Neural Concept Formation in Knowledge Graphs

Anonymous authors

Abstract

In this work, we investigate how to learn novel concepts in Knowledge Graphs (KGs) in a principled way, and how to effectively exploit them to produce more accurate neural link prediction models. Specifically, we show how concept membership relationships learned via unsupervised clustering of entities can be reified and used to augment a KG. In a thorough set of experiments we confirm that neural link predictors trained on these augmented KGs, or in a joint Expectation-Maximization iterative scheme, can generalize better and produce more accurate predictions for infrequent relationships. For instance, our method yields relative improvements of up to 8.6% MRR on WN18RR for rare predicates, and up to 82% in small-data regimes, where the model has access to just a small subset of the training triples. Furthermore, our proposed models are able to learn meaningful concepts.

1. Introduction

One of the most remarkable aspects of human intelligence is arguably the capacity to abstract and summarize knowledge into *concepts*. It is believed to play a central role in allowing humans to learn quickly from few examples [Lake et al., 2015] and to robustly generalize to unseen data [Pothos and Chater, 2002, Lakoff and Johnson, 1980, Rosch et al., 1976]. It is no wonder that many machine learning and knowledge representation methods have tried to "reverse-engineer" how humans learn concepts [Tenenbaum, 2018, Hassabis et al., 2017] in order to automate reasoning as well as knowledge base construction [Kok and Domingos, 2007, Kemp et al., 2006, Xu et al., 2006]. Among the most prominent knowledge representation formalisms, there are *Knowledge Graphs* (KGs) – graph-structured knowledge bases where knowledge about the world is encoded in the form of relationships between entities. KGs encode facts about entities and the relationships between them (edges) as *subject-predicate-object* triples, each denoting a relationship of type *predicate* between the *subject* and *object* of the triple. A fundamental task in the construction of KGs is *link prediction*, which consists of identifying missing edges between entities in the KG.

Neural link predictors are a class of link prediction models achieving state-of-the-art results on several link prediction benchmarks while being able to scale to very large KGs. Neural link predictors learn embedding representations for each entity and relation in the

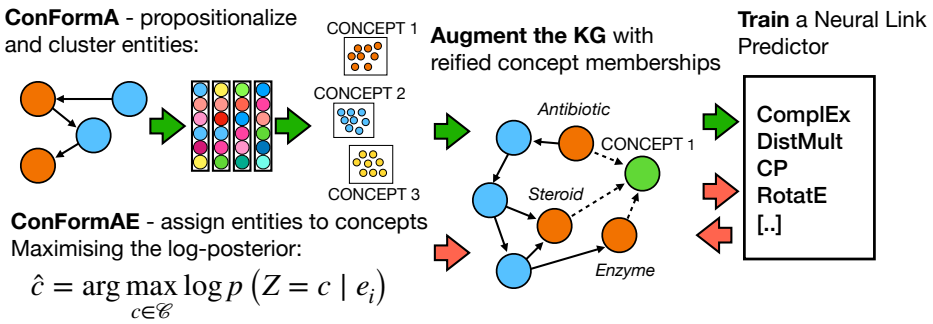

Figure 1: High-level visualization of ConFormA and ConFormAE: entities are propositionalized and clustered in *concepts*, and concept memberships are added to the KG as triples. In ConFormAE, concept memberships are iteratively revised to maximize the likelihood of the data.

KG via back-propagation [Nickel et al., 2016]. However, neural link predictors are known not to be accurate in the presence of sparse KGs, i.e., when entities appear only in few triples [Pujara et al., 2017], also referred to as the *cold-start problem*, and may not be able to learn patterns involving *groups of entities* [Evans and Grefenstette, 2018].

In this work, we propose to *learn concepts* in neural link predictors as a principled way to elicit discrete latent information that can alleviate the generalization issues of existing models. Moreover, learning new concepts and turning them into entities can help to automate the construction of KGs. Specifically, we make the following contributions. First, we formalize concept learning as an unsupervised clustering step over entities in a KG. We do this first by *reifying* concept membership relationships into KG facts and by incorporating them in the KG, in a process called *KG augmentation*. Training out-of-the-box neural link predictors on these augmented KGs improves their accuracy. Secondly, we introduce a single, principled probabilistic framework for jointly learning concept memberships and neural link prediction models at once, by maximizing the likelihood of the KG triples via an Expectation-Maximization scheme. Lastly, we execute a rigorous empirical evaluation on several real-world KG benchmarks, showing that both of our approaches, named ConFormA and ConFormAE, are capable of learning semantically meaningful concepts. We observe that explicitly augmenting the KG with the newly-learned concepts can improve generalization over rare predicates by up to 8.6% in terms of Mean Reciprocal Rank (MRR) on WN18RR and 2.1% on FB15K-237. Furthermore, we perform a sparsification analysis where neural link predictors are trained on only a small subset of the training set, showing that ConFormA and ConFormAE can achieve up to 82% relative improvement on WN18RR and 21% on FB15K-237 when trained on only 5% of the data. This highlights the potential of our approach for addressing the cold-start problem in sparse KGs.

2. Background

Let a KG $\mathcal{G}$ be represented as a set of N triples, i.e., $\mathcal{G} = \{\langle s, r, o \rangle_i\}_{i=1}^{N} \subseteq \mathcal{E} \times \mathcal{R} \times \mathcal{E}$ where $\mathcal{E} = \{e_i\}_{i=1}^{N_e}$ is the set of subject (s) and object (o) entities, and $\mathcal{R} = \{r_i\}_{i=1}^{N_r}$ the

Algorithm 1 ConFormA($\mathcal{G}$, N_c, n)

Input: KG $\mathcal{G}$, no. of clusters N_c, no. of epochs n
Output: Parameters $\mathbf{\Theta}$, cluster memberships $\mathcal{S} = \{S_c \mid c \in \mathcal{C}\}$
$\mathbf{P} \leftarrow \mathsf{propositionalization}(\mathcal{G})$ ▷ E.g., random paths
$\mathcal{S} \leftarrow \mathsf{Clustering}(\mathbf{P}, N_c)$ ▷ E.g., spectral clustering
$\mathcal{G}' \leftarrow \mathcal{G} \cup \{\langle e_i, \text{ISA}, c_j \rangle \mid e_i \in S_c, c \in \mathcal{C}\}$ ▷ Create an augmented KG
$\mathbf{\Theta} \leftarrow \mathsf{init}()$
for n epochs **do** ▷ Train the parameters $\mathbf{\Theta}$ of a neural link predictor on $\mathcal{G}'$
  $\mathbf{\Theta} \leftarrow \mathsf{train}(\mathcal{G}', \mathbf{\Theta})$
return $\mathbf{\Theta}, \mathcal{S}$

set of relation types (r). Neural link predictors can be framed as learning a k-dimensional representation, i.e., an *embedding* vector $\mathbf{e} \in \mathbb{C}^k$, for all entities in $\mathcal{E}$ appearing in $\mathcal{G}$. Given a triple $\langle s, r, o \rangle \in \mathcal{E} \times \mathcal{R} \times \mathcal{E}$, a neural link predictor defines a scoring function $\phi_r : \mathbb{C}^k \times \mathbb{C}^k \mapsto \mathbb{R}$ that, given the embedding representations $\mathbf{e}_s \in \mathbb{C}^k$ and $\mathbf{e}_o \in \mathbb{C}^k$ of the subject s and the object o of the triple, returns the score $\phi_r(\mathbf{e}_s, \mathbf{e}_o) \in \mathbb{R}$ that s and o are related by the relation type r. The scoring function ϕ_r implicitly defines a probability distribution p over triples. Let x_{sro} be a binary random variable denoting the existence of the triple $\langle s, r, o \rangle$, then we have that the probability of observing the triple is proportional to its score, i.e., $p(x_{sro}) \propto \phi_r(\mathbf{e}_s, \mathbf{e}_o)$ [Nickel et al., 2016].

Let $\mathbf{E} \in \mathbb{C}^{N_e \times k}$ and $\mathbf{W} \in \mathbb{C}^{N_r \times k}$ be the relation and entity embeddings. Then, learning a neural link predictor from data consists of finding the best set of parameters $\mathbf{\Theta} = \{\mathbf{E}, \mathbf{W}\}$ that solve the optimization problem $\arg\min_{\mathbf{\Theta}} \mathcal{L}(\mathcal{G}; \mathbf{\Theta}) + \lambda \Omega(\mathbf{\Theta})$, where $\mathcal{L}(\mathcal{G}; \mathbf{\Theta})$ denotes a model-specific loss, typically proportional to the negative log-likelihood of the triples [Trouillon et al., 2016], and $\Omega(\cdot)$ is a regularization term, such as the L_2 or the nuclear weighted 3-norm [N3, Lacroix et al., 2018], whose weight is determined by a coefficient $\lambda \geq 0$. In the following section, we discuss how this learning loop for neural link predictors can be extended to learn concepts in a KG, *while treating the link predictor model as a black-box.*

3. ConFormA: Learning Concepts by Augmenting Knowledge Graphs

Given a KG $\mathcal{G}$, *concept learning* means to identify sets of entities $S_1, \ldots, S_{N_c} \subseteq \mathcal{E}$ that are semantically related and can be *abstracted* into concepts $c_1, \ldots, c_{N_c} \in \mathcal{C}$. We aim at finding a *partitioning* of entities $\mathcal{S} = \{S_i\}_{i=1}^{N_c}$, such that each entity is assigned to a single concept at a time, i.e., $\forall S_i, S_j \in \mathcal{S} \rightarrow S_i \cap S_j = \emptyset$, and $\bigcup_{S \in \mathcal{S}} S = \mathcal{E}$. To this end, a natural solution is to perform a *hard clustering* of the entities in the KG $\mathcal{G}$. Then, we can reify the cluster membership relations, i.e., introducing the concepts $c_1, \ldots, c_{N_c}$ as *new entities*, and materializing the concept membership relations as *new triples*. We refer to this process as *KG augmentation*. Algorithm 1 summarizes our framework, which can be instantiated for different clustering and neural link prediction models. We name it Concept Formation via Augmentation (ConFormA). Next, we discuss *how* to perform these two steps, and the reasons *why* neural link predictors can benefit from being trained on augmented KGs.

