10-2020

CO2Vec: Embeddings of co-ordered networks based on mutual reinforcement

Meng-Fen CHIANG

Ee-Peng LIM

Wang-Chien LEE

Philips Kokoh PRASETYO

CO2Vec: Embeddings of Co-Ordered Networks Based on Mutual Reinforcement

Meng-Fen Chiang*, Ee-Peng Lim†, Wang-Chien Lee‡, and Philips Kokoh PRASETYO*
*{mfchiang,pprasetyo}@smu.edu.sg, Living Analytics Research Centre, Singapore
†eplim@smu.edu.sg, Singapore Management University, Singapore
‡wlee@cse.psu.edu, The Pennsylvania State University, USA

***Abstract*—We study the problem of representation learning for multiple types of entities in a *co-ordered network* where order relations exist among entities of the same type, and association relations exist across entities of different types. The key challenge in learning co-ordered network embedding is to preserve order relations among entities of the same type while leveraging on the general consistency in order relations between different entity types. In this paper, we propose an embedding model, *CO2Vec*, that addresses this challenge using mutually reinforced order dependencies. Specifically, CO2Vec explores indirect order dependencies as supplementary evidence to enhance order representation learning across different types of entities. We conduct extensive experiments on both synthetic and real world datasets to demonstrate the robustness and effectiveness of CO2Vec against several strong baselines in link prediction task. We also design a comprehensive evaluation framework to study the performance of CO2Vec under different settings. In particular, our results show the robustness of CO2Vec with the removal of order relations from the original networks.**

I. Introduction

Motivation. Knowledge representations that effectively capture knowledge semantics have been used in search, recommendation and question-answering applications. Order relation represents an important class of knowledge that has been studied in recent representation learning research [2], [6], [26], [28]. While past research focuses on order relations in single-type networks [2], [6], [26], [28], this paper studies the representation learning for an important class of knowledge structures known as **co-ordered networks**. A co-ordered network consists of two or more types of entities, where order relations exist between entities of the same type, and association relations exist between entities of different types. Co-ordered networks exist in many applications. For example, Figure 1 illustrates a simple co-ordered network related to massive open online courses (MOOC), where courses and concepts are depicted by rectangles and circles respectively. As shown, "Machine Learning" → "Probability and Computing" denotes a course order relation as the latter is a prerequisite of the former. Similarly, "SVM" → "Probability" denotes a concept order relation as the latter concept should be learned before the former. Meanwhile, "Machine Learning"- - -"SVM" denotes a course-concept association. While many co-ordered networks exist in the real world (e.g., co-ordering between jobs and skills in career progression, co-ordering between historical events and characters by time, etc.), the representations of entities in the co-ordered networks have not yet been studied and compared.

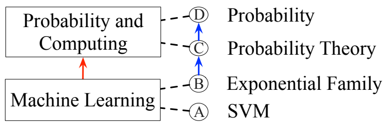


Fig. 1: Co-Ordered Network Example.

In this paper, we aim to develop effective embedding schemes for co-ordered networks to preserve the order semantics existing among entities of each entity type so as to support downstream predictive applications, such as order-aware entity search and recommendation. Despite it being a kind of network, co-ordered network has both *order* and *association* semantics that cannot be effectively modeled using traditional network embedding techniques [8]–[11], [21], [23], [24] which focus on modeling structural proximity. While knowledge graph embedding models [3], [7], [18], [22], [25], [29] can deal with general relations among entities, their effectiveness in modeling order relations in co-ordered networks also has not been validated.

In recent years, several order embeddings schemes have been proposed for *single-type order network*, i.e., those consist of *single-type ordered entities*. For instance, Vendrov et al. proposed to learn embeddings of entities in a single-type order networks using vector order embeddings (VOE) representing each entity as a point in the embedding space [26]. VOE and other follow-up order embedding models [2], [28], however, are not designed for co-ordered networks which involve two or more types of ordered entities connected by some association relations. In this work, we tackle the *co-ordered network embedding* problem defined as follows.

Co-Ordered Network Embedding Problem. *Given a network consists of two (or more) types of entities, a set of order relations for each type of entities, and a set of association relations between different types of entities, the co-ordered network embedding problem (or simply co-order embedding) is to learn low-dimensional representations for all the entities such that their order semantics are preserved.*

Two major research challenges arise in this co-order embedding problem. The first research challenge is due to the **sparsity of order relations** in co-ordered networks. For instance, the co-ordered network used in research on course prerequisite analytics [13], [15] consists of 654 course entities

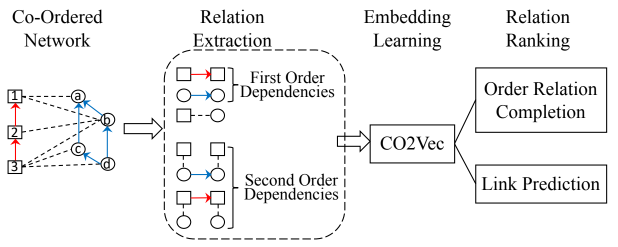


Fig. 2: Overview of the Proposed Model.

and 407 concept entities but there exist only 850+ and 990+ order relations among course entities and concept entities, respectively. The very sparse order relations imply insufficient order relations to learn the entity embeddings. Moreover, those entity pairs that do not form order relations could either be negatively ordered or unknown in their ordering [4], [16], [17], [29]. These observations motivate us to explore auxiliary information to aid the learning of co-order embedding.

The second challenge lies in the **order consistency** between order relations of different entity types. Order consistency in co-ordered networks refers to the correlation of order relations between different types of entities through associations. If the order relations in different types of entities are very much aligned (correlated) with each other, i.e., there is a strong order consistency between them, they may be able to provide additional supports via associations between these different types of entities, which are particularly useful for co-ordered representation learning. Effective co-order embedding schemes thus have to learn the representations of entities of different types together so as to incorporate such correlations. The issue of order consistency leads to our exploration of the mutual reinforcement property. Formal definition of order consistency is provided in Section III.

Overview of Our Proposed Approach. To address the above challenges, we propose a co-order embedding framework as shown in Figure 2. In this framework, given a co-ordered network with two or more entity types, we perform: (1) relation extraction, (2) embedding learning, and (3) relation ranking.

In *relation extraction*, we extract both *first* and *second order dependencies* from the co-ordered network. First order dependency $v_i \rightarrow v_j$ refers to a direct order relation. Second order dependency refers to a chain that connects an entity v_i to another entity v_j of the same type via associations with an ordered pair of entities of different type. The *co-order embedding learning* essentially takes positive and negative samples from extracted first order and second order dependencies to jointly learn order embeddings for two types of entities. In *relation ranking*, the learned co-order embedding can be used to perform several downstream tasks such as order relation completion or link prediction tasks. The former seeks to find all entities that a given query entity depends on. The latter seeks to determine if an order relation exists between two entities of the same type.

Contributions. The contributions of this paper are summarized below:

- We identify mutual reinforcement properties between different types of entities and formulate second order dependencies, which provide auxiliary information to learn high quality order embeddings in the co-ordered networks.
- We propose a generic co-order embedding model, *CO2Vec*, which incorporates the first and second order dependencies to leverage on the mutual reinforcement between the order networks of different entity types.
- We conduct extensive experiments on four synthetic and two real world datasets to demonstrate the robustness and effectiveness of CO2Vec model compared against several strong baseline embedding models. Our experiment results show the superiority of CO2Vec in learning high quality order representations in link prediction tasks. Furthermore, we demonstrate two useful applications driven by co-order embeddings. Lastly, we demonstrates the robustness of CO2Vec when order relation sparsity issue is presented.

