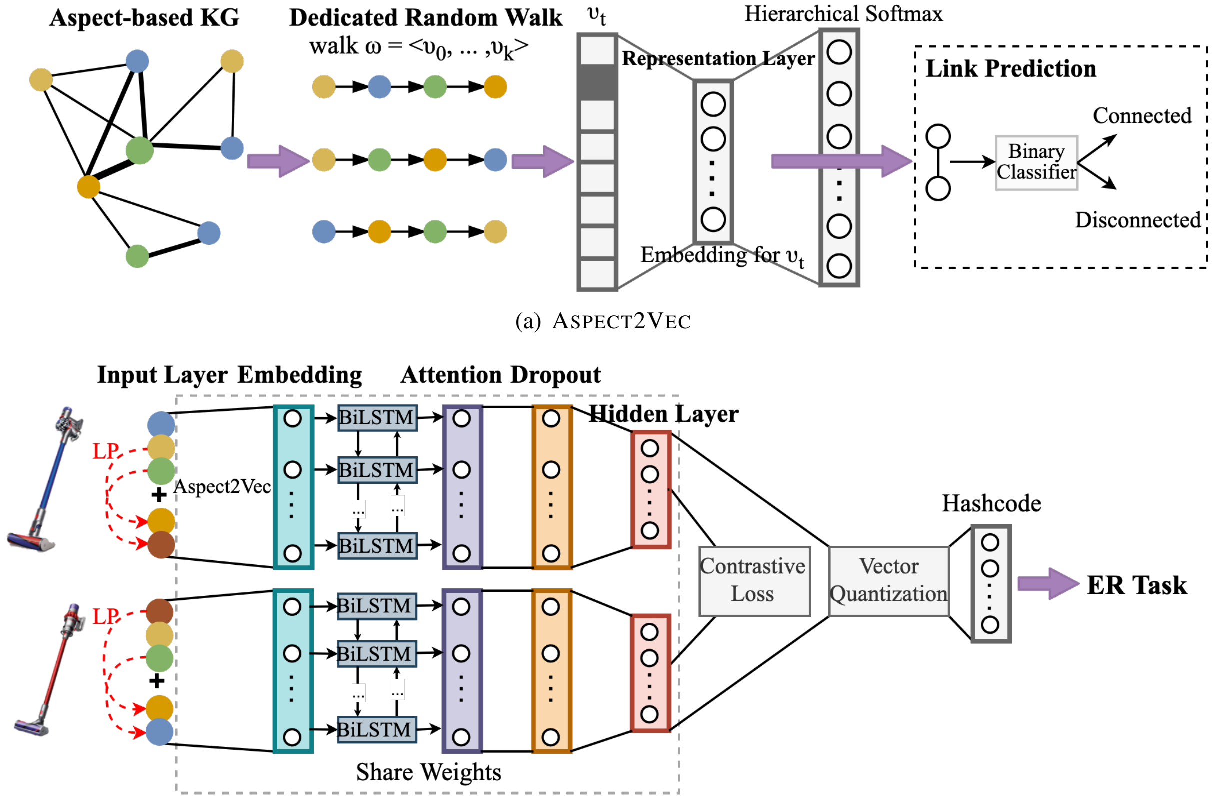
We apply ASPECT2VEC to resolve entity augmentation. In the first place, link prediction in KG is implemented and used to estimate the likelihood of linkages between aspects. This step can retrieve missing aspects of entities. Then, deep Siamese networks are constructed to generate high-quality hash codes based on semantic-preserving vectors of aspect sequences. Finally, the hashing method is employed to evaluate the performance of pairwise matching in ER.

Our main contributions are as follows:

* ASPECT2VEC. We propose a flexible aspect representation learning framework. The framework adopts a novel heuristic feedback method to generate reasonable subgraphs in an aspect-based KG, while preventing long tails phenomenon due to high-frequency aspects. Moreover, we encode aspects into a continuous vector space while preserving the semantic associations. These enrich the connotation of representation learning.

* Novel Problem Modeling. We model entity augmentation in ER as a link prediction task in KG. Normally KG is constructed from the observed interactions between aspects, which may be incomplete or inaccurate. Thus the challenge of data augmentation lies in measuring the likelihood of links between aspects.