Clustering entities. Ideally, the clustering step in ConFormA could be performed by any relational clustering algorithm. However, classical probabilistic approaches such as

Algorithm 2 ConFormAE($\mathbf{\Theta}, \mathcal{S}, N_c, n, t$)

Input: no. of clusters N_c, initial parameters $\mathbf{\Theta}$, no. of epochs n, and no. of iterations t.
Output: Updated parameters $\mathbf{\Theta}$, cluster memberships $\mathcal{S} = \{S_c \mid c \in \mathcal{C}\}$
for t iterations **do**
  for $c \in \mathcal{C}$ **do** ▷ Initialise the cluster memberships
    $S_c \leftarrow \emptyset$
  for $e \in \mathcal{E}$ **do**
    $\hat{c} \leftarrow \arg\max_{c \in \mathcal{C}} \phi_{\text{ISA}}(\mathbf{e}_e, \mathbf{e}_c)$
    $S_{\hat{c}} \leftarrow S_{\hat{c}} \cup \{e\}$
  $\mathcal{G}' \leftarrow \mathcal{G} \cup \{\langle e, \text{ISA}, c\rangle \mid e \in S_c, c \in \mathcal{C}\}$ ▷ E-step: Make hard assignments
  for n epochs **do** ▷ M-step: Refine the neural link prediction model
    $\mathbf{\Theta} \leftarrow \text{train}(\mathcal{G}', \mathbf{\Theta})$
return $\mathbf{\Theta}, \mathcal{S}$

statistical predicate invention [Kok and Domingos, 2007] and stochastic block models [Kemp et al., 2006, Xu et al., 2006] would hardly scale to modern KGs with hundreds of thousands of entities. This poses a challenge also to kernel-based approaches [Blondel et al., 2008, de Vries, 2013, Morris et al., 2017]. To overcome this issue, we opt for a more computationally efficient alternative: we first *propositionalize* entities into d-dimensional embedding vectors [Kramer et al., 2001] and then employ a propositional clustering algorithm – e.g., spectral clustering [Ng et al., 2001] or K-means – over this now tabular representation $\mathbf{P} \in \mathbb{R}^{N_e \times d}$.

We find that executing multi-hop random paths in $\mathcal{G}$, as proposed by Das et al. [2020], provides scalable and accurate entity representations. We also explored clustering directly over the embeddings learned by a neural link prediction model but with scarce link prediction improvements. This could be due to the latent concept information which we explicitly introduce via augmentation being already captured by the neural link predictor embeddings.

Knowledge Graph Augmentation. Given the set $\mathcal{S}$, we reify the cluster membership relations by materializing new triples to augment $\mathcal{G}$. Specifically, for each entity e participating in a cluster $c \in \mathcal{C}$ we create a new triple of the form $\langle e, \text{ISA}, c\rangle$, where c is a new entity denoting the j-th concept, and ISA is a freshly introduced relation denoting concept memberships.

Let $\mathcal{G}'$ denote the augmented KG, i.e., $\mathcal{G}' \leftarrow \mathcal{G} \cup \{\langle e, \text{ISA}, c\rangle \mid e \in S_c, c \in \mathcal{C}\}$, where S_c is the set of entities assigned to concept c. Learning a neural link predictor simply requires calling its usual training routine (lines 6-8 of Algorithm 1) over the augmented KG $\mathcal{G}'$. Specifically, its set of parameters $\mathbf{\Theta}' = \{\mathbf{E}, \mathbf{W}, \mathbf{C}\}$, which now includes the concept embeddings $\mathbf{C} \in \mathbb{C}^{N_c \times k}$ for the newly introduced concept entities $\mathcal{C} = \{c_1, \ldots, c_{N_c}\}$, can be updated by minimizing the neural link predictor's loss function, along its other parameters.

ConFormA is likely to improve the generalization of a neural link predictor for the following reasons. Firstly, this kind of augmentation acts as injecting background knowledge that does not need to be learned from scratch, akin to when inverse relation triples [Lacroix et al., 2018, Kazemi and Poole, 2018] or hierarchical relation information [Zhang et al., 2018] are explicitly added to KGs. Secondly, they help to make very sparse KGs more dense, tackling the sparsity issues in neural link predictors [Pujara et al., 2017].

4. ConFormAE: Jointly learning Concepts and Embeddings

ConFormA is a flexible framework: it can be customized with any propositionalization and clustering routines, and wrapped around any neural link prediction model. A natural question then arises: *is it possible to automatically devise a propositionalization scheme that enhances clustering and embedding quality, that is, to jointly learn both the concepts and the embeddings?* Ideally, we could cast this as a joint optimization problem to maximize the marginal log-likelihood of the triples in $\mathcal{G}$, where marginalization is performed over some latent variable Z denoting the cluster assignments, i.e., having values $c \in \mathcal{C}$. As directly maximizing this marginal likelihood is intractable, we adopt an iterative Expectation-Maximization (EM)-like scheme [Dempster et al., 1977]. Algorithm 2 summarizes the whole process, which we name ConFormAE – Concept Formation with Augmentation via EM. We next discuss in detail how to design the expectation (E) and maximization (M) steps efficiently.

E-step. Let $p(Z \mid e)$ denote the distribution over concept-memberships induced by the neural link predictor for the entity e. Recall from Section 2 that the probability of assigning entity e to concept c is proportional to the score assigned to the reified triple (Section 3) encoding its concept membership, i.e., $p(Z = c \mid e) \propto \phi_{\text{ISA}}(\mathbf{e}_e, \mathbf{e}_c)$, where $\mathbf{e}_e, \mathbf{e}_c$ denote the embeddings of e and c, respectively. Exactly computing all the cluster memberships $p(Z = c \mid e)$ for each entity e and concept $c \in \mathcal{C}$ is a hard problem, since we would need to compute an intractable partition function. We therefore resort to compute *hard cluster assignments*, a practical approximation commonly adopted in many *hard-EM variants* [Samdani et al., 2012, Kok and Domingos, 2007]. That is, we are interested in solving $\hat{c} = \arg\max_{c \in \mathcal{C}} p(Z = c \mid e_i)$ for each entity e_i. Note that this can be done exactly and efficiently as $\hat{c} = \arg\max_{c \in \mathcal{C}} \phi_{\text{ISA}}(\mathbf{e}_e, \mathbf{e}_c)$ and therefore it reduces to predicting the most probable link between the entity e and a concept in $\mathcal{C}$.

M-step. The aim of this step is to find the best set of parameters $\mathbf{\Theta}'$ for a neural link predictor by maximizing its log-likelihood, i.e., $\mathbb{E}_{c_e \sim p(\cdot \mid e)} \left[\log p(X) + \sum_{e \in \mathcal{E}} \log p(Z = c_e \mid e) \right]$ where $\log p(X)$ here compactly denotes the likelihood of the data in $\mathcal{G}$, and $c_e \in \mathcal{C}$ denotes the concept associated with the entity $e \in \mathcal{E}$. This quantity can be efficiently approximated via our reification and augmentation scheme. In fact, at the end of the E-step, we had retrieved the clustering $\mathcal{S}$ (as in ConFormA, cf. Section 3). Therefore, to find $\mathbf{\Theta}'$ we can simply train the neural link prediction model for a certain number of epochs n over the augmented KG $\mathcal{G}'$.

We refer to Appendix A for an in-depth analysis of the time and space complexity of ConFormA and ConFormAE.

5. Related Work

Concept Learning for Relational Data. Relational clustering, sometimes known as statistical predicate invention [Kok and Domingos, 2007], has been addressed in different communities and under different relational formalisms. For instance, the Infinite Relational Model (IRM) [Kemp et al., 2006], is a Bayesian non-parametric interaction method for detecting community of users in networks, later extended to multiple relation types [Xu et al., 2006]. The Multiple Relational Clusterings model (MRC) [Richardson and Domingos, 2006] extends the IRM to learning multiple cross-cutting clusterings, i.e. allowing each object

Table 1: MRR and Hits at K (H@K) for CONFORMA and CONFORMAE when using DistMult or ComplEx as baselines on WN18RR and FB15K-237 for different values of embedding size (k). Each configuration was repeated with 30 random seeds, and we report the means of each metric. For assessing whether the MRR values are significantly higher than the baseline, we used a one-sided Wilcoxon signed-rank test, where ▲ (resp. △) denotes a p-value ≤ 0.01 (resp. 0.1).