II. Related Work

Prerequisite Relation Learning. Prerequisite relation is a kind of order relation which exists among courses and concepts. There are several works studying prerequisite relation learning among courses and concepts but they did not address the representation learning issues [1], [5], [12], [13], [15], [19], [20]. For example, Some of these works focused on predicting prerequisite relations among concepts associated with university courses [13], [15]. Liu et al. learned the prerequisite relations among concepts using some observed course level prerequisite relations and the mapping from courses to concepts.

Pan et al. proposed to automatically identify all course concepts from online MOOCs video clips before their prerequisite relations are predicted [19], [20]. Liang et al. applied active learning to address limited training data for concept prerequisite prediction [12]. None of these existing works explores representation learning for co-order relations.

Knowledge Graph Embedding. Lately, several embedding models have been proposed to learn representations of entities and relations in a knowledge graph [3], [7], [14], [17], [18], [25], [29], [30]. Translation-based models, e.g., TransE [3], TransH [14], embed entities and relations by imposing a geometrical structural bias such that the head entity representation would be close to the tail entity representation once the head is translated by the corresponding relation vector. Such translation-based models focus largely on turning the symbolic relations in knowledge graphs into different translation vectors. Order relation on the other hand has special ranking semantics which is not found in ordinary symbol relations. The knowledge graph embedding models therefore may not learn order embedding well.

Order Embedding Learning. Order embeddings for single-type entities have been effectively applied to word hypernym classification, image-caption ranking and textual entailment [2], [26]–[28]. Vendrov et al. proposed to learn vector order embeddings (VOE) of non-negative coordinates from observed order relations [26]. Due to its limitation inherited from deterministic vector order embeddings, recent works incorporate uncertainty in order representation to enrich expressiveness and to perform prediction with uncertainty, such as probabilistic extensions of order embeddings [2], [28] and box lattice representation of order embeddings [27]. Athiwaratkun et al. introduced density order embedding (DOE) to model hierarchical data via encapsulation of probability densities [2]. Our work represents an important extension of order embedding to co-ordered networks.

III. Preliminaries

In this section, we define a few key terms before presenting our proposed embedding model in Section IV

Co-Ordered Network. A co-ordered network G=(V,E) consists of two or more types of entities $\{V_1, V_2, \cdots, V_M\}$ with associations between entities of different types, and order relations among entities.

To keep the discussion simple, we consider two types of entities henceforth. We denote one entity type by A and another by B. We use $V_A \in V$ and $V_B \in V$ to denote the set of type-A and type-B entities, respectively. E_{AA} (and E_{BB}) denotes the set of order relations among entities from V_A (and V_B, respectively). Lastly, E_{AB} denotes the associations between type-A and type-B entities. The definitions and ideas presented can be easily extended to accommodate more types of entities.

Example. Consider the earlier example of course-concept co-ordered network in Figure 1. V_{course}=$\{v_{Prob\ and\ Comp}$, $v_{Mach\ Learning}\}$ and $V_{concept}$=$\{v_{Probability}$, $v_{Prob\ Theory}$, $v_{Exp\ Family}$, $v_{SVM}\}$ denote the course entity set and concept entity set, respectively; E_{AA} (red links) and E_{AA} (blue links) denote the set of course order relations and concept order relations, respectively; and lastly E_{AB} (dashed black links) denotes the associations between course and concept entities.

Order-Preserving Representation. Let a type-m entity set V_m have the order relation $\rightarrow$ such that for all v_i, v_j, $v_k \in V_m$, the following non-trivial properties hold: (1) if $v_i \rightarrow v_j$ and $v_i \neq v_j$, then $v_j \not\rightarrow v_i$ (antisymmetry), and (2) if $v_i \rightarrow v_j$ and $v_j \rightarrow v_k$, then $v_i \rightarrow v_k$ (transitivity). If the representations of all type-m entities satisfy both antisymmetry and transitivity properties, the entity representations are *order-preserving*.

Co-Ordered Relations. Co-ordered relations involve order relations ($\rightarrow$) for different types of entities such that *order consistency* between associated entities of different types is observed. That is, given type-A entity set V_A and type-B entity set V_B ($V_A \neq V_B$), if $v_{A,i} \rightarrow v_{A,j}$ where $v_{A,i}, v_{A,j} \in V_A$, there exists $v_{B,p} \rightarrow v_{B,q}$ where $v_{B,p}, v_{B,q} \in V_B$, such that $v_{A,i}$ is associated with $v_{B,p}$, and $v_{A,j}$ is associated with $v_{B,q}$.

Order Consistency. The order consistency involves in two set of new relations to be found in a co-ordered network. One is the supporting chains to infer potential type-A entity ordered pairs via associations with other ordered pairs of entities of type-B, and the other is the supporting chains to infer potential type-B entity ordered pairs via associations with other ordered pairs of entities of type-A. To quantitatively measure the order consistency, we define the as follows:

$$\beta_A = \frac{|E'_{AA}|}{|V_A| \times (|V_A| - 1)}, \qquad (1)$$

where E'_{AA} is the number of chains that bridge type-A pairs via associations with an ordered pair of entities of type-B, and the denominator is the maximum number of entity pairs in type-A dependency network. $\beta_A \in \mathbb{R}$ is essentially the co-ordered density for type-A dependency network. Likewise, $\beta_B \in \mathbb{R}$ is the co-ordered density for type-B dependency network. The combination of $\beta_A * \beta_B$ altogether suggests the degree of order consistency in a co-ordered network. The larger the value, the higher degree of order consistency.

IV. The Proposed Model

Given a co-ordered network G=(V,E), our goal to learn low-dimensional representations of both types of entities such that the entity representations are order-preserving with respect to the observed order relations for both type-A and type-B entities. In this section, we develop a new order embedding model, called *CO2Vec*, that incorporates both first order and second order dependencies in co-ordered networks.

A. First Order Dependency

To learn good order-preserving representations, Vendrov et al. defined a relaxed geometric relation between two entities in the embedding space $\vec{V}_m \in R^{|D|}$ based on the conjunction of total order on each dimension of the embedding space, where D refers to the set of embedding dimensions [26]. This idea is realised by a loss function which penalizes order violations in a given set of order relations $v_{m,i} \rightarrow v_{m,j}$:

$$\delta(v_{m,i}, v_{m,j}) = \sum_{d=1}^{|D|} || \max(0, \vec{v}_{m,i,d} - \vec{v}_{m,j,d}) || \qquad (2)$$

where $\vec{v}$, a vector, is the order embedding of an entity v. $\vec{v}_{m,i,d}$ denotes the d-th component of the i-th entity of type m. $\delta(v_{m,i}, v_{m,j}) = 0$ if $\vec{v}_{m,i} \rightarrow \vec{v}_{m,j}$ according to the conjunction of total orders; and $\delta(v_{m,i}, v_{m,j}) > 0$ if there is an order violation in some dimension $d \in D$.

