Neural Concept Formation in Knowledge Graphs

Agnieszka Dobrowolska[1] **Antonio Vergari**[2] **Pasquale Minervini**[1]
[1]University College London [2] University of California, Los Angeles
aga.dobrowolska.16@ucl.ac.uk aver@cs.ucla.edu p.minervini@ucl.ac.uk

Abstract

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 effectively used to augment a KG. In our experiments we show 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 while delivering meaningful concept representations.

1 Introduction

One of the most remarkable aspects of human intelligence is arguably the capacity to abstract and summarize knowledge into *concepts*, which is believed to play a central role in the ability to quickly learn from few examples [21] and to robustly generalize to unseen data [30, 23, 32]. It is no wonder that many machine learning and knowledge representation methods have tried to "reverse-engineer" how humans learn concepts [22, 14] in order to automate reasoning as well as Knowledge Base (KB) construction [18, 17, 38].

Among the most prominent knowledge representation formalisms, there are *Knowledge Graphs* (KGs) – graph-structured KBs where knowledge about the world is encoded in the form of relationships between entities. Reasoning and learning routines for KGs build upon the *link prediction* task, which consists in identifying missing links between entities in the KG. Current state-of-the-art link prediction models are *neural link predictors* – also referred to as Knowledge Graph Embedding (KGE) models – that learn an embedding representation for each entity in the KG via back-propagation [28]. However, neural link predictors were shown not to be very accurate in the presence of sparse observations [31], and may not be able to learn patterns involving *sets of entities* [12].

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, while providing meaningful representations for downstream tasks. Specifically, we make the following contributions. Firstly, we formalize concept learning as an unsupervised clustering step over entities in a KG, noting that by *reifying* concept membership relationships into KG facts and by incorporating them in the KG, we can produce more accurate neural link prediction models (see Section 2). Secondly, we introduce a single, principled probabilistic framework for to jointly learning concept memberships and neural link prediction models at once, by maximizing the likelihood of the KG triples via an Expectation-Maximization scheme (see Section 3). Lastly, we execute a rigorous empirical evaluation on several real-world KG benchmarks, including a new dataset in the biomedical domain, showing that performing concept learning on KGs can be an effective way to improve the statistical accuracy of neural link predictors.

4th Knowledge Representation and Reasoning Meets Machine Learning Workshop (KR2ML 2020), at NeurIPS.

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

1: **Input:** A KG $\mathcal{G}$, number of clusters N_c and number of epochs n
2: **Output:** parameters $\mathbf{\Theta}$ and cluster assignments $\mathcal{S} = \{S_1, \ldots, S_{N_c}\}$
3: $\mathbf{P} \leftarrow \mathsf{propositionalization}(\mathcal{G})$ ▷ E.g., by random path generation (cf. Appendix A)
4: $\mathcal{S} \leftarrow \mathsf{Clustering}(\mathbf{P}, N_c)$ ▷ E.g., spectral clustering
5: $\mathcal{G}' \leftarrow \mathcal{G} \cup \{\langle e_i, \text{ISA}, c_j\rangle \mid e_i \in S_j, S_j \in \mathcal{S}\}$ ▷ KG augmentation
6: $\mathbf{\Theta} \leftarrow \mathsf{init}()$
7: **for** n epochs **do**
8: $\quad \mathbf{\Theta} \leftarrow \mathsf{train}(\mathcal{G}', \mathbf{\Theta})$ ▷ Update the parameters $\mathbf{\Theta}$ of a KGE model on $\mathcal{G}'$
return $\mathbf{\Theta}, \mathcal{S}$

2 CONFORMA: Learning Concepts by Augmenting Knowledge Graphs

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 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 and the object of the triple, returns the likelihood that s and o are related by the relation type r: $\phi_r(\mathbf{e}_s, \mathbf{e}_o) \in \mathbb{R}$. This scoring function implicitly defines a probability distribution over triples, i.e., $\log p(\langle s, r, o\rangle) \propto \phi_r(\mathbf{e}_s, \mathbf{e}_o)$. Model parameters can be learned from the data by maximizing the likelihood of triples in the $\mathcal{G}$.