			COMPLEX				DISTMULT			
	k	MODEL	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10
WN18RR	500	BASELINE	48.06	43.71	49.73	56.57	44.23	40.21	45.34	52.72
		CONFORMA	48.48 ▲	43.83	**50.23**	**57.65**	44.33 △	40.15	**45.36**	**53.19**
		CONFORMAE	**48.51** ▲	**44.05**	50.13	57.40	**44.36** ▲	**40.29**	45.28	53.10
	1000	BASELINE	48.58	44.18	50.21	57.16	44.53	40.34	45.68	53.20
		CONFORMA	**49.13** ▲	44.43	**50.82**	**58.57**	**45.59** ▲	**41.25**	**46.76**	**54.72**
		CONFORMAE	48.99 ▲	**44.45**	50.60	58.12	45.15 ▲	40.83	46.35	54.31
	2000	BASELINE	48.81	44.39	50.41	57.45	**45.17**	**40.89**	**46.49**	53.91
		CONFORMA	**49.28** ▲	**44.63**	**50.94**	**58.73**	44.92	40.59	46.08	53.76
		CONFORMAE	49.14 ▲	**44.63**	50.75	58.22	44.96	40.65	46.04	**53.92**
FB15K237	500	BASELINE	35.99	26.60	**39.54**	54.90	34.82	25.52	38.24	53.64
		CONFORMA	35.97	26.55	39.50	**54.98**	34.86 ▲	25.55	38.29	**53.68**
		CONFORMAE	**36.06** ▲	**26.66**	39.53	55.08	**34.95** ▲	**25.67**	**38.29**	**53.68**
	1000	BASELINE	36.11	26.68	39.65	55.15	34.95	25.61	38.42	53.73
		CONFORMA	36.20 △	26.69	39.71	55.20	35.32 ▲	25.59	38.50	**53.89**
		CONFORMAE	**36.25** ▲	**26.72**	**39.72**	**55.28**	**35.35** ▲	**25.62**	**38.52**	53.88
	2000	BASELINE	36.26	26.83	39.79	55.33	35.39	25.99	38.86	54.37
		CONFORMA	36.31 ▲	26.86	**39.86**	55.39	35.49 ▲	26.11	38.89	**54.49**
		CONFORMAE	**36.35** ▲	**26.95**	39.84	**55.44**	**35.50** ▲	**26.12**	**38.94**	54.48

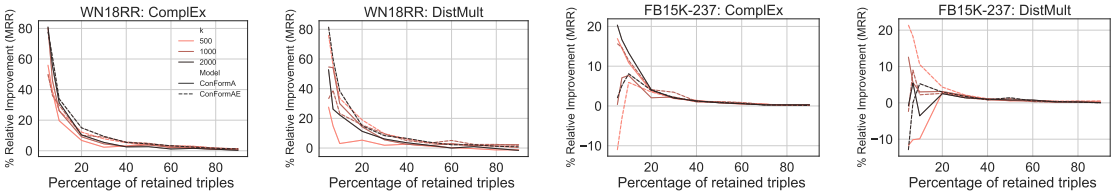

Table 2: Relative improvement in MRR for CONFORMA and CONFORMAE upon the baseline models – either DistMult or ComplEx – on sparsified WN18RR and FB15K-237 KGs for different values of embedding size (k). Each experiment was repeated with 6 different random seeds.

to belong to more than one cluster, under the general framework of Markov logic networks (MLNs). The above approaches (and their variants) would hardly scale to the KGs we employ in our experiments. To see why, consider that to train either the IRM or the MRC on a small knowledge graph such as UMLS [McCray, 2003] (135 entities, see next Section) took 10 hours, while both our algorithms require less than 3 minutes.

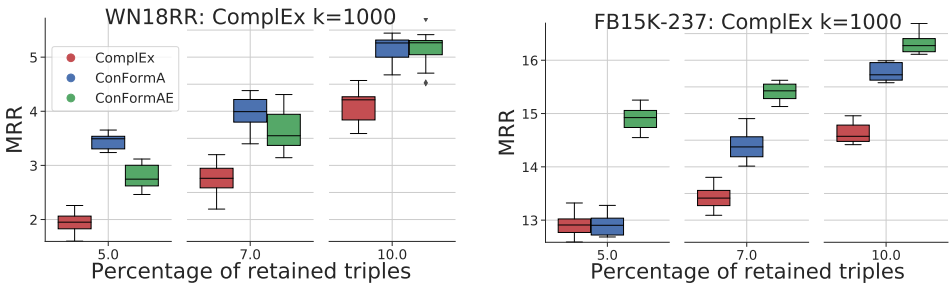

Figure 2: MRR of ConFormA, ConFormAE, and ComplEx on sparsified WN18RR and FB15K-237 KGs, where the percentage of training triples is in $\{5, 7, 10\}$ for different values of embedding size (k). Each experiment was repeated with 6 different random seeds. Results for other baselines and embedding sizes are shown in Table 17.

Concept learning for KGE models. In a preliminary study, Nickel and Tresp [2011] explore how to reconstruct a taxonomy over entities by performing hierarchical clustering of entity embeddings. Differently from our work, they do not learn embedding representations for the learned concepts nor use concept learning to improve link prediction on the original KG. Zhang et al. [2018], on the other hand, proposed learning and explicitly modeling taxonomies over relations. To do so, they cluster relation embeddings in a three-layer hierarchy. Our augmentation approach, while focusing on entity concepts, can be generalized to relations as well, after reifying them.

Concept Learning for Deep Graph Classification. Ying et al. [2018] introduce a differentiable graph pooling module in graph neural networks (GNNs) to for perform hierarchical clustering of nodes in graphs. While improving the accuracy over small-scale graph classification benchmarks, their work cannot be readily adapted to link prediction on KGs and would not scale to large benchmark KGs, such as those used in our experiments.

Data Augmentation for Link Prediction. Various KG augmentation schemes have been proposed in the neural link prediction literature. E.g., Lacroix et al. [2018] show that introducing reciprocal relations as explicit triples greatly enhances the performance of KGE models on many benchmarks. Minervini et al. [2017], instead, temporarily augment a KG during training by generating sets of adversarial examples that maximize an inconsistency loss encoding certain background knowledge.

6. Experiments

In this Section, we aim to answer the following research questions: **Q1**) are the concepts learned by ConFormA and ConFormAE semantically-meaningful?, **Q2**) can unsupervised concept learning boost neural link prediction performance?, **Q3**) can concept reification help alleviate the cold-start problem in sparse in KGs?, and **Q4**) how does augmentation impact generalization over rare relation types?. We proceed by outlining our experimental setting.

Datasets. We perform experiments on three datasets: WN18RR [Dettmers et al., 2018] and FB15K-237 [Toutanova and Chen, 2015] – two large benchmark KGs and UMLS [McCray, 2003] – a small biomedical KG. Summary statistics of the entity and relation distributions

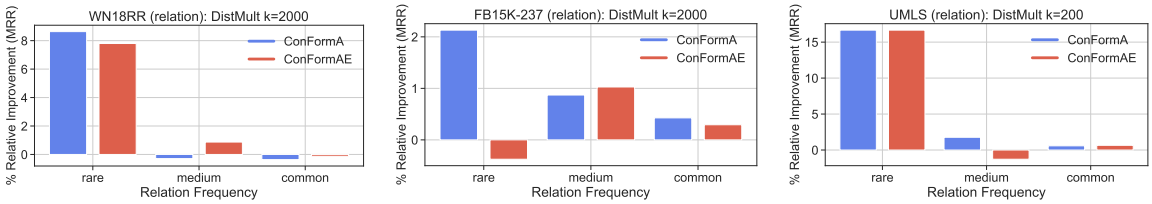

Figure 3: Relative improvement on predicate sub-populations in terms of MRR achieved by ConFormA and ConFormAE over the DistMult baseline on WN18RR and FB15K-237 for $k = 2000$, and on UMLS for $k = 200$ using bins from Table 19.

for each dataset are shown in Table 18. The two large KGs come with unique challenges – in FB15K-237, the number of predicates is relatively high (Table 18), and it may be difficult to jointly model all of them. In WN18RR, on the other hand, most entities are sparsely represented in the training set. As in [Pujara et al., 2017], we consider the sparsity of each graph by computing the *entity density* (ED) and *relation density* (RD), i.e., the average number of triples per entity or relation: RD $= |\mathcal{T}|/|\mathcal{R}|$, ED $= 2|\mathcal{T}|/|\mathcal{E}|$ where $|\mathcal{T}|$ is the number of train triples. We note that the entity density in WN18RR is extremely low, with each entity occurring on average in only four triples (Table 18).

Baselines. To investigate the ability of ConFormA and ConFormAE to work with different out-of-the-box neural link predictors, we employ two different baselines: ComplEx [Trouillon et al., 2016] and DistMult [Yang et al., 2015]. For all experiments we used the nuclear N3 norm [Lacroix et al., 2018] as a regularizer, the standard multi-class loss proposed by Lacroix et al. [2018], and the AdaGrad optimizer [Duchi et al., 2011]. Hyperparameter values can be found in Appendix F. We trained each model till convergence for 100 epochs and computed the filtered Mean Reciprocal Rank (MRR) and HITS@K [Bordes et al., 2013] every 3 epochs on the validation and test sets. The highest validation performance was extracted and the corresponding test performance was reported.