To preserve order relations in embedding learning for each type of entities, a general practice is to collect a set of positive order relations and a set of negative order relations as the first order dependencies. To form negative samples E^-_{mm} for type-m entities, for each positive order relation $(v_{m,i}, v_{m,j}) \in E^+_{mm}$ we generate a negative sample of either $(v_{m,i}, v_{m,k}) \notin E^+_{mm}$ or $(v_{m,k}, v_{m,j}) \notin E^+_{mm}$ where $v_{m,k}$'s are randomly selected from V_m. Given a set of observed order relations E^+_{mm} and a set of negative sample relations E^-_{mm} for all $m \in \{A, B\}$, the objective function to learn the order embeddings for type-A and type-B entities, respectively, is defined as a max-margin loss O^1_m that encourages positive order relations to have zero penalty, and negative order relations to have penalty greater than a margin α:

$$O_m^1 = \sum_{(v_{m,i},v_{m,j})\in E_{mm}^+} \delta(v_{m,i},v_{m,j}) + \sum_{(v_{m,k},v_{m,j})\in E_{mm}^-} \max\{0, \alpha - \delta(v_{m,k},v_{m,j})\} \tag{3}$$

where $m \in \{A, B\}$ is the entity type.

B. Second Order Dependency

Motivation. Due to the existence of co-ordered relations in co-ordered networks, we explore the idea of *mutual reinforcement* to enhance co-order embeddings learning. *Mutual reinforcement* refers to the inference of order relations for one type of entities using the observed order relations among entities of the other type, given that these two types of order relations are expected to be *order consistent*. We formally model and integrate order violations using identified *mutual reinforcement properties* and propose **CO2Vec**.

Formulation. For illustration, suppose we have no knowledge of the order relation between the two courses in Figure 1. Nonetheless, knowing that concept "Exponential Family" depends on concept "Probability" supports a potential dependency from "Machine Learning" to "Probability and Computing". Here ("Exponential Family", "Probability") is referred to an inferred concept pairs due to the explicit association between "Machine Learning" and "Exponential Family", coupled with the explicit association between "Probability and Computing" and "Probability". Hence, the tuple ("Machine Learning", "Exponential Family", "Probability", "Probability and Computing") is a qualified instance of co-ordered relation because of the second order dependency formed directly via explicitly-associated concept pairs.

Let E_{ABBA}^+=$\{(v_{A,i}, v_{B,p}, v_{B,q}, v_{A,j})\}$ be the set of instances of co-ordered relations that satisfy the second order dependency, where $(v_{A,i}, v_{B,p}) \in E_{AB}$ and $(v_{A,j}, v_{B,q}) \in E_{AB}$. Let $\phi(v_{A,i}, v_{A,j})$ be the number of co-ordered relation instances that satisfy the second order dependency in support of the type-A entity pair $(v_{A,i}, v_{A,j})$. We can control the quality of the set of co-ordered relation instances by specifying a support threshold η as follows:

$$E_{ABBA}^+ = \{(v_{A,i}, v_{B,p}, v_{B,q}, v_{A,j}) | \phi(v_{A,i}, v_{A,j}) \geq \eta\}. \tag{4}$$

We generate negative samples by randomly replacing one of the two values $v_{A,i}$ and $v_{A,j}$ with $v_{A,k}$ such that the entity pair either ($v_{A,i}$,$v_{A,k}$) or ($v_{A,j}$,$v_{A,k}$) is not qualified as an instance of the first order or second order dependency:

$$E_{ABBA}^- = \{(v_{A,i}, v_{B,p}, v_{B,q}, v_{A,j}) | (v_{A,i}, v_{A,j}) \notin E_{AA}^+ \vee (v_{B,p}, v_{B,q}) \notin E_{BB}^+\}. \tag{5}$$

Given positive and negative instances of the second order dependency, we formulate their order violations as a max-margin loss O_m^2, for all $m \in \{A, B\}$, that encourages positive co-ordered relations to have zero penalty, and negative co-ordered relations to have penalty greater than a margin α:

$$O_A^2 = \sum_{(v_{A,i},\cdot,\cdot,v_{A,j})\in E_{ABBA}^+} \phi(v_{A,i},v_{A,j}) \times \delta(v_{A,i},v_{A,j}) + \sum_{(v_{A,k},\cdot,\cdot,v_{A,j})\in E_{ABBA}^-} \phi(v_{A,k},v_{A,j}) \times \max\{0, \alpha - \delta(v_{A,k},v_{A,j})\} \tag{6}$$

where $\phi(v_{A,i}, v_{A,j})$ denotes the number of instances satisfying the second order dependency in support of the type-A pair $(v_{A,i}, v_{A,j})$. $\phi(v_{A,i}, v_{A,j})$ is used to amplify the importance of type-A pair $(v_{A,i}, v_{A,j})$. Likewise, we explore the second order dependency in support of the type-B pairs $(v_{B,i}, v_{B,j})$ to derive the order loss among type-B entities, denoted as O_B^2.

Unified Learning Objective. **CO2Vec** uses both first and second order dependencies with the following objective function.

$$L_{MR} = \Sigma_m O_m^1 + \Sigma_m O_m^2 + \lambda \|\Omega\| \tag{7}$$

where $\Sigma_m O_m^2$ is the overall order violations in light of the second order dependency across each entity type-m, and Ω is the embeddings for all types of entities with regularization parameterized by λ on Ω to prevent overfitting. λ is set to 10^{-5} as default in the experiments. We optimize L_{MR}, which forces order embeddings of positively ordered pairs, via either first order or second order dependencies, to be close to zero violation, while forcing negatively ordered pairs to be greater than a margin violation. The order embeddings can be jointly learned with alternating optimization scheme among all loss terms. That is, each loss term, O_m^1 or O_m^2 for each entity type-m is alternatively optimized until all are converged.

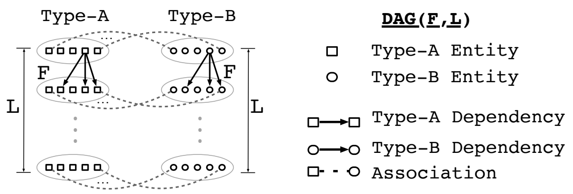


Fig. 3: An illustration of synthetic co-ordered network.

V. Model Validation on Synthetic Datasets

To validate our ideas, we first conduct experiments on synthetic datasets to quantitatively study the quality of co-order embeddings in *link prediction* tasks. We also study the usefulness of mutual reinforcement and the impact of dependency range, i.e., number of hops connecting an ordered entity pair in a co-ordered network.