* Evaluation. Here, we evaluate the quality of our aspect representations on downstream pairwise matching problems. The method shows significant improvements over several state-of-the-art methodologies on real public E-commerce data sets. This starts new directions for exploring data quality problems in the E-commerce field.



(a) ASPECT2VEC

(b) Deep Semantic Hashing

Figure 2: ASPECT2VEC (2a) employs a dedicated random walk algorithm to generate reasonable aspect sequences. Then SkipGram and Hierarchical Softmax are applied to convert the aspects into low-dimensional vector space for downstream tasks. In (2b), Link Prediction (LP) helps retrieve missing aspects for entity augmentation. Deep semantic hashing uses attention-based BiLSTM networks and vector quantization to generate discriminative hash codes, therefore similar pairs could be easily distinguished from dissimilar ones.

2 Proposed Approach

2.1 Problem Formulation

Given a set of entities U and a set of aspects A, each $u_i \in U$ corresponds to a series of $\{a_1, a_2, \cdots, a_m\} \subset A$. In this paper, entity resolution is scaled to a pairwise matching problem. The target of ER is employing aspects to generate discriminative hash codes so that similar pairs could be easily distinguished from dissimilar ones.

Let $G = (V, E, W)$ denote a weighted undirected graph, where V, E and W represent nodes set, edges set and weights set respectively. Each node $v \in V$ refers to an aspect, and edges refer to the co-occurrence of aspects in entities. Each weight $w \in W$ represents the co-occurrence times of aspects. A pairwise labeled dataset T is created with triples$\{(u_i, u_j, y)\}$, where $u_i, u_j \in U$ are the combinations of entities and y is a boolean label representing whether the pair of entities are matching or not.

2.2 Aspect Representation Learning

Aspect representation learning encodes aspects into a continuous vector space while preserving the semantic associations in the graph. To dig deep into the problem, we propose ASPECT2VEC, which leverages dedicated random walk to learn latent representations of nodes in the aspect-based KG. Figure 2 (a) shows a schematic diagram of ASPECT2VEC. Considering that aspects often have fixed types, the dedicated random walk helps generate reasonable sequences of aspects and avoid exhaustive search. This method lays the foundation for downstream entity augmentation.

2.2.1 Dedicated Walk

Swarm intelligence like ant colony optimization (ACO) (Dorigo and Stützle, 2019) algorithm has excellent performance in solving combination optimization problems. Artificial ants in ACO communicate with each other via pheromone, leading to a heuristic positive feedback mechanism. Inspired by this idea, we propose a novel traversal approach, which applies a heuristic feedback mechanism and tabu search to generate reasonable subgraphs. A walk $\omega = \langle v_0, \dots, v_n \rangle$ is defined as a sequence of nodes where $(v_i, v_{i+1}) \in E$. Specifically, the k-th walk moves from node v_i to node v_j with probability $p^k_{i,j}$, as defined in equation 1:

$$p^k_{i,j} = \frac{\tau^{\alpha}_{i,j} \cdot \eta^{\beta}_{i,j}}{\sum_{r \in \Gamma(i)} \tau^{\alpha}_{i,r} \cdot \eta^{\beta}_{i,r}} \tag{1}$$

where $\tau_{i,j}$ represents degree of *freshness* of the hop from node v_i to node v_j. $\alpha \geq 0$ is a parameter to control the influence of $\tau_{i,j}$. *Freshness* is initiated with a constant τ_0, and indicates the visited frequency of edge (v_i, v_j) during traversal. $\eta_{i,j}$ describes the attractiveness of the hop from v_i to v_j, which is typically set to $w_{i,j}$. $\beta \geq 1$ is a parameter to control the influence of $\eta_{i,j}$. $\Gamma(i)$ is the 1-hop neighbors of v_i.