ConFormA. We use the propositionalization scheme leveraging random paths proposed by Das et al. [2020]. To construct the vector embeddings, we represent each entity $e \in \mathcal{E}$ using $\mathbf{p}$ where each entry $\mathbf{p}_i$ is given by the number of times we have traveled along relation r_i across the n paths. We distinguish as to whether we have traveled along a relation in the forward or inverse direction, hence the resulting embeddings are $\mathbf{p} \in \mathbb{R}^{2N_r}$. We clustered the resulting representations using the Spectral Clustering algorithm [Ng et al., 2001] using the default parameters, and the number of clusters in $\{50, 100, 500, 1000\}$ for WN18RR and FB15K-237, and $\{30, 50, 100\}$ for UMLS. The hyperparameters for training the neural link predictors were selected with the baseline model on a held-out validation set.

ConFormAE. To train ConFormAE we experimented with initializing cluster memberships in two ways: randomly and using memberships learned via clustering of simple propositionalized embeddings, such as those used for ConFormA. We found that in most cases a random initialization performed competitively and reduced compute time hence this is the strategy we have opted to use throughout this work. To obtain the ConFormAE results quoted in Table 1 the initial number of clusters was in $\{50, 100, 500, 1000\}$. Our experiments

Table 3: Fragments of prototypical concepts learned for FB15K-237 by ConFormA and for WN18RR by ConFormAE, using ComplEx with $k = 2000$.

FB15K-237: ConFormA			WN18RR: ConFormAE		
Concept 1	**Concept 2**	**Concept 3**	**Concept 1**	**Concept 2**	**Concept 3**
political satire	hypothyroidism	Royal College of Music	mathematics	bird family	russia
absurdism	Crohn's disease	Royal Academy of Music	physics	arthropod genus	norway
experimental film	yellow fever	Moscow Conservatory	psychology	asterid dicot genus	israel
Surrealism	angina pectoris	Manhattan School of Music	computer science	arthropod family	mexico
independent film	pancreatitis	Milan Conservatory	chemistry	fish family	antarctica

showed that setting $n = 1$ i.e., training the neural predictor for one epoch after every E-step was sufficient for fast convergence (see Appendix Appendix F). Again, we use the same hyperparameters used for the baselines.

Experimental Results. In order to answer **Q1** we inspect the entities which form the concepts learned by ConFormA and ConFormAE on UMLS, FB15K-237 and WN18RR. Excerpts of prototypical clusters learned by ConFormA/E are shown in Table 3, while full clustering of UMLS and further examples can be found in Appendix C. Across all datasets entities appear to be meaningfully clustered into e.g., diseases, music schools and geographical locations. Hence, we can answer **Q1** affirmatively and note one advantage of ConFormAE over ConFormA– ConFormAE can yield meaningful concepts from random cluster initialization, without requiring any a priori knowledge on the structure of the KG. Ideally, we would like to also assess cluster quality in a quantitative manner, by e.g., computing point-wise mutual information between a clustering achieved by ConFormA or ConFormAE with a ground truth clustering of entities. Unfortunately, to the best of our knowledge there exists no large KG for which such information is available.

To answer **Q2** Table 1 reports the MRR and Hits@K for ConFormA and ConFormAE, for different values of embedding size k, after a grid search on regularizers, batch-size and learning rates for the baselines. We report additional results in Appendix D where we quantitatively inspect which triples benefit most from concept learning and report results with TuckER [Balazevic et al., 2019] – an additional neural link predictor. In general, we see a consistent boost over both DistMult and ComplEx baselines. The boost is especially striking on WN18RR. For example, a smaller ($k = 500$) model learned by ConFormA or ConFormAE is equally good or better than a much larger one ($k = 2000$) learned by ComplEx in terms of in terms of H@10. If we perform additional augmentations, such as adding reciprocal relationships [Lacroix et al., 2018] we find that the improvements from the two methods are additive, as reported in Appendix B. For FB15K-237 we also see an improvement in Table 1, though of a smaller magnitude. This can be explained by the fact that WN18RR is a much sparser KG and as such it can benefit more from our concept learning scheme. Therefore, we hypothesize that the concept reification and explicit augmentation might be especially beneficial for performing link prediction in a small data regime and thus alleviate the *cold-start problem* [Bobadilla et al., 2012]. To verify this (**Q3**), we performed a series of *sparsification* experiments, decreasing the percentage of training triples available to the propositionalisation algorithm and the link predictor. Table 2 shows the relative improvement upon the baseline in terms of MRR for the percentage of retained triples in

$\{5, 7, 10, 20, \ldots, 90\}$. Across all configurations we see a clear boost, more evident for few training triples. For WN18RR, ConFormA and ConFormAE improve over 80% w.r.t. their baselines when only 5% of training data is available. For FB15K-237, we also see a consistent boost, though there is some stochasticity for less than 20% training triples - this is likely due to its large number of predicates which increase the minimum number of training triples required to learn a good model.

To answer **Q4**, we inspect how generalization affects different triples after binning them w.r.t. the frequency (rare, medium, common) of their relations. The bins used to categorize relations into their frequency-based sub-populations (Table 19) were constructed by considering the total number of training examples. Fig. 3 reports the relative improvement of ConFormA and ConFormAE over the baselines on predicate sub-populations in terms of MRR w.r.t. the baseline for the aforementioned bins. Across all datasets, the largest improvement is observed on the rare predicates sub-population, with relative improvements of up to 8% on WN18RR, 15% on UMLS and 2% on FB15K-237 confirming that discovering concepts and augmenting KGs with them helps neural link predictors on triples whose predicates are underrepresented.

We note that while generally ConFormA and ConFormAE perform similarly, occasionally randomly-initialized ConFormAE performs worse, as can be seen for FB15K-237 in Fig. 3. As is the case for other Hard-EM algorithms, the quality of the clustering learned by ConFormAE depends to a certain extent on the cluster initialization. We find that the cluster memberships learned by ConFormA through clustering of propositionalized representations can provide a good initialization for ConFormAE and accelerate convergence. Strictly speaking, ConFormA can be seen as a special case of ConFormAE with a non-random initialization, where only a series of M-steps is performed.

Lastly, we consider the run-times. Averaging across all embedding sizes, datasets, and six random trials we find that ConFormA and ConFormAE only add a minimal overhead over the respective baselines: they are only 1.09 and 1.14 times slower. For example, training DistMult on WN18RR using $k = 2000$ for 100 epochs took 176min for the baseline, 211min for ConFormA and 227min for ConFormAE with random initialization.

7. Conclusions

In this work we have introduced the task of *unsupervised concept formation in KGs* and proposed two algorithms to achieve it – ConFormA and ConFormAE– based on entity clustering and KG augmentation. Our experiments show that our approaches can learn semantically-meaningful concepts and improve the accuracy on downstream link prediction tasks. We find that leveraging latent concept information helps neural link predictors to generalize to rare predicates and is especially beneficial in sparse KGs, where entities participate in few training triples. Moreover, learning new concepts as entities can help to automate the construction of KGs and the learned concept representations can be used for a variety of downstream tasks. While the assumption that every entity participates in exactly one concept can be unrealistic, it points to exciting future work on learning concept hierarchies. Lastly, our work also paves the way for principled probabilistic approaches to elicit discrete latent variables in neural link prediction models.

References

Ivana Balazevic, Carl Allen, and Timothy M. Hospedales. Tucker: Tensor factorization for knowledge graph completion. In *EMNLP/IJCNLP (1)*, pages 5184–5193. Association for Computational Linguistics, 2019.

Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008, Oct 2008.

Jesús Bobadilla, Fernando Ortega, Antonio Hernando, and Jesus Bernal. A collaborative filtering approach to mitigate the new user cold start problem. *Knowl. Based Syst.*, 26: 225–238, 2012.

Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *NIPS*, pages 2787–2795, 2013.

Rajarshi Das, Ameya Godbole, Shehzaad Dhuliawala, Manzil Zaheer, and Andrew McCallum. Non-parametric reasoning in knowledge bases. In *Automated Knowledge Base Construction*, 2020.

Gerben KD de Vries. A fast approximation of the weisfeiler-lehman graph kernel for rdf data. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 606–621. Springer, 2013.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22, 1977.

Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. Convolutional 2d knowledge graph embeddings. In *AAAI*, pages 1811–1818. AAAI Press, 2018.

John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12:2121–2159, 07 2011.

Richard Evans and Edward Grefenstette. Learning explanatory rules from noisy data. *J. Artif. Intell. Res.*, 61:1–64, 2018.

Demis Hassabis, Dharshan Kumaran, Christopher Summerfield, and Matthew Botvinick. Neuroscience-inspired artificial intelligence. *Neuron*, 95(2):245 – 258, 2017.

Seyed Mehran Kazemi and David Poole. Simple embedding for link prediction in knowledge graphs. In *NeurIPS*, pages 4289–4300, 2018.