A. Ground-Truth Network Generation

Consider a synthetic co-ordered network G=(V,E) where (1) V consists of two sets of entities: V_A and V_B; and (2) E contains three types of relations: type-A order relations E_{AA}, type-B order relations E_{BB}, and cross-entity associations E_{AB}. We use directed acyclic graphs (DAGs) as building blocks to generate synthetic co-ordered networks because DAGs, consisting of directed links without cycles, well capture the dependency (or ordering) relations in the targeted co-ordered networks. Specifically, each synthetic co-ordered network consists of two associated DAGs. Each DAG consists of directed links that denote the dependency relations E_{mm} for $m \in \{A, B\}$. The undirected links connecting the two DAGs are the association relations E_{AB}. The number of directed links of the DAG is controlled by fan-out F. The longest dependency range of the DAG is controlled by network depth L. Larger L results in dependency relations of longer ranges, ranging from one hop to L-1 hops. We generate a DAG by layers, each of which consists of n=5 nodes. Therefore, the total number of nodes in a DAG of network depth L (i.e., the DAG has 5 layers) is $|V_m|$=5×L for $m \in \{A, B\}$. The total number of directed links in a DAG is $|E_{mm}|$=5×$F \times (L - 1)$ for $m \in \{A, B\}$. Figure 3 provides an illustration of synthetic co-ordered networks. We generate four synthetic co-ordered networks controlled by fixing F = 3 [1] and varying the dependency range parameter L: DAG(F3,L10), DAG(F3,L20), DAG(F3,L40) and DAG(F3,L80). The statistics of these synthetic co-ordered networks are summarized in Table I.

Step 1: Co-Ordered Relation Generation. We first generate order relations for type-A and type-B entities. For the purpose of studying the impact of co-ordered properties, we create the co-ordered network with equal number of type-A and type-B entities with 1-1 matching. The type-A network topology is created as a DAG with fan-out size F and varied network depth L. These order relations are mirrored type-B entities for simplicity to create order consistency in the co-ordered network.

Step 2: Association Relation Generation. The associations between type-A and type-B entities are generated to finalize order consistency between two types of entities. For each type-A entity, we create an association relation to at least one type-B entity which are either the matching type-B entities or its other entities. The same is done for each type-B entity. The sub-network density of the co-ordered network is defined as $\rho_{m,n} = \frac{|E_{mn}|}{|V_m| \times |V_n|}$, where $m, n \in \{A, B\}$. Namely, the three sub-network densities include type-A network ($\rho_{A,A}$), type-B network ($\rho_{B,B}$), and association network ($\rho_{A,B}$). The order consistency of type-m dependency network is defined as $\beta_m = \frac{|E'_{mm}|}{|V_m| \times (|V_m| - 1)}$ for $m \in \{A, B\}$, where E'_{mm} is the total number of second order dependencies discovered through order pairs of another type. β_m measures the average number of second order dependencies per entity pair of type-m (β_m). The correlations includes type-A (β_A) and type-B order consistency (β_B), respectively. The higher the association density ($\rho_{m,n}$) and order densities (β_A, β_B) are, the more instances of co-ordered relation (in terms of second order dependency) can be leveraged to improve co-order embeddings, namely, the order relations in type-A and type-B networks can be exploited to complement each other.

[1]We leave the study of varied fan-out sizes as a future work.

TABLE I: Data statistics: sub-network density ρ and order consistency β (10^{-3}).

Dataset	Relations	#type-A/B nodes	ρ	β
DAG(F3,L10)	A, *Depends on*, A	50 / 50	46	β_A=46.1
	A, *Associates*, B	50 / 50	1.3	β_B=46.1
	B, *Depends on*, B	50 / 50	46	
DAG(F3,L20)	A, *Depends on*, A	100 / 100	24	β_A=23.5
	A, *Associates*, B	100 / 100	2.5	β_B=23.5
	B, *Depends on*, B	100 / 100	24	
DAG(F3,L40)	A, *Depends on*, A	200 / 200	12	β_A=12.3
	A, *Associates*, B	200 / 200	5	β_B=12.3
	B, *Depends on*, B	200 / 200	12	
DAG(F3,L80)	A, *Depends on*, A	400 / 400	7	β_A=7.3
	A, *Associates*, B	400 / 400	10	β_B=7.3
	B, *Depends on*, B	400 / 400	7	
UNIV	Course, *Depends on*, Course	654 / 654	2.0	β_A=60.0
	Course, *Associates*, Concept	596 / 320	6.5	β_B=3.5
	Concept, *Depends on*, Concept	407 / 407	6.1	
MOOC	Video , *Depends on*, Video	997 / 997	1	β_A=4.2
	Video, *Associates*, Concept	997 / 380	142.7	β_B=211.4
	Concept, *Depends on*, Concept	442 / 442	9	

TABLE II: Quantitative results on synthetic datasets ($|D|$=16). Best (second best) of each column are in bold (underlined).

Link Prediction Task

Data	Model	Type-A		Type-B	
		nMRR@G	NDCG	nMRR@G	NDCG
DAG (F3,L10)	RotatE	0.012	0.556	0.012	0.556
	ComplEx	0.043	0.581	0.017	0.565
	TransE	0.004	0.519	0.004	0.519
	GCN	0.034	0.581	0.011	0.561
	VOE	**0.997**	**0.992**	**0.99**	**0.99**
	CO2Vec	0.984	0.989	0.981	0.985
DAG (F3,L20)	RotatE	0.003	0.451	0.003	0.451
	ComplEx	0.004	0.457	0.005	0.458
	TransE	0.001	0.414	0.001	0.414
	GCN	0.002	0.447	0.007	0.468
	VOE	**0.961**	**0.985**	**0.972**	**0.987**
	CO2Vec	0.959	0.984	0.965	0.986
DAG (F3,L40)	RotatE	0.0	0.373	0.0	0.373
	ComplEx	0.001	0.381	0.001	0.383
	TransE	0.0	0.353	0.0	0.353
	GCN	0.002	0.4	0.001	0.387
	VOE	0.966	0.986	0.971	0.985
	CO2Vec	**0.998**	**0.995**	**0.994**	**0.992**
DAG (F3,L80)	RotatE	0.0001	0.336	0.0001	0.336
	ComplEx	0.0003	0.337	0.0004	0.341
	TransE	0.0	0.312	0.0	0.316
	GCN	0.0002	0.336	0.001	0.346
	VOE	0.997	0.98	0.9896	0.983
	CO2Vec	**0.998**	**0.99**	**1.0**	**0.994**

B. Methods for Comparison

We conduct experiments to evaluate CO2Vec and other baseline models in link prediction task. We consider three types of representation learning methods as baselines: knowledge graph embedding methods, network embedding methods, and order embedding methods, including:

- **TransE** [3]: a strong generic translation-based knowledge graph embedding method for entities and relations of multiple types.
- **RotatE** [22]: the state-of-the-art knowledge graph embedding method, where antisymmetric relations are captured in complex vectors.
- **ComplEx** [25]: a strong complex embeddings method, where antisymmetric relations are captured in complex vectors.
- **GCN** [11]: a strong network embedding method via first-order approximation of spectral graph convolutions.

- **VOE** [26]: a modified vector-based order embedding baseline, which employs the first order dependency to jointly learn order embeddings for multiple types of entities in heterogeneous networks.

C. Link Prediction

To verify the effectiveness of obtained entity embeddings, our evaluation goal is to measure how accurate our model can find the true order relations from a set of entity pairs in given co-ordered network G. Specifically, given a pair of entities, $v_{m,i}$ and $v_{m,j}$, this task aims to quantify the likelihood of $v_{m,i} \rightarrow v_{m,j}$.