Degrees of *freshness* are updated when a walk is completed, decreasing the value corresponding to its moves. An example of a global *freshness* updating rule is

$$\tau_{i,j} \leftarrow \begin{cases} (1-\rho)\tau_{i,j} & \text{if } (v_i,v_j) \text{ belongs to the } k\text{-th walk} \\ \tau_{i,j} & \text{otherwise} \end{cases} \tag{2}$$

where ρ is the *freshness* decay coefficient. The value of ρ depends on $\sum_k d^k_{i,j}$, which is the total length of k-th walk. $d_{i,j} = 1/w_{i,j}$ is the shortest distance between v_i and v_j.

$$\rho = \frac{1}{1 + \sum_k d^k_{i,j}} \tag{3}$$

2.2.2 Roulette Wheel Selection

To guarantee the stochastic properties of the walk, a roulette wheel selection method is adopted to choose the next hop in a walk, as shown in Algorithm 1. This method keeps the algorithm from falling into greedy search.

Algorithm 1: Roulette Wheel Selection

```
Input: $v_x$: current node; $\Gamma(x)$: one-hop neighbors of $v_x$; $N_k$: forbidden nodes in $k$-th walk;
Output: $\varphi$: the next hop node;
$\varphi = -1$;
$\mu = random(0.0, 1.0)$;
for each $v_z \in \Gamma(x)$ do
    $\mu \leftarrow \mu - p_{x,z}$;
    if $(\mu < 0)$ && $(v_z \notin N_k)$ then
        $\varphi = v_z$;
        break;
if $(\varphi == -1)$ then
    $\varphi = random(v \in \Gamma(x))$;
$update(N_k)$;
return $\varphi$;
```

Aspects that connect to similar others and have the same types in a graph are considered structural equivalence. Here, each entity only owns one specific value for a certain aspect type. Thus we restrict the walk length to the number of aspect types. If an aspect is visited during a walk, then the nodes that are with the same type as it will be added to the forbidden node set N_k.

Algorithm 2 shows procedures of how a dedicated walk generates total subgraphs. At the start of the algorithm, all parameters are initialized, including distance matrix and *freshness* matrix. In this method, degrees of *freshness* are the key to achieve heuristics. And the randomness of the algorithm is achieved through roulette wheel selection.

Algorithm 2: Dedicated Random Walk

```
Input: $V$: node set; $\chi$: node type constrains;
Output: $\lambda$: all walks;
Initialize all parameters;
for each $v \in V$ do
    $\omega = dedicatedRandomWalk(v, \chi)$;
    $updateGlobalFreshness()$;
    $\lambda.add(\omega)$;
return $\lambda$
```

2.2.3 ASPECT2VEC

SkipGram works as a language model to maximize the co-occurrence probability among the words appearing within a window. Compared to continuous bag-of-words (CBOW), SkipGram weighs nearby context words more heavily than distant context words. In ASPECT2VEC, SkipGram is applied to convert the aspects into low-dimensional vector space.

Algorithm 2 generates almost all reasonable aspect sequences. After that, each aspect node will be encoded to a corresponding representation vector. Moreover, to maximize the appearance probability of its neighbors in the walk, Hierarchical Softmax is used to approximate the probability distribution.

2.2.4 Entity Augmentation

Entity augmentation is modeled as a link prediction problem, namely predicting whether two nodes in a graph should have a link. The challenge lies in identifying spurious interactions and predicting missing links. The original connection information between aspects can be obtained from the KG and utilized to train a supervised model for LP.

We complete the entity augmentation task with a two-step solution-recall and classification. The original aspects are mapped into vector space, and the nearest neighbors that belong to the missing aspect types are recalled as candidates(the default size of the recall set is 10). Then the neighbors that are most likely to have connections with the query aspects are selected as supplement aspects. We build the LP model with a Siamese MLP structure. The input of the model is two aspect vectors, and the objective function is the contrastive loss. Accurate aspect representations facilitate entity augmentation, which greatly helps resolve downstream ER problems.

2.3 Deep Semantic Hashing

Deep semantic hashing uses deep neural networks to generate discriminative hash codes so that similar pairs could be easily distinguished from dissimilar ones (Suthee et al., 2018). Our semantic hashing method is implemented by a deep Siamese network and vector quantization.