Charles Kemp, Joshua B. Tenenbaum, Thomas L. Griffiths, Takeshi Yamada, and Naonori Ueda. Learning systems of concepts with an infinite relational model. In *AAAI*, pages 381–388. AAAI Press, 2006.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR (Poster)*, 2015.

Stanley Kok and Pedro M. Domingos. Statistical predicate invention. In *ICML*, volume 227 of *ACM International Conference Proceeding Series*, pages 433–440. ACM, 2007.

Stefan Kramer, Nada Lavrač, and Peter Flach. *Propositionalization Approaches to Relational Data Mining*, pages 262–291. Springer Berlin Heidelberg, Berlin, Heidelberg, 2001.

Timothée Lacroix, Nicolas Usunier, and Guillaume Obozinski. Canonical tensor decomposition for knowledge base completion. In *ICML*, volume 80 of *Proceedings of Machine Learning Research*, pages 2869–2878. PMLR, 2018.

Brenden M. Lake, Ruslan Salakhutdinov, and Joshua B. Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015.

George Lakoff and Mark Johnson. The metaphorical structure of the human conceptual system. *Cognitive Science*, 4(2):195 – 208, 1980.

Alexa McCray. An upper-level ontology for the biomedical domain. *Comparative and functional genomics*, 4:80–4, 01 2003.

Pasquale Minervini, Thomas Demeester, Tim Rocktäschel, and Sebastian Riedel. Adversarial sets for regularising neural link predictors. In *UAI*. AUAI Press, 2017.

Christopher Morris, Kristian Kersting, and Petra Mutzel. Glocalized weisfeiler-lehman graph kernels: Global-local feature maps of graphs. In *ICDM*, pages 327–336. IEEE Computer Society, 2017.

Andrew Y. Ng, Michael I. Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. In *NIPS*, pages 849–856. MIT Press, 2001.

Maximilian Nickel and Volker Tresp. Learning taxonomies from multi-relational data via hierarchical link-based clustering. In *Learning Semantics Workshop*, 2011.

Maximilian Nickel, Kevin Murphy, Volker Tresp, and Evgeniy Gabrilovich. A review of relational machine learning for knowledge graphs. *Proc. IEEE*, 104(1):11–33, 2016.

Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: online learning of social representations. In *KDD*, pages 701–710. ACM, 2014.

Emmanuel M. Pothos and Nick Chater. A simplicity principle in unsupervised human categorization. *Cognitive Science*, 26(3):303 – 343, 2002.

Jay Pujara, Eriq Augustine, and Lise Getoor. Sparsity and noise: Where knowledge graph embeddings fall short. In *EMNLP*, pages 1751–1756. Association for Computational Linguistics, 2017.

Matthew Richardson and Pedro M. Domingos. Markov logic networks. *Mach. Learn.*, 62 (1-2):107–136, 2006.

E. Rosch, C. Mervis, Wayne D. Gray, D. M. Johnson, and Penny Boyes-Braem. Basic objects in natural categories. *Cognitive Psychology*, 8:382–439, 1976.

Daniel Ruffinelli, Samuel Broscheit, and Rainer Gemulla. You CAN teach an old dog new tricks! on training knowledge graph embeddings. In *ICLR*. OpenReview.net, 2020.

Rajhans Samdani, Ming-Wei Chang, and Dan Roth. Unified expectation maximization. In *HLT-NAACL*, pages 688–698. The Association for Computational Linguistics, 2012.

Josh Tenenbaum. Building machines that learn and think like people. In *AAMAS*, page 5. International Foundation for Autonomous Agents and Multiagent Systems Richland, SC, USA / ACM, 2018.

Kristina Toutanova and Danqi Chen. Observed versus latent features for knowledge base and text inference. In *Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality*, page 57–66, 2015.

Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. Complex embeddings for simple link prediction. In *ICML*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 2071–2080. JMLR.org, 2016.

Zhao Xu, Volker Tresp, Kai Yu, and Hans-Peter Kriegel. Infinite hidden relational models. In *UAI*. AUAI Press, 2006.

Bishan Yang, Scott Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. Embedding entities and relations for learning and inference in knowledge bases. In *Proceedings of the International Conference on Learning Representations (ICLR) 2015*, May 2015.

Zhitao Ying, Jiaxuan You, Christopher Morris, Xiang Ren, William L. Hamilton, and Jure Leskovec. Hierarchical graph representation learning with differentiable pooling. In *NeurIPS*, pages 4805–4815, 2018.

Zhao Zhang, Fuzhen Zhuang, Meng Qu, Fen Lin, and Qing He. Knowledge graph embedding with hierarchical relation structure. In *EMNLP*, pages 3198–3207. Association for Computational Linguistics, 2018.

Below we provide an extension of the results presented in the main paper. We begin by presenting an analysis of computational complexity in Appendix A. Next, in Appendix B we explore combining ConFormA and ConFormAE with another augmentation method – reciprocal relations [Lacroix et al., 2018] – and find that best results are achieved by either combining the two approaches, or by our approach alone. In Appendix C we provide extended examples of the concept clusters learned by ConFormA and ConFormAE, while in Appendix D we provide further link prediction results, including examples of triples on which ConFormA and ConFormAE achieve the greatest gains over the baselines, and sparsification results for a range of embedding sizes and link predictors. Lastly, in Appendix E we provide additional information about the datasets used and in Appendix F we provide details of the experimental set-up for replicating our results.

Appendix A. Computational Complexity

The time and space complexity of ConFormA depends on the propositionalization, clustering, and neural link prediction model being used. In our experiments, the random path propositionalization has time complexity $\mathcal{O}(kL)$, where L is the max length of a path and k is the number of paths, while its space complexity is $\Theta(kN_r)$.

Clustering. For a vanilla spectral clustering implementation the complexity would be dominated by the $\mathcal{O}(N_e^3)$ cost of computing the Singular Value Decomposition of the propositionalization embedding matrix $\mathbf{P}$. Alternatively, the K-means algorithm would require $\mathcal{O}(tkN_eN_c)$ time, where t is the number of iterations and k is the embedding size.

Neural Link Prediction. For the cost of training and evaluating neural link prediction models, we refer the reader to their respective papers [Bordes et al., 2013, Trouillon et al., 2016, Yang et al., 2015, Dettmers et al., 2018, Lacroix et al., 2018]. We refer to [Ruffinelli et al., 2020] for a comparison of different choices of the loss function on several downstream link prediction tasks. We point out that, in our case, the number of entities in $\mathcal{G}$ becomes $N_e + N_c$, as the new set of entities in the augmented KG $\mathcal{G}'$ would include N_c concept entities.

Expectation-Maximization. In ConFormAE, the complexity of the M-step is that of training the neural link predictor. In the E-step we need to evaluate the score and loop through all of the entities and all the concepts, which results in $\mathcal{O}\left(N_eN_c\right)$ steps. Note that, in practice, this step can be efficiently parallelized on GPU. In Section 6 we report average run-times, showing that the computational cost is only marginally higher than that of an out-of-the-box link predictor: in our experiments, ConFormA and ConFormAE are respectively only 1.09 and 1.14 times slower than the neural link predictor alone.

Appendix B. Reciprocal Relations with ConFormA/E

Reciprocal relations [Lacroix et al., 2018] is a popular method of augmenting KGs by introducing an inverse of every relation into the graph. In Table 4 we compare link prediction performance between training neural link predictors on standard KGs, KGs augmented with inverse relations, training with concept augmentations (ConFormA and ConFormAE) and lastly, the effect of combining inverse relations with ConFormA and ConFormAE. Across all datasets and models we find that best performance is achieved either by combining inverse relations with ConFormA and ConFormAE, or by our method alone.

Table 4: Comparison of link prediction results between neural link predictors and ConFormA and ConFormAE trained on standard KGs and on KGs augmented reciprocal relations. All results are averages of runs with 5 different random seeds for the rank of 1000.

FB15K-237			**WN18RR**		
Model	Relations	MRR	Model	Relations	MRR
ComplEx	Standard	36.11	ComplEx	Standard	48.58
ComplEx	Reciprocal	36.22	ComplEx	Reciprocal	48.62
ConFormA	Standard	36.20	ConFormA	Standard	49.13
ConFormA	**Reciprocal**	**36.37**	**ConFormA**	**Reciprocal**	**49.25**
ConFormAE	Standard	36.25	ConFormAE	Standard	48.99
ConFormAE	Reciprocal	36.28	ConFormAE	Reciprocal	48.96
DistMult	Standard	35.26	DistMult	Standard	44.53
DistMult	Reciprocal	35.28	DistMult	Reciprocal	44.20
ConFormA	Standard	34.95	**ConFormA**	**Standard**	**45.59**
ConFormA	Reciprocal	35.34	ConFormA	Reciprocal	45.45
ConFormAE	**Standard**	**35.35**	ConFormAE	Standard	45.15
ConFormAE	**Reciprocal**	**35.35**	ConFormAE	Reciprocal	45.12

Table 5: Fragments of prototypical concepts learned for FB15K-237 by ConFormAE, using ComplEx with $k = 2000$.