Training and Testing Configuration. Given a co-ordered network G, we first apply order embedding models to learn embeddings for each entity in G. Then, we perform link prediction for a set of entity pairs using a scoring function to determine the likelihood that $v_{m,i} \rightarrow v_{m,j}$. The testing set of entity pairs comprises of balanced numbers of unseen ordered entity pairs $P^+ = \{(v_{m,i}, v_{m,j}) | v_{m,i} \rightarrow v_{m,j}\}$ for all $m \in \{A, B\}$, and unseen non-ordered entity pairs $P^- = \{(v_{m,i}, v_{m,j}) | \neg v_{m,i} \rightarrow v_{m,j}\}$. The ranking function used to rank $P = P^+ \cup P^-$ is model-specific, i,e., different for each type of embedding models. For knowledge graph embedding models, e.g., TransE, the closeness between a candidate tail entity and the head entity is defined as the Euclidean distance between the tail entity representation and the head entity representation after the corresponding relation-specific translation. Similarly, the closeness in RotatE is also defined by the distance between a candidate tail entity and the head entity after translation operation. The closeness in ComplEx is defined by translation alike scoring function except the operation is performed on both complex entity and relation representations. For order embedding models, VOE and CO2Vec, the distance from one entity to another is defined by the order violations between two entity representations given in Eq. (2). A special case occurs when many pairs of entities have zero violations, we then use Euclidean distance as a secondary indicator to distinguish the order quality for pairs with zero violations to break ties. Among the pairs with zero violations, the greater Euclidean distance of a pair is, the higher rank the pair is since smaller Euclidean distance of a pair of entities inherently suggests there is barely ordering difference between them. The margin α in Eq. (2) and the support threshold η are set to 1 for all synthetic datasets. We set 16 as the default dimensionality for all methods. As shown in Figure 4, the model-specific loss decreases as entity representation dimensionality increases across all methods, suggests that our loss function does not improve much further beyond dimension of 16. Due to space limitation, we show only the first three synthetic datasets.

Evaluation Metrics. An ideal model is expected not only to give higher scores to ordered entity pairs P^+ over negative ones P^-, but also give higher scores to long-range ordered entity pairs over short-range ones. We define the ground truth ranking $T = \{p_{ij} | p_{ij} = (v_{m,i}, v_{m,j}) \in P\}$ as the list of entity pairs ranked by their dependency range $len(v_{m,i}, v_{m,j})$ in descending order. Note that the dependency length for non-ordered entity pairs without dependency $p_{ij} \in P^-$ is set to 0 to position at the bottom of the list. Let $M = \{p_{ij} | p_{ij} \in P\}$ denote the ranked list of entity pairs returned by model M. Let $r(p_{ij}) \in Z^+$ denote the rank of a ground truth entity p_{ij} by model M. Let the group of entity pairs G=$\{p_{ij} | len(v_{m,i}, v_{m,j}) = \hat{l}\}$ denote the set of entity pairs with the longest dependency range $\hat{l}$. We particular use the group G as an indicator group for a sanity check since ideally the entity pairs in G should be ranked at the top of the list. To measure ranking qualities capturing the dependency range, we report the order-compliant quality in two metrics:(1) Normalized Mean Reciprocal Rank (nMRR@G), and (2) Normalized DCG (NDCG). nMRR@G is formally defined as follow:

$$nMRR@G = \frac{1}{IMRR_G} \sum_{p_{ij} \in G} \frac{1}{r(p_{ij})} \tag{8}$$

where $IMRR_G$ is the ideal MRR on G when the entity pairs with the longest dependency range in G are perfectly ranked ahead of other pair of entities with shorter dependency range. The greater nMRR@G values the better the ranking quality. nMRR@G equals to 1 if a model perfectly returns the indicator group ahead of at the top of the list.

NDCG is a standard metric to take relevance quality into account. The relevance quality in our context is equivalent to the dependency range. Specifically, the Discounted Cumulative Gain (DCG) is formally defined as:

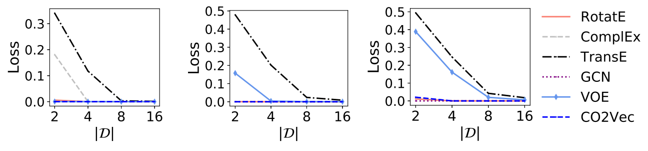
$$DCG = \sum_{p_{ij} \in P} \frac{2^{len(v_{m,i}, v_{m,j})}}{\log(r(p_{ij}) + 1)} \tag{9}$$

for all $p_{ij} \in P$ ranked at $r(p_{ij})$ by a specific model. To make DCG comparable among different models, we normalize DCG to [0,1] by the ideal Discounted Cumulative Gain (IDCG) over the ground truth ranking as follows:

$$IDCG = \sum_{p_{ij} \in T} \frac{2^{len(v_{m,i}, v_{m,j})}}{\log(r_T(p_{ij} + 1)}} \tag{10}$$

for all $p_{ij} \in T$ ranked at $r_T(p_{ij})$ by ground truth. NDCG = $\frac{DCG}{IDCG}$ equals to 1 if a model returns the ground truth ranking.

Results. Table II summarizes our results on the link prediction task. Firstly, the order embedding models, VOE and CO2Vec, demonstrate robust and remarkable capabilities to predict transitive links of longer dependency range atop across synthetic datasets. For instance, The performance gap increases as high as 68.4%/68.2% between CO2Vec (0.99/0.994 NDCG) and TransE (0.312/0.316 NDCG) in DAG(F3,F80) for type-A and type-B link predictions, respectively. This indicates the effectiveness of the order-compliant learning objectives. Secondly, CO2Vec demonstrates remarkable capabilities in precisely predicting links with a wide range of dependency lengths. The length of the set of transitive links in DAG(F3,F80), for instance, range from 2 up to 80 hops. As shown in DAG(F3,F40) and DAG(F3,F80), CO2Vec robustly remains effective compared to VOE. Structural complexity, on the contrary, has been reported to drastically deteriorate the performance of GCN and knowledge graph embedding baselines. Lastly, translation-based knowledge graph embedding methods (e.g., TransE, RotatE, ComplEx) alone do not suffice to approximate transitive links prediction. TransE, despite its simplicity, still outperforms its recently proposed peers with more complex representations.



(a) DAG(F3,L10) (b) DAG(F3,L20) (c) DAG(F3,L40)

Fig. 4: Convergence of training loss at varying dimensions.

TABLE III: Mutual reinforcement study (|D|=16).

Single Removal Setting

DAG (F3,L40)	Model	Type-A		Type-B	
		nMRR@G	NDCG	nMRR@G	NDCG
p=20%	VOE	0.881	0.921	0.989	0.99
	CO2Vec	1.0	1.0	1.0	1.0
p=40%	VOE	0.667	0.774	0.992	0.994
	CO2Vec	1.0	1.0	1.0	1.0
p=60%	VOE	0.373	0.58	0.994	0.994
	CO2Vec	1.0	1.0	1.0	1.0
p=80%	VOE	0.215	0.506	0.998	0.995
	CO2Vec	1.0	0.999	1.0	1.0
p=100%	VOE	0.189	0.497	0.998	0.996
	CO2Vec	1.0	1.0	1.0	1.0