2.3.1 Siamese Network

In the pairwise-preserving hashing method, the Siamese network is applied to explore the inner representation of symmetrical objects. We construct a deep bidirectional long short-term memory (BiLSTM) network with hierarchical attention (Z. et al., 2016) as the base structure. This model takes symmetrical input, as shown in Figure 2 (b). During the training process, the symmetrical parts share the neural weights of the network. The loss function applied here is contrastive loss (Nicosia and Moschitti, 2017) based on Euclidean distance, which can be defined as:

$$\begin{aligned} &\min \ell = \frac{1}{2N} \sum_{n=1}^{N} y_n \varepsilon_n^2 + (1 - y_n) \max(margin - \varepsilon_n, 0)^2 \\ &\varepsilon_n = ||a_n - b_n||_2 \end{aligned} \tag{4}$$

where y_n denotes whether the pair is matching or not, ε_n is the Euclidean distance between two output vectors a_n and b_n, and $margin$ is the default threshold. The loss function makes a mapping from high to low dimensional space which maps similar input vectors to nearby points on the output manifold and dissimilar vectors to distant points. In the deepest layer of the Siamese network, we apply a fully connected neural layer with Softsign activation function, which polarizes the activation value and easily converts it to binary code.

2.3.2 Vector Quantization

Hash codes are widely used in information retrieval for O(1) time complexity and data compression. Vector quantization works by dividing a large set of vectors into groups, and each group is represented by its centroid point. Utilizing the output of the last layer of the network, we can get the vectors corresponding to the aspect sequences. We apply k-means clustering to **every dimension** of the output vectors, fitting the distribution of binary codes. It means that for each dimension there will be two clusters. For a multidimensional vector, dimension independent quantization divides values into discrete groups.

3 Experimental Evaluation

Our experiments on ASPECT2VEC consist of two parts, namely link prediction and entity resolution. Each experiment compares ASPECT2VEC with several state-of-the-art graph embedding methods, including DEEPWALK (Perozzi et al., 2014), LINE (Tang et al., 2015), NODE2VEC (Grover and Leskovec, 2016) and STRUC2VEC (Ribeiro et al., 2017) on two E-commerce datasets. The comparison includes link prediction as well as pairwise matching by hash codes.

3.1 Dataset

We select two public E-commerce datasets with different sizes and sparsity for experiments. The Flipkart dataset[i] contains 20000 products, the density of aspect data is 0.08% (32569 nodes, 426202 edges). The eBay dataset[ii] contains more than 8000 vacuum cleaner items, the density of aspect data is 0.15% (22841 nodes, 401973 edges). More than one hundred thousand entity pairs are constructed from each data set, where the label is generated from UPC/EAN[iii] in eBay and item title in Flipkart. The ratio of the training set to test set is controlled at four to one by random sampling.

3.2 Experiment Setting

Each kind of product entities have their main aspect types, so the length and order of the generated sequences can be determined by restricting the aspect types, which is also utilized in tabu search. For ASPECT2VEC, α and β are both set to 1, enabling balanced heuristic weight between $\tau_{i,j}$ and $\eta_{i,j}$. τ_0 is set to 1 to initialize the *freshness* matrix. For a fair comparison, parameters of neural networks used by different algorithms are the same. The deep models for link prediction and entity resolution are Siamese network with dense layers and deep Siamese BiLSTM, respectively. And the bits of hash code is set to 64 in pairwise matching, which is corresponding to the dimensions of the output vector.

3.3 Evaluation Results

Table 1 shows the evaluation result on link prediction between aspects, and ASPECT2VEC obviously outperforms all other methods on accuracy, precision, recall, and f1-score metrics. In ASPECT2VEC, the dedicated random walk takes the co-occurrence between aspects as the heuristic factor to choose the next

[i] https://www.kaggle.com/PromptCloudHQ/flipkart-products

[ii] https://www.kaggle.com/zhenqizhao/ebay-vacuum-cleaner-products

[iii] UPC stands for Universal Product Code and EAN stands for European Article Number, both for product identification.