FB15K-237: ConFormAE			
Concept 1	**Concept 2**	**Concept 3**	**Concept 4**
Turkey	Ridley Scott	traditional pop music	Academy Award for Best Animated Feature
Lithuania	Jerry Bruckheimer	electro house	Golden Globe Award for Best Animated Feature Film
Kuwait	Sidney Lumet	electric guitar	Grammy Award for Best Music Film
Guatemala	Mike Leigh	post-rock	MTV Video Music Award for Best Pop Video
Sri Lanka	Peter Weir	street punk	Grammy Award

Appendix C. Example Concepts

In this section we provide further examples of concepts learned by ConFormA and ConFormAE. Tables 5 and 6 show fragments of prototypical concepts learned by ConFormA for WN18RR and by ConFormAE for FB15K-237, respectively, while Table 7 shows the full clustering of ConFormAE for UMLS.

Table 6: Fragments of prototypical concepts learned for WN18RR by ConFormA, using ComplEx with $k = 2000$.

WN18RR: ConFormA				
Concept 1	**Concept 2**	**Concept 3**	**Concept 4**	**Concept 5**
carboxyl	counterintelligence	country	spanish-american war	quality
aconite	cyber-terrorism	national capital	vietnam war	trait
wintergreen oil	terrorism	geographical area	operation desert storm	property
protropin	military	city	world war	shape
uranyl	bioterrorism	island	battle of britain	skill

Table 7: Concepts learned by ConFormAE for UMLS, initialized with 50 random clusters.

Concept 1 cell_or_molecular-_dysfunction, disease_or_syndrome, experimen-tal_model_of_disease, injury_or_poisoning, mental_or_behavioral-_dysfunction, neoplastic_process, pathologic_function. **Concept 5** acquired_abnormality, age_group, anatomical_abnormality, congenital_abnormality, family_group, group, pa-tient_or_disabled_group, population_group, professional_or_occupa-tional_group. **Concept 9** biologic_function, cell_function, genetic_function, mental_process, molecular_function, organ_or_tissue_function, natural_phenomenon-_or_process, organism_function, physiologic_function.	**Concept 2** alga, amphibian, animal, archaen, virus, bacterium, bird, fish, fungus, human, invertebrate, mammal, organism, plant, reptile, rickttei s_or_chlamydia, vertebrate. **Concept 6** clinical_attribute, organism_attribute. **Concept 10** amino_acid_sequence, body_location_or_region, body_system, carbohydrate_sequence, classification, clinical_drug, conceptual_entity, drug_delivery_device, entity, finding, functional_concept, geographic_area, group_attribute, idea_or_concept, intellectual_product, spatial_concept, laboratory_or_test_result, language, manufactored_object, medical_device, molecular_sequence, nucleotide_sequence, regulation_or_law, research_device, sign_or_symptom.	**Concept 3** environmental_effect_of-_humans, event, phenomenon_or_process, qualitative_concept, quantitative_concept, temporal_concept, human_caused_phenome-non_or_process. **Concept 7** activity, behavior, health_care_activity, daily_or_recreational-_activity, diagnostic_procedure, educational_activity, governamental_or-_regulatory_activity, healt_care_related-_organization, individual_behavior, laboratory_procedure, machine_activity, social_behavior, molecular_biology-_research_technique, occupational_activity, organization, professional_society, research_activity, self_help_or_relief-_organization, therapeutic_or-_preventive_procedure.	**Concept 4** physical_object. **Concept 8** aminoacid_peptide_or-_protein, antibiotic, biologi-cally_active_substance, biomedical_or-_dental_material, body_substance, carbohydrate, chemical, chemical_viewed-_functionally, chemical_view-_structurally, eicosanoid, element_ion_or_isotope, enzyme, food, hazardous_or-_poisonous_substance, hormone, immunologic_factor, indicator_reagent_or-_diagnostic_aid, inorganic_chemical, lipid, neuroactive_substance_or-_biogenic_amine, nucleic_acid_nucleoside-_or_nucleotide, organic_chemical, organophosphorus-_compound, pharmacologic_substance, receptor, steroid, substance, vitamin.

Appendix D. Link Prediction Results

In this section we provide additional link prediction performance results of our method. Tables 9 to 14 show top 10 test triples for which our method achieves greatest improvement

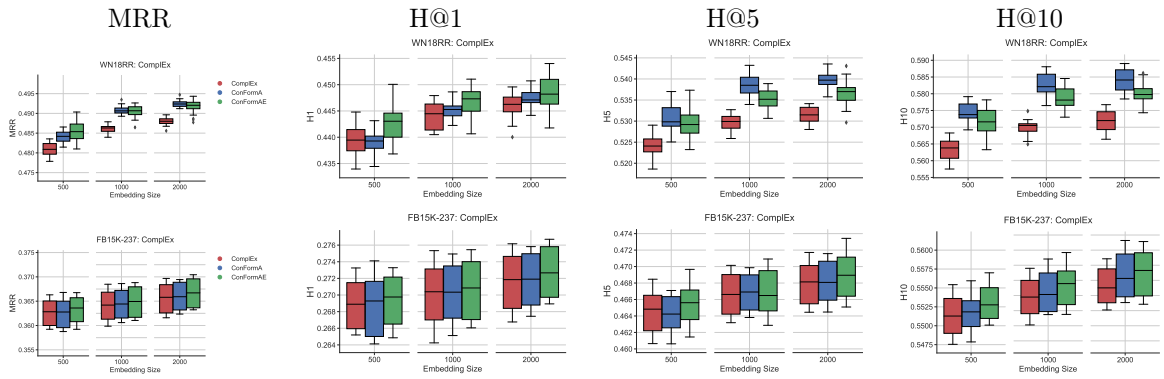

Table 8: MRR and Hits (H) at 1, 5, 10, 50 for ConFormA and ConFormAE when using ComplEx as baseline models on WN18RR and FB15K-237 KGs for different values of embedding size (k). Each configuration has been repeated with 30 different random seeds.

over the baseline neural link predictor. In Table 8 we visualize link prediction results shown earlier in tabular form (Table 1) and in Table 15 we provide link prediction performance for UMLS. Lastly, in Table 17 we provide an extension of results for link prediction on sparsified KGs, showing results for both, ComplEx and DistMult, for a range of embedding sizes.

Table 9: Top 10 UMLS test triples with greatest improvement of ConFormA over baseline ComplEx model, for $k = 50$.

UMLS Test Triple			Reciprocal Rank	
s	p	o	Baseline	ConFormA
mental_process	isa	organism_function	0.06	1.0
disease_or_syndrome	occurs_in	mental_or_behavioral_dysfunction	0.2	1.0
finding	isa	conceptual_entity	0.25	1.0
steroid	interacts_with	eicosanoid	0.33	1.0
tissue	adjacent_to	body_space_or_junction	0.33	1.0
embryonic_structure	location_of	virus	0.33	1.0
machine_activity	isa	activity	0.33	1.0
antibiotic	interacts_with	biologically_active_substance	0.5	1.0
congenital_abnormality	complicates	anatomical_abnormality	0.5	1.0
carbohydrate	affects	mental_process	0.5	1.0
laboratory_or_test_result	co-occurs_with	sign_or_symptom	0.5	1.0

D.1 Link Prediction with TuckER

To further demonstrate that our method can improve upon a wide range of neural link predictors and embedding sizes, we report link prediction results with TuckER [Balazevic et al., 2019] – a recent neural link predictor which has achieved competitive performance for small embedding sizes.

Table 10: Top 10 UMLS test triples with greatest improvement of ConFormAE over baseline ComplEx model, for $k = 50$.

UMLS Test Triple			Reciprocal Rank	
s	p	o	Baseline	ConFormAE
mental_process	isa	organism_function	0.06	1.0
environmental_effect_of_humans	isa	phenomenon_or_process	0.14	1.0
cell	part_of	body_part_organ_or_organ_component	0.2	1.0
neuroreactive_substance_or_biogenic_amine	isa	biologically_active_substance	0.2	1.0
human_caused_phenomenon_or_process	isa	event	0.25	1.0
fully_formed_anatomical_structure	location_of	virus	0.25	1.0
steroid	interacts_with	eicosanoid	0.33	1.0
cell_component	location_of	body_space_or_junction	0.33	1.0
cell_component	location_of	body_space_or_junction	0.33	1.0
organism_function	produces	hormone	0.33	1.0
therapeutic_or_preventive_procedure	complicates	pathologic_function	0.5	1.0

Table 11: Top 10 WN18RR test triples with greatest improvement of ConFormA over baseline ComplEx model, for $k = 500$.

WN18RR Test Triple			Reciprocal Rank	
s	p	o	Baseline	ConFormA
latin.n.03	_hypernym	person.n.01	0.0	1.0
periwinkle.n.02	_hypernym	herb.n.01	0.0	1.0
threepence.n.01	_hypernym	coin.n.01	0.0	1.0
stress.n.03	_hypernym	emphasis.n.01	0.01	1.0
trade_name.n.01	_member_of_domain_usage	clomiphene.n.01	0.01	1.0
red-winged_blackbird.n.01	_hypernym	new_world_blackbird.n.01	0.01	1.0
libel.n.01	_synset_domain_topic_of	law.n.01	0.09	1.0
merginae.n.01	_member_meronym	mergus.n.01	0.14	1.0
saxifraga.n.01	_member_meronym	saxifrage.n.01	0.14	1.0
regimentals.n.01	_hypernym	military_uniform.n.01	0.17	1.0
new_zealand.n.01	_member_of_domain_region	returning_officer.n.01	0.2	1.0

To compute the TuckER baselines for ranks k in $\{50, 100, 500\}$, following the training set-up described in [Balazevic et al., 2019], we used the Adam optimizer [Kingma and Ba, 2015] and performed a gridsearch over the learning rates in {0.1, 0.05, 0.01, 0.005, 0.001, 0.0005, 0.0001}. We set the predicate embedding size equal to the entity embedding size for both datasets. A batch-size of 128 and learning rate decay of 1.0 were held constant for all experiments.