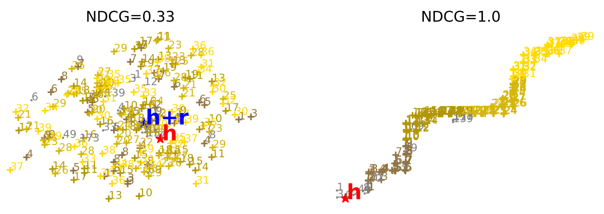
Double Removal Setting

DAG (F3,L40)	Model	Type-A		Type-B	
		nMRR@G	NDCG	nMRR@G	NDCG
p=10%	VOE	0.986	0.989	0.974	0.976
	CO2Vec	1.0	1.0	1.0	1.0
p=20%	VOE	0.918	0.945	0.918	0.937
	CO2Vec	0.994	0.995	0.994	0.995
p=30%	VOE	0.848	0.892	0.858	0.897
	CO2Vec	0.989	0.993	0.991	0.995
p=40%	VOE	0.612	0.752	0.539	0.703
	CO2Vec	0.972	0.979	0.977	0.983
p=50%	VOE	0.304	0.572	0.414	0.638
	CO2Vec	0.95	0.964	0.928	0.956

D. Mutual Reinforcement Study

To answer the research question "*How to quantify the benefit the second order dependency?*", we perform experiments to measure recovery accuracy by holding out order relations with and without complementary evidences obtained by the second order dependency. Specifically, we design two experimental settings for link prediction task to enable the recovery of missing links with varied degrees of challenges: (1) *single removal*, and (2) *double removal*. In the single removal setting, p% relations are held-out from type-A network while type-B network remains intact. In the double removal setting, p% relations are held-out from both type-A and type-B networks. As order embedding models, VOE and CO2Vec, achieve the best performance in previous two experiments, we take a closer look at VOE and CO2Vec to study the impact of complementary evidences.

Results. Table III summarizes our results on link prediction task under both settings. The main findings under single removal setting are two-fold. First, CO2Vec is able to close the performance gap between type-A and type-B link prediction completely by utilizing the second order dependency. The performances of type-B link prediction accomplished by CO2Vec and VOE are close to perfect given full knowledge of the type-B dependency network. Increasing p% held-out



(a) TransE (b) CO2Vec

Fig. 5: Comparison of order embeddings obtained from TransE and CO2Vec models on synthetic network DAG(F3,L40). An example query entity is depicted in red along with its search entities in varying yellow. The lighter color the search entity, the longer the dependency range (i.e., number of hops) from query entity to the search entity in the dependency network. In CO2Vec embedding space, search entities are orderly positioned according to their dependency range w.r.t. the query entity (NDCG=1.0) and thus result in a nice shade from darker yellow (shorter dependency range from query entity) to lighter yellow (longer dependency range from query entity). Varying dependency ranges are contrarily mixed up in TransE embedding space, demonstrating little ordering among search entities (NDCG=0.33).

relations on type-A dependency network starts to reveal notable differences between CO2Vec and VOE. Second, CO2Vec remains perfect as held-out relations (p%) increase, while VOE clearly suffers from the inability to manage type-A knowledge loss. For instance in type-A link prediction task, the performance gap between VOE and CO2Vec increases from 0.079 to 0.503 in NDCG as p increases from 20% to 100%.

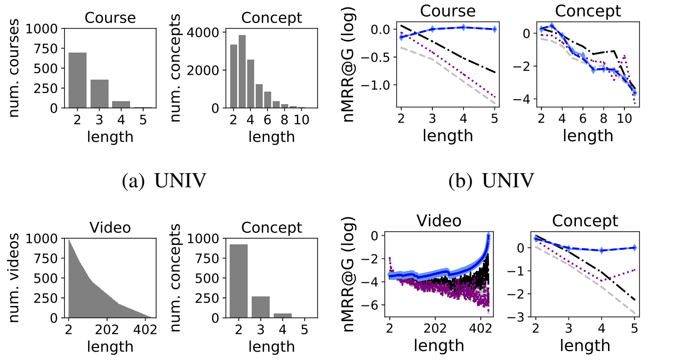
We observe resembling behavior under double removal setting in Table III. First, CO2Vec is robust to withstand double knowledge loss from both type-A and type-B dependency networks. CO2Vec still manages to completely close the performance gap between type-A and type-B link prediction by utilizing second order dependency. This indicates that the second order dependency of both types in CO2Vec mutually complements each other to achieve better performance despite double knowledge loss from both networks. Second, CO2Vec is robust to withstand the increase of knowledge loss (p%). CO2Vec remains close to perfect as held-out relations (p%) increase, while VOE clearly suffers from the inability to manage knowledge loss from both type-A and type-B networks. For instance, the performance gap between VOE and CO2Vec increases from 0.011 to 0.392 in NDCG as p increases from 10% to 50%.

VI. Experiments on Real Datasets

We conduct experiments on real-world co-ordered networks to (1) quantitatively study the effectiveness of co-order embedding in downstream tasks, and (2) qualitatively demonstrate the insights unveiled by co-order embeddings in real data.

TABLE IV: Quantitative results (nMRR@G) for link prediction task on real datasets. Best (second best) of each column are in bold (underlined).

Model	UNIV			MOOC		
	$\lvert D\rvert$	Course	Concept	$\lvert D\rvert$	Video	Concept
RotatE	16	0.0078	0.0001	32	0.0	0.001
ComplEx	16	0.0052	0.0002	32	0.0	0.0013
TransE	16	0.007	0.0002	32	0.0	0.0047
GCN	16	0.005	0.0001	32	0.0	0.0014
VOE	16	0.0485	**0.0004**	32	**1.0**	0.073
CO2Vec	16	**0.083**	0.0002	32	**1.0**	**0.5622**



(a) UNIV (b) UNIV

(c) MOOC (d) MOOC

Fig. 6: Dependency length study on real datasets: dependency length distribution (a)(c), and performance comparison (b)(d). Please refer to Figure 4 for legend description.

A. Datasets

UNIV is a course and concept dataset[2] from 11 US universities, consisting of course dependency hierarchy, concept dependency hierarchy, and associations between courses and concepts. We extract 'dependencies' among courses and 'dependencies' among concepts. In addition, we perform concept matching on course descriptions to establish 'associations' between courses and concepts.

MOOC consists of video dependency hierarchy, concept dependency hierarchy, and associations between videos and concepts. Given a course, a latter video clip 'depends on' an immediate previous video clip in the same course. We obtain such dependencies among video clips for five courses, resulting in five linear chains of video clips with extremely sparse network density (ρ=1×10^{-3}). We observe that a concept 'depends on' 10.6 advanced concepts on average, resulting in dense concept dependency network (ρ=9×10^{-3}). Each video on average 'associates' with 11.8 concepts. The statistics of both datasets are summarized in Table I (in Section V-A).

B. Link Prediction

To verify the effectiveness of obtained entity embeddings, in this experiment, we measure how accurate our model can find the true order relations of various dependency ranges in a given co-ordered network G. Specifically, given a pair of entities, $v_{m,i}$ and $v_{m,j}$, this task aims to quantify the likelihood of $v_{m,i} \rightarrow v_{m,j}$. Given a co-ordered network G, we first apply order embedding models to learn embeddings for each entity in G. We then perform link prediction for a set of entity pairs using model-specific scoring functions to determine the likelihood of $v_{m,i} \rightarrow v_{m,j}$.