Table 1: Comparison of aspect representation methods on link prediction

Datasets	**Methods**	accuracy	precision	recall	f1-score
Flipkart	DeepWalk	0.7964	0.9424	0.6315	0.7562
	LINE	0.7831	0.9245	0.6166	0.7398
	Node2Vec	0.6748	0.9373	0.3745	0.5352
	Struc2Vec	0.7225	0.9141	0.4911	0.6390
	Aspect2Vec	**0.8183**	**0.9525**	**0.6701**	**0.7867**
eBay	DeepWalk	0.6797	0.9756	0.3596	0.5255
	LINE	0.8190	0.9709	0.6524	0.7804
	Node2Vec	0.7012	0.9764	0.4039	0.5714
	Struc2Vec	0.6746	0.9632	0.3536	0.5173
	Aspect2Vec	**0.8534**	**0.9825**	**0.7155**	**0.8280**

Table 2: Comparison of aspect representation methods on pairwise matching

Datasets	**Methods**	accuracy	precision	recall	f1-score
Flipkart	DeepWalk	0.9151	0.9623	0.4889	0.6484
	LINE	0.9472	**0.9786**	0.6855	0.8062
	Node2Vec	0.9204	0.9694	0.5192	0.6762
	Struc2Vec	0.9107	0.9688	0.4570	0.6210
	Aspect2Vec	**0.9608**	0.9512	**0.7959**	**0.8667**
eBay	DeepWalk	0.8987	0.9909	0.6232	0.7652
	LINE	0.9140	0.9909	0.6815	0.8076
	Node2Vec	0.9124	0.9887	0.6771	0.8038
	Struc2Vec	0.8943	**0.9915**	0.6063	0.7524
	Aspect2Vec	**0.9187**	0.9900	**0.7001**	**0.8201**

hop, and captures deep potential connections rather than random hopping. Higher accuracy indicates that the method can not only connect missing links, but also identify spurious or incorrect links. Accurate link prediction facilitates entity augmentation. Table 2 shows the result of different methods on resolving pairwise matching. Attention-based BiLSTM can accurately capture the contribution of different aspect types to the final result, and the pairwise learning method fully understands symmetrical and asymmetric information between different pairs. Compared to other methods, Aspect2Vec sacrifices a little precision but greatly improves the recall rate. The improved accuracy proves the ability to identify different kinds of entities, and the hashing method enables very fast matching. The experimental result shows the effectiveness of our method to entity augmentation, and the increase in overall performance on entity resolution.

4 Related Work

Entity resolution has attracted the interest of a large number of researchers in recent years. With the development of deep learning (DL), a growing number of DL methods are applied to solve ER problems (Mudgal et al., 2018). End-to-end deep matching models (Nie et al., 2019; Fu et al., 2020; Zhao and He, 2019) adopt similarity measures or semantic features of attributes for ER, especially dealing with heterogeneous entities. DL often requires a lot of labeled data as a training set, which is expensive to obtain. Therefore, transfer learning methods, based on a pre-trained model, are employed to solve ER tasks with little or no training data (Zhao and He, 2019). Besides, there have been many unsupervised methods to solve the data labeling problem, particularly focusing on machine labeling and error label correction (B. et al., 2019; R. et al., 2020; Chen et al., 2020). Some of the methods mentioned above pay attention to overcoming the dirty or heterogeneous data. However, how to deal with incomplete data and augment data quality in ER still needs further research. The method we proposed applies graph representation learning to resolve this problem.

Graph representation learning is dedicated to mapping nodes in networks into the same vector space, while maintaining the semantic association between nodes(Perozzi et al., 2014; Grover and Leskovec, 2016; Ribeiro et al., 2017; Shi et al., 2018; Tang et al., 2015; Wang et al., 2016; Ristoski and Paulheim, 2016). This kind of technique has received significant attention in the last few years with the development of natural language processing. The quality of the generated vectors is often measured by link prediction and node classification (Zhang and Chen, 2018; Ying et al., 2018; Trouillon et al., 2016). Previous researchers focused on the breadth and depth of graph traversal, but few of them take the node type into consideration during the progress of the random walk. In addition, how to avoid the long tail phenomenon as well as generating reasonable sequences in the traversal process is also a problem worth exploring.