Link prediction results with TuckER are shown in Appendix D.1, where we find that ConFormA and ConFormAE consistently outperform the baseline TuckER model for nearly all of the configurations.

Table 12: Top 10 WN18RR test triples with greatest improvement of ConFormAE over baseline ComplEx model, for $k = 500$.

WN18RR Test Triple			Reciprocal Rank	
s	p	o	Baseline	ConFormAE
periwinkle.n.02	_hypernym	herb.n.01	0.0	1.0
trade_name.n.01	_member_of_domain_usage	clomiphene.n.01	0.01	1.0
ranunculaceae.n.01	_member_meronym	isopyrum.n.01	0.04	1.0
shiite.n.01	_hypernym	muslim.n.01	0.05	1.0
sapindaceae.n.01	_member_meronym	genus_harpullia.n.01	0.08	1.0
libel.n.01	_synset_domain_topic_of	law.n.01	0.09	1.0
right_to_vote.n.01	_synset_domain_topic_of	law.n.01	0.12	1.0
united_states.n.01	_has_part	missouri.n.02	0.14	1.0
compositae.n.01	_member_meronym	balsamorhiza.n.01	0.17	1.0
cupressaceae.n.01	_member_meronym	taxodium.n.01	0.17	1.0
plural.n.01	_member_of_domain_usage	sunglasses.n.01	0.17	1.0

Table 13: Top 10 FB15K-237 test triples with greatest improvement of ConFormA over baseline ComplEx model, for $k = 500$.

FB15K-237 Test Triple			Reciprocal Rank	
s	p	o	Baseline	ConFormA
Ocean Software	/business/[...]/industry	video game	0.0	1.0
Alaska	/location/[...]/contains	Nome Census Area	0.01	1.0
Republican Party	/government/[...]/politician	Kevin Smith	0.03	1.0
Bancroft Prize	/award/[...]/category_of	Bancroft Prize	0.03	1.0
Kate Hudson	/people/[...]/type_of_union	domestic partnership	0.03	1.0
Jacqueline Bisset	/people/[...]/gender	female organism	0.05	1.0
Slumdog Millionaire	/film/[...]/film_release_region	United States of America	0.05	1.0
Phil LaMarr	/people/[...]/profession	actor	0.05	1.0
FilmFlex	/film/[...]/film	Night at the Museum	0.06	1.0
The Portrait of a Lady	/film/film/genre	film adaptation	0.07	1.0
Australia	/location/[...]/currency	Australian dollar	0.07	1.0

Appendix E. Datasets

For each of the datasets used to evaluate our approach – UMLS, WN18RR and FB15K-237 – we provide distribution statistics, including entity and relation densities, in Table 18. The frequency bins shown in Table 19 were used to divide relations into frequency sub-populations for computing the link prediction results in Fig. 3.

Table 14: Top 10 FB15K-237 test triples with greatest improvement of ConFormAE over baseline ComplEx model, for $k = 500$.

FB15K-237 Test Triple			Reciprocal Rank	
s	p	o	Baseline	ConFormAE
Ocean Software	/business/[...]/industry	video game	0.0	1.0
Republican Party	/government/[...]/politician	Kevin Smith	0.03	1.0
Bancroft Prize	/award/[...]/category_of	Bancroft Prize	0.03	1.0
The Untouchables	/film/[...]/genre	crime fiction	0.03	1.0
Slumdog Millionaire	/film/[...]e/film_release_region	United States of America	0.05	1.0
Seattle University	/education/[...]/student	Duff McKagan	0.06	1.0
Ryan Reynolds	/people/[...]/nationality	Canada	0.06	1.0
Satellite Awards 2008	/award/[...]/award_winner	Tom McCarthy	0.08	1.0
Omaha	/location/[...]/time_zones	Central Time Zone	0.09	1.0
Mr. Nobody	/film/[...]/film_release_region	Finland	0.09	1.0
The Best Exotic Marigold Hotel	/film/[...]/film_release_region	Finland	0.09	1.0

Table 15: Mean Reciprocal Rank (MRR) and Hits at K (H@K) for ConFormA and ConFormAE when using DistMult or ComplEx as baselines on UMLS for different values of embedding size (k). Each configuration was repeated with 30 random seeds, and we report the means of each metric.

			ComplEx				DistMult			
	k	Model	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10
UMLS	50	Baseline	94.64	90.92	**98.26**	99.59	75.50	66.62	81.01	91.70
		ConFormA	94.74	91.10	98.19	99.59	75.36	66.47	80.83	**91.72**
		ConFormAE	**95.12**	**91.92**	98.06	**99.63**	**75.76**	**67.14**	**81.13**	91.67
	100	Baseline	95.45	92.34	98.44	**99.66**	**76.13**	**68.28**	80.65	91.42
		ConFormA	**95.68**	**92.77**	98.46	99.64	76.11	68.07	80.56	91.59
		ConFormAE	95.57	92.56	**98.48**	99.63	76.05	67.85	**80.98**	**91.66**
	200	Baseline	96.46	94.08	**98.84**	99.72	76.21	68.37	80.69	91.36
		ConFormA	96.29	93.77	98.68	99.70	**76.43**	**68.92**	80.52	91.43
		ConFormAE	**96.47**	**94.09**	98.67	**99.75**	76.16	68.25	**80.85**	**91.59**

Appendix F. Training Details

F.1 Baselines

To obtain the best parameters for the baselines, we ran the same grid search for all of the datasets on both, ComplEx and DistMult, with the ranks set to $\{50, 100, 200\}$ for UMLS and $\{500, 1000, 2000\}$ for WN18RR and FB15K-237, using the standard train/validation/test splits. The grid consisted of three batch-sizes in $\{50, 100, 500\}$, three learning rates: $\{10^{-1}, 10^{-2}, 10^{-3}\}$ and six regularization strengths in $\{10^{-3}, 5 \times 10^{-3}, \ldots, 10^{-1}, 5 \times 10^{-1}\}$.

F.2 ConFormA

Propositionalisation To generate the representations we explored the parameter range suggested by Perozzi et al. [2014], using a *minimum path length* of 2, *maximum path length* in

Table 16: Link prediction performance of TuckER with ConFormA and ConFormAE evaluated on WN18RR and FB15K-237, for a range of embedding sizes, k in $\{50, 100, 500\}$. Each experiment was repeated with 6 different random seeds.

Dataset	k	Model	MRR	H@1	H@3	H@5	H@10	H@50
FB15K-237	50	TuckER	28.96	20.59	31.64	37.62	45.86	64.43
		ConFormA	**29.06**	**20.71**	**31.73**	**37.65**	**45.88**	**64.52**
		ConFormAE	29.01	20.66	**31.73**	37.55	45.77	64.33
	100	TuckER	30.14	**21.74**	32.87	**38.89**	**47.23**	65.52
		ConFormA	**30.15**	21.62	32.76	38.78	47.01	65.22
		ConFormAE	30.05	21.57	**32.91**	38.86	47.21	**65.58**
	500	TuckER	32.85	24.11	36.08	42.17	50.40	67.74
		ConFormA	32.98	24.24	36.19	42.35	50.46	**68.09**
		ConFormAE	**33.05**	**24.28**	**36.34**	**42.42**	**50.54**	67.93
WN18RR	50	TuckER	43.59	**40.86**	44.73	46.22	48.54	54.08
		ConFormA	**43.65**	40.81	**44.74**	**46.56**	**48.93**	**54.83**
		ConFormAE	43.18	40.41	44.46	46.00	48.10	53.73
	100	TuckER	45.10	42.27	46.31	48.10	50.33	55.32
		ConFormA	**45.44**	**42.39**	**46.80**	**48.69**	**51.07**	**56.68**
		ConFormAE	44.84	42.02	46.08	47.72	50.04	55.20
	500	TuckER	46.01	42.29	48.00	50.05	52.53	58.35
		ConFormA	**46.59**	**42.72**	**48.64**	**50.71**	**53.51**	**59.63**
		ConFormAE	46.08	42.37	48.09	50.02	52.60	58.28

$\{3, 5, 10, 20, 30\}$ and two *number of paths* parameters: 32 and 64. We found that the maximum path length parameter was most influential in determining how well a representation captured the characteristics of a given KG. For both, WN18RR and FB15K-237, we found that setting maximum path length to 5 and number of paths to 64 gave competitive results.

Clustering Algorithm We experimented with clustering the propositionalised representations using a number of clustering algorithms: K-Means, Spectral Clustering, Affinity Propagation and DBSCAN. Across these, we have found no significant difference in performance in terms of both, cluster quality and downstream link prediction performance, hence we used Spectral Clustering for all experiments.