Results. The result for link prediction task is reported in Table IV. Firstly, we observe significant performance gap between the order embedding models, VOE and CO2Vec, and other baselines on course dependency network (UNIV) and video dependency network (MOOC) which have sparse dependencies. This suggests that VOE and CO2Vec are able to learn high-quality order embeddings by leveraging complementary evidences from other networks, contrary to other baselines that drastically suffers from the network sparsity. For instance, we observe the performance gap (0.5575 gap in nMRR@G) between CO2Vec (nMRR@G=0.5622) and TransE (nMRR@G=0.0047) for video link prediction in MOOC. Secondly, we observe that CO2Vec performs the best in predicting dependency relations with a longer dependency range as shown in Figures 6(c) and 6(d). For instance, while the video dependency link goes up to 436 hops between two video clips, CO2Vec and VOE manage to rank pairs of video clips with longer dependency range ahead of the rest with shorter dependency ranges (nMRR@G=1).

Dependency Length Study. In this section, we conduct experiment to study the impact of dependency length on the link prediction performance. Firstly, Figures 6(a)6(c) summarize the distribution of dependency length on real datasets at each dependency length. We observe that co-ordered networks may consist of two dependency networks with drastically unbalanced dependency lengths. Take MOOC for instance, the dependency length in video clip deponent network could exceed 400 hops; while the dependency length in concept deponent network is no more than four hops. Secondly, Figure 6(b)6(d) compare the prediction accuracy obtained for different models in nMRR@G at varying dependency length $\hat{l}$. Namely, we compare prediction accuracy for different indicator groups G at corresponding target length $\hat{l}$. The color of each model follows the legend in Figure 4. As shown, VOE (lighblue) and CO2Vec (blue), unlike other baselines (ComplEx, TransE, and GCN), generally achieve better prediction accuracy in nMRR@G for longer-range dependency relations with adequate order consistency provided. This suggests that order-compliant methods indeed are more capable of predicting long-range dependency relations. The only exception is link prediction for concept pairs on UNIV where prediction accuracy decreases for longer dependency relations. This may be due to inadequate order consistency in concept dependency network on UNIV (β_B = 3.5), compared to the concept dependency network on MOOC (β_B = 211.4) On the contrary, increased dependency length presents greater challenges to knowledge graph embedding models (ComplEx, TransE) and network embedding method (GCN) due to their inability to properly encode order relations amongst entities into respective embeddings.

[2]https://github.com/harrylclc/concept-prerequisite-papers

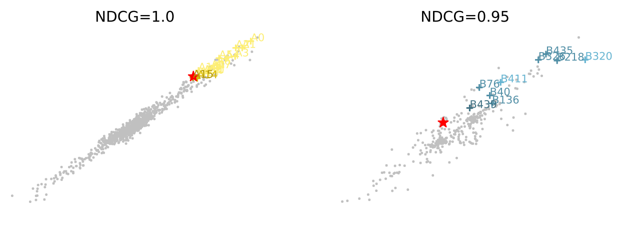


Fig. 7: Prerequisite entity search results by CO2Vec on MOOC. Example query entities $\star$ depicted in red are chosen at random for illustration. We obtain high ordering quality of order embeddings in NDCG, where prerequisite entities of shorter (longer) dependency range are positioned closer (further away) to the query. The lighter (darker) color indicates longer (shorter) dependency range to the query entity.

C. Order-Aware Video Search by Concept

We consider an application scenario of *order-aware video clip search by concepts*, namely *concept query*. Given a concept (e.g., "Heap Sort"), this application aims to compile a list of video clips relevant to "Heap Sort" ranking from basic to advanced levels. To compile the list, we first perform *concept filtering* to collect video clips relevant to "Heap Sort" by exact phrase match. Then, we rank the set of relevant video clips by one of the following functions.

Baseline: We sort the set of relevant video clips according to the video clip ID. Note that as the returned video clips are from five courses, the video clips ordered by video clip ID are generally arranged at random.

Order Embedding (OE): We order the set of relevant video clips using order embeddings by CO2Vec. According to the order embeddings, larger (smaller) order embeddings refer to more basic (advanced) level. We therefore rank relevant video clips with larger (smaller) order embeddings placed atop (bottom) of the list. We generate two search lists using (1) Baseline and (2) OE as ranking functions, respectively, for ten concept queries. We recruit three annotators to judge which search list gives more clear ordering in advance level. Findings demonstrate moderate agreement ($\kappa = 0.535$ on average) between human versus the automated raters. 0.83 of the search results by OE are deemed better than those by Baseline according to three annotators on average. Examples of search lists are summarized in Table VI. Overall, we find that Baseline in Table VI only gives clear ordering for video lectures within a single course, while the ordering among three difference courses (ML1, ML2, DS1) are not clear. OE, on the other hand, manages to figure out some video clips in one course are far more advanced than video clips from other courses. For instance, the video clip ML1-85 in Table VI contains PCA-related concepts, is far more advanced than video clips from other courses (ML2-162, ML2-157) and thus is placed at the fourth position by OE.

TABLE V: Prerequisite entity search results in Figures 7(a) and 7(b). (l := number of hops to the query entity $\star$).

Prerequisite Video Clips	l	**Prerequisite Concepts**	l
Cost Function Intuition I	$\star$	Logistic Function	$\star$
Cost Function	1	Loss Function	1
Model Representation	2	Parameter Vector	2
Unsupervised Learning	3	Machine Learning Algorithm	2
Supervised Learning	4	Optimization Objective	2
Welcome	5	Model Parameters	2
Welcome to Machine Learning	6	Optimization Algorithms	3
		Machine Learning	2
		Optimization Problem	2
		Partial Derivative	3

D. Prerequisite Entity Search

We consider an application scenario of *prerequisite entity search*, namely *prerequisite query*. Given a query entity, this application aims to return a list of prerequisite entities of the same type as query to from basic to advanced levels. Table VII gives examples of recommended prerequisite courses returned by TransE, VOE, and CO2Vec given the course entity "Software Engineering" as query. CO2Vec not only recovers the most of the prerequisite courses, but also presents better ordering, from the basic to advanced levels, among them, e.g., "Introduction to Programming I" followed by "Introduction to Programming II". This validates the superiority of CO2Vec in learning higher quality order representations. Figure 7 shows the result of prerequisite entity search in two types of entity queries on MOOC datasets by CO2Vec, including prerequisite video clip and prerequisite concept queries. As shown in Figures 7(a) and 7(b), both prerequisite video clip and prerequisite concept queries capture the ordering amongst search entities nicely from shorter dependency range to longer dependency range from query entity (NDCG=1.0 and NDCG=0.95). The details of search results for respective queries in Figures 7(a) and 7(b) are summarized in Table V. The query is depicted as $\star$. As shown, the search result (ordered based on their positions to the query in the order embedding space) is highly correlated with the dependency range (l).

VII. Conclusion

We present a generic order embedding model, CO2Vec, aiming to jointly learn order-preserving representations for entities in co-ordered networks. A novel second order dependency for extracting complementary evidences to enhance order representations is proposed. We design a comprehensive evaluation framework to study the quality of CO2Vec representation. Our experiments show that: (1) CO2Vec captures order semantics of entities in co-ordered networks very well and outperforms other strong baselines in link prediction task, and (2) CO2Vec offers a more robust representation by exploring mutual reinforcement via associations among different types of entities. Furthermore, we demonstrate two useful applications driven by co-order embeddings. For the future work, we plan to integrate entity semantics aspect in co-order embedding learning. Another interesting direction is to explore deep learning techniques to enhance co-order embeddings.