Given a KG $\mathcal{G}$, *concept learning* equals to identify sets of entities $S_1, \ldots, S_{N_c} \subseteq \mathcal{E}$ that are semantically related and can be *abstracted* into concepts $c_1, \ldots, c_{N_c}$, here representing some novel entities. We want to find a *partitioning* $\mathcal{S} = \{S_i\}_{i=1}^{N_c}$ of $\mathcal{E}$, i.e., $\bigcup_{S \in \mathcal{S}} S = \mathcal{E}$ and $\forall S_i, S_j \in \mathcal{S} \rightarrow S_i \cap S_j = \emptyset$. To this end, we first *cluster* the entities in $\mathcal{G}$, and then reify cluster membership relations, i.e., materialize them as triples to *augment* the KG $\mathcal{G}$. Algorithm 1 summarizes our framework which can be instantiated for different clustering and neural link prediction models. We name it CONFORMA, Concept Formation via Augmentation. Next, we discuss *how* to perform these two phases, and *why* neural link prediction models can benefit from being trained on augmented KGs.

Clustering entities. Ideally, clustering in CONFORMA could by performed by any relational clustering algorithm. However, classical probabilistic approaches such as statistical predicate invention [18] and stochastic block models [17, 38] would hardly scale to modern KGs with hundreds of thousands of entities. This poses a challenge also to kernel-based [1, 6, 26] approaches. To overcome this issue, we opt for a more computationally efficient alternative: we first *propositionalise* entities into d-dimensional embedding vectors [19] and then employ a propositional clustering algorithm – e.g., spectral clustering [27] – over this now tabular representation $\mathbf{P} \in \mathbb{R}^{N_e \times d}$. We find that generating sparse binary representations by describing entities in terms of the (co-)domains of the relations they participate in, or by executing multi-hop random paths in $\mathcal{G}$, as proposed by [4], provides scalable and accurate representations. [1] Details about the two strategies can be found in Appendix A.

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_i participating in a cluster $S_j \in \mathcal{S}$ we create a new triple of the form $\langle e_i, \text{ISA}, c_j\rangle$, where c_j is a new entity denoting the j-th concept that will be shared among all entities in S_j, and ISA is a freshly introduced relation denoting concept memberships. Let $\mathcal{G}'$ denote the KG augmented in this way, learning a neural link prediction model over it (lines 6-8 of Algorithm 1) would require updating an additional set of parameters $\mathbf{\Theta}$, comprising now the concept embeddings $\mathbf{C} \in \mathbb{C}^{N_c \times k}$ for the newly introduced concept entities $\mathcal{C} = \{c_1, \ldots, c_{N_c}\}$. Moreover, it will yield the following advantages. 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 [20, 16] or hierarchical relation information [41] are explicitly added to KGs. Secondly, they help to make very sparse KGs more dense, tackling the sparsity issues in neural link predictors [31]. Lastly, learning new concepts as entities helps to automate construction of KGs, and their learned embeddings can be exploited in additional downstream tasks.

[1] We explored clustering directly over the embeddings learned by a neural link prediction model like ComplEx or DistMult, but with scarce success. We describe and discuss this experiment in Appendix D.