5 Conclusion

In this paper, we proposed a novel aspect representation learning framework ASPECT2VEC, which resolves the entity augmentation problem in ER by modeling it as a link prediction problem in KG. ASPECT2VEC collaboratively explores dedicated random walks and captures semantic information between nodes in a network. Moreover, through extensive experiments on link prediction and deep semantic hashing, we demonstrated the superiority of the proposed framework to several state-of-the-art methods. Furthermore, dedicated random walk is flexible and also has great potential capability of parallelism to be explored in future research.

Acknowledgements

The research is supported by th Key Projects of Philosophy and Social Sciences Research of Chinese Ministry of Education under Grant 19JZD021. Assistance provided by eBay, Ads Shanghai Director Hua Yang, Director Wei Fang, Manager Hansi Wu was greatly appreciated.

References

Hou B., Chen Q., Shen J., Liu X., Zhong P., Wang Y., Chen Z., and Li Z. 2019. Gradual machine learning for entity resolution. In *WWW 2019*, pages 3526–3530.

Zhaoqiang Chen, Qun Chen, Boyi Hou, Zhanhuai Li, and Guoliang Li. 2020. Towards interpretable and learnable risk analysis for entity resolution. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*, pages 1165–1180.

Marco Dorigo and Thomas Stützle. 2019. Ant colony optimization: overview and recent advances. In *Handbook of metaheuristics*, pages 311–351. Springer.

Cheng Fu, Xianpei Han, Jiaming He, and Le Sun. 2020. Hierarchical matching network for heterogeneous entity resolution. pages 3637–3643, 07.

Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864.

Sidharth Mudgal, Han Li, Theodoros Rekatsinas, AnHai Doan, Youngchoon Park, Ganesh Krishnan, Rohit Deep, Esteban Arcaute, and Vijay Raghavendra. 2018. Deep learning for entity matching: A design space exploration. In *Proceedings of the 2018 International Conference on Management of Data*, pages 19–34.

Massimo Nicosia and Alessandro Moschitti. 2017. Accurate sentence matching with hybrid siamese networks. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 2235–2238.

Hao Nie, Xianpei Han, Ben He, Le Sun, Bo Chen, Wei Zhang, Suhui Wu, and Hao Kong. 2019. Deep sequence-to-sequence entity matching for heterogeneous entity resolution. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management(CIKM)*, pages 629–638.

Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710.

Wu R., Chaba S., Sawlani S., Chu X., and Thirumuruganathan S. 2020. Zeroer: Entity resolution using zero labeled examples. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*, pages 1149–1164.

Leonardo FR Ribeiro, Pedro HP Saverese, and Daniel R Figueiredo. 2017. struc2vec: Learning node representations from structural identity. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 385–394.

Petar Ristoski and Heiko Paulheim. 2016. Rdf2vec: Rdf graph embeddings for data mining. In *International Semantic Web Conference*, pages 498–514. Springer.

Yu Shi, Huan Gui, Qi Zhu, Lance Kaplan, and Jiawei Han. 2018. Aspem: Embedding learning by aspects in heterogeneous information networks. In *Proceedings of the 2018 SIAM International Conference on Data Mining*, pages 144–152. SIAM.

Chaidaroon Suthee, Ebesu Travis, and Fang Yi. 2018. Deep semantic text hashing with weak supervision. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 1109–1112.

Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. 2015. Line: Large-scale information network embedding. In *Proceedings of the 24th international conference on world wide web*, pages 1067–1077.

Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex embeddings for simple link prediction. International Conference on Machine Learning (ICML).

Daixin Wang, Peng Cui, and Wenwu Zhu. 2016. Structural deep network embedding. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1225–1234.

Zhitao Ying, Jiaxuan You, Christopher Morris, Xiang Ren, Will Hamilton, and Jure Leskovec. 2018. Hierarchical graph representation learning with differentiable pooling. In *Advances in neural information processing systems*, pages 4800–4810.

Yang Z., Yang D., Dyer C., He X., Smola A., and Hovy E. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 1480–1489.

Muhan Zhang and Yixin Chen. 2018. Link prediction based on graph neural networks. In *Advances in Neural Information Processing Systems*, pages 5165–5175.

Chen Zhao and Yeye He. 2019. Auto-em: End-to-end fuzzy entity-matching using pre-trained deep models and transfer learning. In *The World Wide Web Conference*, pages 2413–2424.