TABLE VI: Search results of concept query by (1) video ID, and (2) order embeddings. (Query Concept='Square Matrix').

ID	Search Results 1 (Baseline)	Covered Concepts
ML1-17	Inverse and Transpose	Matrix Transpose, Linear Algebra, Training Data, Machine Learning, Learning Method, Square...
ML1-85	Choosing the Number of Principal Components	Training Data, PCA Algorithm, Covariance Matrix, Principle Components, Square...
ML2-157	Rewriting the Single Observation Model in Vector Notation	Linear Algebra, Square Matrix, Inner Product, Matrix, Gradient Descent, Regression Model...
ML2-162	Discussing the Closed Form Solution	O Notation, Square Matrix, Matrix, Features, Big-o Notation
DS1-59	Symbol Table Applications Sparse Vectors Optional	Vector Multiplication, Square Matrix, Matrix, Matrix Multiplication, Matrix Vector Multiplication

ID	Search Results 2 (OE)	Covered Concepts
ML1-17	Inverse and Transpose	Matrix Transpose, Linear Algebra, Training Data, Machine Learning, Learning Method, Square...
ML2-162	Discussing the Closed Form Solution	O Notation, Square Matrix, Matrix, Features, Big-o Notation
ML2-157	Rewriting the Single Observation Model in Vector Notation	Linear Algebra, Square Matrix, Inner Product, Matrix, Gradient Descent...
ML1-85	Choosing the Number of Principal Components	Training Data, PCA Algorithm, Covariance Matrix, Principle Components, Square...
DS1-59	Symbol Table Applications Sparse Vectors Optional	Vector Multiplication, Square Matrix, Matrix, Matrix Multiplication, Matrix Vector Multiplication

TABLE VII: Search results by prerequisite query on UNIV dataset. Italic blue indicates the true prerequisite search entities.

Query Entity: Software Engineering, **Order Relation** ($\rightarrow$): 'is prerequisite of'.

Rank	TransE	VOE	CO2Vec
1	Computer Networks	*Introduction to Programming I*	*Introduction to Programming I*
2	*Algorithms and Data Structures*	*Discrete Structures in Computer Science*	*Introduction to Programming II*
3	Mobile Application Development	*Introduction to Programming II*	*Discrete Structures in Computer Science*
4	*Introduction to Programming II*	Computing Concepts and Competencies	*Algorithms and Data Structures*
5	Operating Systems	Computer Networks	Computer Networks

ACKNOWLEDGMENT

This research / project is supported by the National Research Foundation, Singapore under its International Research Centres in Singapore Funding Initiative, and in part by National Science Foundation, USA under Grant No. IIS-1717084. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of National Research Foundation, Singapore, and National Science Foundation, USA.

REFERENCES

[1] F. ALSaad, A. Boughoula, C. Geigle, H. Sundaram, and C. Zhai, "Mining mooc lecture transcripts to construct concept dependency graphs," in *EDM*, 2018.

[2] B. Athiwaratkun and A. G. Wilson, "Hierarchical density order embeddings," in *ICLR*, 2018.

[3] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," in *NIPS*, 2013.

[4] H. Cai, V. W. Zheng, and K. Chang, "A comprehensive survey of graph embedding: problems, techniques and applications," *TKDE*, 2018.

[5] Y. Chen, J. P. González-Brenes, and J. Tian, "Joint discovery of skill prerequisite graphs and student models," in *EDM*, 2016.

[6] M.-F. Chiang, E.-P. Lim, W.-C. Lee, X. J. S. Ashok, and P. K. Prasetyo, "One-class order embedding for dependency relation prediction," in *SIGIR*, 2019.

[7] T. Dettmers, M. Pasquale, S. Pontus, and S. Riedel, "Convolutional 2d knowledge graph embeddings," in *AAAI*, 2018.

[8] Y. Dong, N. V. Chawla, and A. Swami, "metapath2vec: Scalable representation learning for heterogeneous networks," in *KDD*, 2017.

[9] T.-y. Fu, W.-C. Lee, and Z. Lei, "Hin2vec: Explore meta-paths in heterogeneous information networks for representation learning," in *CIKM*, 2017.

[10] A. Grover and J. Leskovec, "Node2vec: Scalable feature learning for networks," in *KDD*, 2016.

[11] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *ICLR*, 2016.

[12] C. Liang, J. Ye, S. Wang, B. Pursel, and C. L. Giles, "Investigating active learning for concept prerequisite learning," in *EAAI*, 2018.

[13] C. Liang, J. Ye, Z. Wu, B. Pursel, and C. L. Giles, "Recovering concept prerequisite relations from university course dependencies," in *AAAI*, 2017.

[14] Y. Lin, Z. Liu, M. Sun, Y. Liu, and X. Zhu, "Learning entity and relation embeddings for knowledge graph completion," in *AAAI*, 2015.

[15] H. Liu, W. Ma, Y. Yang, and J. Carbonell, "Learning concept graphs from online educational data," *JAIR*, vol. 55, 2016.

[16] J. Minker, "On indefinite databases and the closed world assumption," in *CADE*, 1982.

[17] M. Nickel, K. Murphy, V. Tresp, and E. Gabrilovich, "A review of relational machine learning for knowledge graphs," *Proceedings of the IEEE*, vol. 104, no. 1, pp. 11–33, 2016.

[18] M. Nickel, L. Rosasco, and T. A. Poggio, "Holographic embeddings of knowledge graphs," in *AAAI*, 2016.

[19] L. Pan, C. Li, J. Li, and J. Tang, "Prerequisite relation learning for concepts in moocs," in *ACL*, 2017.

[20] L. Pan, X. Wang, C. Li, J. Li, and J. Tang, "Course concept extraction in moocs via embedding-based graph propagation," in *IJCNLP*, 2017.

[21] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: Online learning of social representations," in *KDD*, 2014.

[22] Z. Sun, Z.-H. Deng, J.-Y. Nie, and J. Tang, "Rotate: Knowledge graph embedding by relational rotation in complex space," in *ICLR*, 2019.

[23] J. Tang, M. Qu, and Q. Mei, "Pte: Predictive text embedding through large-scale heterogeneous text networks," in *KDD*, 2015.

[24] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "Line: Large-scale information network embedding," in *WWW*, 2015.

[25] T. Trouillon, J. Welbl, S. Riedel, É. Gaussier, and G. Bouchard, "Complex embeddings for simple link prediction," in *ICML*, 2016.

[26] I. Vendrov, R. Kiros, S. Fidler, and R. Urtasun, "Order-embeddings of images and language," in *ICLR*, 2016.

[27] L. Vilnis, X. Li, S. Murty, and A. McCallum, "Probabilistic embedding of knowledge graphs with box lattice measures," in *ACL*, 2018.

[28] L. Vilnis and A. McCallum, "Word representations via gaussian embedding," in *ICLR*, 2015.

[29] Q. Wang, Z. Mao, B. Wang, and L. Guo, "Knowledge graph embedding: A survey of approaches and applications," *TKDE*, vol. 29, no. 12, pp. 2724–2743, 2017.

[30] B. Yang, W.-t. Yih, X. He, J. Gao, and L. Deng, "Learning multi-relational semantics using neural-embedding models," *arXiv preprint arXiv:1411.4072*, 2014.