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

Input: number of clusters N_c, initial parameters $\mathbf{\Theta}$, cluster assignments $\mathcal{S}$, and number of epochs n and t.
Output: updated parameters $\mathbf{\Theta}$ and cluster assignments $\mathcal{S}$
for t iterations **do**
  $\mathcal{S} \leftarrow$ emptyAssignments()
  for $e \in \mathcal{E}$ **do** ▷ E-step: Make hard assignments
    $\hat{c} \leftarrow \arg\max_{c \in \mathcal{C}} \phi_{\text{ISA}}(\mathbf{e}_e, \mathbf{e}_c)$
    $S_{\hat{c}} \leftarrow S_{\hat{c}} \cup \{e_i\}$
  $\mathcal{G}' \leftarrow \mathcal{G} \cup \{\langle e_i, \text{ISA}, c_j \rangle \mid e_i \in S_j, S_j \in \mathcal{S}\}$ ▷ Augment the KG
  for n epochs **do**
    $\mathbf{\Theta} \leftarrow \text{train}(\mathcal{G}', \mathbf{\Theta})$ ▷ M-step: refine the neural link prediction model
return $\mathbf{\Theta}, \mathcal{S}$

3 CONFORMAE: Joint learning of Concepts and Embeddings via EM

CONFORMA is quite flexible: 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 learn both the concepts and the embeddings jointly, in a single loop? Ideally, we could cast this as a joint optimization problem to maximise 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 $z \in \mathcal{C}$. As directly maximising this marginal likelihood is intractable, we adopt an iterative expectation-maximisation (EM)-like scheme [7]. 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 E and M steps efficiently[2].

E step. Exactly computing all the cluster memberships $p(Z = c_j | e_i)$ for entity e_i and concept $c_j \in \mathcal{C}$ 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* [34, 18], i.e., $\hat{c} = \arg\max_{c \in \mathcal{C}} \log p(Z = c \mid e_i)$ for each entity e_i. Note that this can be done exactly and efficiently as $\hat{c} = \arg\max_{c \in \mathcal{C}} \phi_{\text{ISA}}(\mathbf{e}_e, \mathbf{e}_c)$ since in our augmentation scheme it holds that $\log p(Z = c|e) \propto \phi_{\text{ISA}}(\mathbf{e}_e, \mathbf{e}_c)$.

M step. The aim of the step is to find the best set of parameters $\mathbf{\Theta}$ for a neural link prediction model by maximizing the expected log-likelihood $\mathbb{E}_{z \sim p(Z=k|\mathcal{E})} [\log p(X, Z)]$, where the observed variables X denote the original triples in $\mathcal{G}$. Again, computing this exactly is hard in our scenario. Nevertheless, it 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). 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 $\mathcal{G}'$.[3]

4 Experiments

We aim to answer the following research questions: **Q1**) can unsupervised concept learning boost neural link prediction performance?, **Q2**) are the learned concepts semantically-meaningful? and **Q3**) how does the augmentation impact generalization over rare entities and relation types? To this end, we consider two neural link prediction models as baselines: ComplEx [37] and DistMult [39], and experiment on four datasets: WN18RR [9] and FB15K-237 [36] – two large benchmark KGs and UMLS [25] and Hetionet [15] – a small and a large biomedical KG. We report here results for the first two and refer the reader to Appendix C for complete results and implementation details.

Tables 1 and 3 report the mean reciprocal ranks and hits 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 and varying the number of clusters (see Appendix C). In all settings, both algorithms generally improve over the DistMult and ComplEx baselines. The boost is striking on WN18RR:

[2] We found that providing a random initialization to cluster memberships works best in practice. Convergence can be established by monitoring the KGE loss on a held-out set or the number of iterations t can be fixed.

[3] We found in our experiments that setting $n = 1$ is sufficient for fast convergence.

Table 1: Mean reciprocal rank (MRR) and Hits (H) at 1,3,10 for ConFormA and ConFormAE when using DistMult or ComplEx as baseline KGE models on WN18RR and FB15K-237 KGs for different values of embedding sizes (k). Best values for each metric and k in bold.