Number of clusters The number of clusters was treated as a hyperparameter and chosen from $\{50, 100, 500, 1000\}$ for FB15K-237 and WN188R, and from $\{30, 50, 100\}$ for UMLS. As shown in Fig. 4, lower numbers of clusters on average resulted in better link prediction performance.

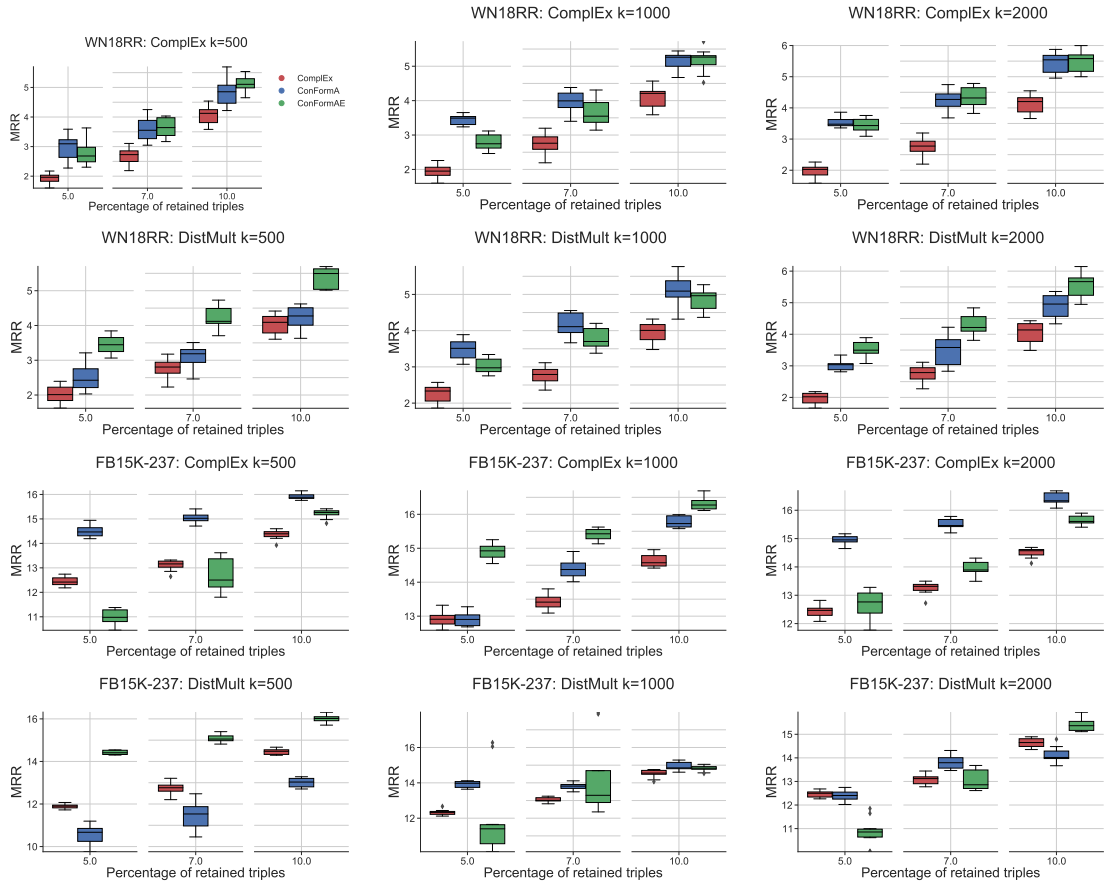

Table 17: MRR of ConFormA, ConFormAE, and baseline models – either DistMult or ComplEx – on sparsified WN18RR and FB15K-237, where the percentage of retained training triples is in $\{5, 7, 10\}$, for different values of embedding size (k). Each experiment was repeated with 6 different random seeds.

F.3 ConFormAE

Number of epochs, n, per E-Step: To explore the effect of changing the number of epochs, n, per E-step, we trained a range models and embedding sizes, varying n in $\{1, 2, 3, 5\}$. We found that $n = 1$ provided a good compromise in terms of link prediction performance across different embedding sizes, datasets and neural link predictors - a visualization for WN18RR can be found in Figure Fig. 5. Our heuristic experiments also suggested that setting $n = 1$ yielded best cluster quality.

Cluster initialization: Between the two initialization methods we experimented with – random initialization and Spectral Clustering of random paths propositionalisation – we found no significant difference in performance, hence we opted for random initialization due to its reduced complexity.

Table 18: Statistics of knowledge graphs. Train, val and test denote the number of facts in the training, validation and test sets, respectively. $|\mathcal{E}|$ and $|\mathcal{R}|$ represent the number of unique entities and relations in the KG. RD and ED are measures of relation and entity density, respectively.

	Train	Val	Test	$\lvert\mathcal{E}\rvert$	$\lvert\mathcal{R}\rvert$	RD	ED
UMLS	5.2k	652	661	135	46	113	77
WN18RR	86.8k	3k	3.1k	40.5k	11	7891	4
FB15K-237	272.1k	17.5k	20.4k	14.5k	237	1148	38

Table 19: Bins for categorizing relations into sub-populations based on their frequency, N, in the training set.

Sub-population	WN18RR	FB15K-237	UMLS
Rare	$N < 10^3$	$N < 10^2$	$N < 20$
Medium	$10^3 < N \leq 10^4$	$10^2 < N \leq 10^3$	$20 < N \leq 150$
Common	$N > 10^4$	$N > 10^3$	$N > 150$

Number of clusters: We used the same ranges to select the number of clusters as for ConFormA. It is worth noting, however, that while for ConFormA the number of clusters is fixed, in ConFormAE that number of clusters can decrease during training if no entities are assigned to some concepts. The number of clusters specified in Fig. 4 for ConFormAE corresponds to the initial number of clusters.

F.4 Evaluation

During evaluation, we only consider the triples and entities appearing in the original dataset, to make sure the evaluation metrics for ConFormA and ConFormAE are computed using exactly the same protocol as for the baselines.

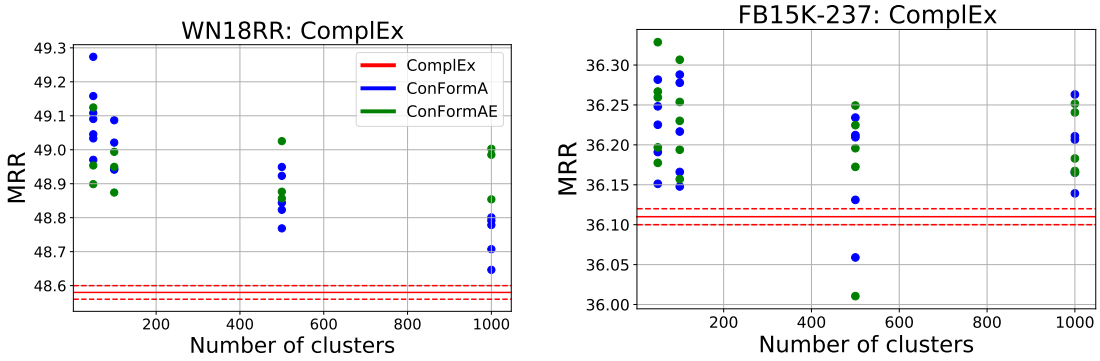

Figure 4: The effect of varying the number of concepts on Mean Reciprocal Rank (MRR), shown for WN18RR and FB15K-237 with ComplEx, using rank size of 1000.

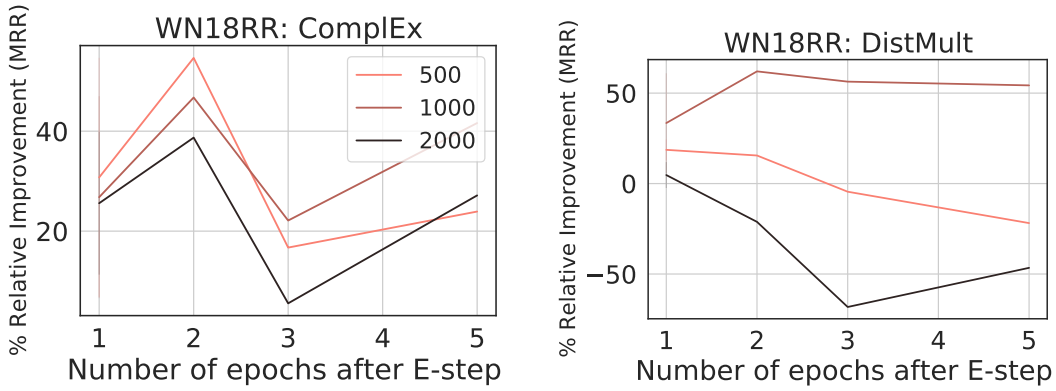

Figure 5: The relative improvement on Mean Reciprocal Rank (MRR) with varying the number of neural link predictor training epochs between every E-step in ConFormAE, shown for WN18RR with ComplEx and DistMult, with rank size (k) in $\{500, 1000, 2000\}$. Each experiment was repeated with 5 different random seeds and averages plotted.