			DistMult				ComplEx			
	k	Model	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10
WN18RR	500	Baseline	44.40	40.30	45.63	52.81	47.85	43.41	49.44	56.57
		ConFormA	44.19	39.96	45.33	52.94	48.55	43.94	50.24	**57.67**
		ConFormAE	**44.89**	**40.84**	**46.00**	**53.38**	**48.77**	**44.42**	**50.48**	57.29
	1000	Baseline	44.80	41.02	45.80	52.52	48.34	43.81	50.18	56.99
		ConFormA	**45.47**	**41.16**	**46.76**	**54.28**	49.12	44.42	**50.83**	**58.70**
		ConFormAE	45.20	40.97	46.41	54.05	**49.25**	**44.81**	50.81	58.33
	2000	Baseline	45.20	41.05	**46.39**	53.75	48.62	44.07	50.34	57.28
		ConFormA	44.93	40.60	45.98	53.67	**49.40**	44.80	**50.86**	**59.00**
		ConFormAE	**45.38**	**41.16**	**46.39**	**54.04**	49.42	**45.20**	50.41	58.42
FB15K237	500	Baseline	34.88	25.56	38.34	53.52	35.89	26.47	39.31	54.82
		ConFormA	34.92	25.65	38.39	53.54	36.08	**26.77**	39.39	55.00
		ConFormAE	**35.01**	**25.72**	**38.40**	**53.65**	**36.13**	26.76	**39.46**	**55.09**
	1000	Baseline	35.26	25.83	**38.82**	54.26	36.18	26.69	**39.81**	55.21
		ConFormA	35.30	25.91	38.78	**54.28**	36.26	**26.88**	39.74	55.22
		ConFormAE	**35.40**	**26.11**	38.77	**54.28**	**36.27**	26.84	**39.81**	**55.34**
	2000	Baseline	35.47	26.13	38.76	54.42	**36.37**	**27.01**	**39.89**	**55.45**
		ConFormA	35.55	26.18	**39.03**	54.32	**36.37**	26.99	**39.89**	55.19
		ConFormAE	**35.62**	**26.31**	39.02	**54.43**	36.34	26.96	39.81	55.36

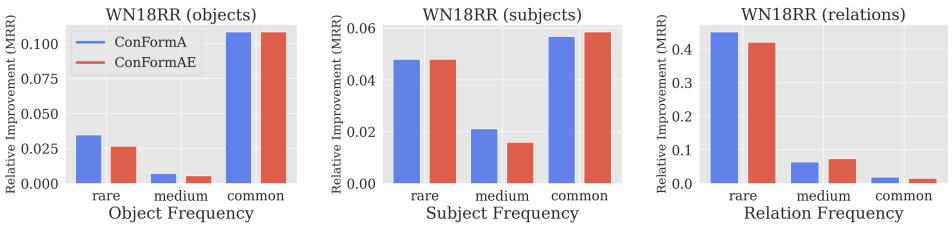

Figure 1: Relative improvement in terms of MRR of ConFormA and ConFormAE over the ComplEx baseline for WN18RR with $k = 100$.

a smaller ($k = 500$) model learned by ConFormAE is equally good or better than a much larger one ($k = 2000$) learned by ComplEx. We can then answer question **Q1** affirmatively. We answer **Q2** affirmatively by inspecting the entities which form the concepts learned by ConFormAE on UMLS, which is small enough to allow for easy qualitative analysis. As shown in Appendix F, entities appear to be meaningfully clustered into e.g., biological, chemical and anatomical groups. Lastly, to answer **Q3** we inspect how generalization affects different triples after binning them w.r.t. the frequency (rare, medium, common) of their subjects, objects or relations. Figures 1 and 3 report the relative improvement of ConFormA and ConFormAE in terms of MRR w.r.t. the baseline for the aforementioned bins. We can see a clear boost for both algorithms on WN18RR for the rare relation types sub-population, and for ConFormAE alone on FB15K-237. This indicates that discovering concepts and augmenting KGs with them helps neural link predictors deal with sparse KGs.

5 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 – which perform entity clustering and KG augmentation to improve neural link predictor performances. They also pave the way for principled probabilistic ways to elicit discrete latent variables in neural link prediction models – an interesting research venue which we are currently exploring.

References

[1] 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.

[2] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *Advances in neural information processing systems*, pages 2787–2795, 2013.

[3] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Durán, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, page 2787–2795, Red Hook, NY, USA, 2013. Curran Associates Inc.

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

[5] Gerben K. D. de Vries. A fast approximation of the weisfeiler-lehman graph kernel for rdf data. In Hendrik Blockeel, Kristian Kersting, Siegfried Nijssen, and Filip Železný, editors, *Machine Learning and Knowledge Discovery in Databases*, pages 606–621, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[6] 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.

[7] 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.

[8] Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. Convolutional 2d knowledge graph embeddings. *arXiv preprint arXiv:1707.01476*, 2017.

[9] Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. Convolutional 2d knowledge graph embeddings. In Sheila A. McIlraith and Kilian Q. Weinberger, editors, *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 1811–1818. AAAI Press, 2018.

[10] 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.

[11] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD'96, page 226–231. AAAI Press, 1996.

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

[13] Brendan J. Frey and Delbert Dueck. Clustering by passing messages between data points. *Science*, 315(5814):972–976, 2007.

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

[15] Daniel Himmelstein, Antoine Lizee, Christine Hessler, Leo Brueggeman, Sabrina Chen, Dexter Hadley, Ari Green, Pouya Khankhanian, and Sergio Baranzini. Systematic integration of biomedical knowledge prioritizes drugs for repurposing. *eLife*, 6:e26726, 09 2017.

[16] Seyed Mehran Kazemi and David Poole. Simple embedding for link prediction in knowledge graphs, 2018.

[17] Charles Kemp, Joshua Tenenbaum, Thomas Griffiths, Takeshi Yamada, and Naonori Ueda. Learning systems of concepts with an infinite relational model. *Cognitive Science*, 21, 01 2006.

[18] Stanley Kok and Pedro Domingos. Statistical predicate invention. In *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, page 433–440, New York, NY, USA, 2007. Association for Computing Machinery.

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

[20] Timothée Lacroix, Nicolas Usunier, and Guillaume Obozinski. Canonical tensor decomposition for knowledge base completion, 2018.

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

[22] Brenden M. Lake, Tomer D. Ullman, Joshua B. Tenenbaum, and Samuel J. Gershman. Building machines that learn and think like people. *Behavioral and Brain Sciences*, 40:e253, 2017.

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

[24] J. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*, pages 281–297, Berkeley, Calif., 1967. University of California Press.

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

[26] Christopher Morris, Kristian Kersting, and Petra Mutzel. Glocalized weisfeiler-lehman graph kernels: Global-local feature maps of graphs. In *2017 IEEE International Conference on Data Mining (ICDM)*, pages 327–336. IEEE, 2017.

[27] Andrew Y Ng, Michael I Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in neural information processing systems*, pages 849–856, 2002.

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

[29] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk. *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '14*, 2014.

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

[31] 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.

[32] 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.

[33] 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.

[34] Rajhans Samdani, Ming-Wei Chang, and Dan Roth. Unified expectation maximization. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 688–698, 2012.

[35] Nino Shervashidze, Pascal Schweitzer, Erik Jan van Leeuwen, Kurt Mehlhorn, and Karsten M. Borgwardt. Weisfeiler-lehman graph kernels. *Journal of Machine Learning Research*, 12(77):2539–2561, 2011.

[36] 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 Composi-tionality*, page 57–66, 2015.

[37] Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. Complex embeddings for simple link prediction. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, page 2071–2080. JMLR.org, 2016.

[38] Zhao Xu, Volker Tresp, Kai Yu, and Hans-Peter Kriegel. Infinite hidden relational models. *CoRR*, abs/1206.6864, 2012.

[39] 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.

[40] Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. Embedding entities and relations for learning and inference in knowledge bases. *arXiv preprint arXiv:1412.6575*, 2014.

[41] Zhao Zhang, Fuzhen Zhuang, Meng Qu, Fen Lin, and Qing He. Knowledge graph embedding with hierarchical relation structure. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3198–3207, Brussels, Belgium, October-November 2018. Association for Computational Linguistics.

A Propositionalization schemes for CONFORMA

A.1 In-range, In-domain Representation

Combining latent and graph-based approaches is a promising area of research [28], which has shown to increase predictive power. Learning concepts from representations built using rule-based reasoning and training a latent feature model on the augmented dataset – as we do using CONFORMA – can be seen as a novel way of marrying these two complimentary approaches.

ILP approaches hinge upon defining a set of first-order clauses, which in turn induce a disjuctive hypothesis. To begin with, we induce two very simple clauses, `in_range` and `in_domain`, which test whether a given entity participates in a triple with relation r as either object or subject. Formally:

$$\texttt{in\_range}(e, r) = \begin{cases} \text{True} & \text{if } \exists\,(e, r, o) \in \mathcal{G}, \text{ for any } o \\ \text{False} & \text{otherwise} \end{cases} \tag{1}$$

$$\texttt{in\_domain}(e, r) = \begin{cases} \text{True} & \text{if } \exists\,(s, r, e) \in \mathcal{G}, \text{ for any } s \\ \text{False} & \text{otherwise} \end{cases} \tag{2}$$

For a given entity e and some relation r, we construct a vector $\hat{\mathbf{p}} \in \mathbb{B}^2$ such that:

$$\begin{aligned} &(\texttt{in\_range}(e, r) \rightarrow \hat{\mathbf{p}}_1 = 1) \wedge (\neg\texttt{in\_range}(e, r) \rightarrow \hat{\mathbf{p}}_1 = 0) \\ &(\texttt{in\_domain}(e, r) \rightarrow \hat{\mathbf{p}}_2 = 1) \wedge (\neg\texttt{in\_domain}(e, r) \rightarrow \hat{\mathbf{p}}_2 = 0) \end{aligned} \tag{3}$$

Each entity e is then represented by a vector $\mathbf{p}$ created by concatenating $\hat{\mathbf{p}}$ vectors which test the participation of e for every relation $r \in \mathcal{R}$. Hence, $\mathbf{p} \in \mathbb{B}^{2N_r}$ where $N_r = |\mathcal{R}|$ is the number of relations in the graph.

A.2 Random Paths Representation

The random paths representation used in this work is based on the algorithm proposed by [4]. First, for each entity, e, we generate n random paths starting at e of maximum length L, where a path is defined as a list of triples of the form $(e_i, r_j, e_k, direction)$. Direction specifies whether the we travel along the relation in forward or inverse direction. The paths are generated such that we avoid an immediate reversal step, i.e. steps of the type $(e_1, r, e_2), (e_2, r^{-1}, e_1)$. We also check whether loops have occurred, where we define loops as having traveled along a segment $(e_1, r_1, e_2, r_2, e_3, r_3, e_4)$ more than once in a single path. If the path is of length greater than six and we have encountered a loop, the path is terminated. A simplified algorithm for obtaining paths is shown in Algorithm 3.

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 $\mathbf{p} \in \mathbb{R}^{2N_r}$.

B Complexity of CONFORMA and CONFORMAE

The worst-case time and space complexity of CONFORMA depend on the propositionalization, clustering and neural link prediction model, and would equal the complexity of the slowest step of the three. For our experiments, the propositionalization techniques involved have the following complexities. Range-based propositionalization takes $\mathcal{O}(|\mathcal{G}|)$ in time since with one pass over the KG triples we can build the propositional embeddings $\mathbf{P}$ which will require exactly $\Theta(2N_eN_r)$ space. For the random-path propositionalization instead, the time complexity is $\mathcal{O}(kL)$, where L is the max length of a path and k is the number of paths and its space complexity $\Theta(kN_e)$.

Concerning the clustering step, for a vanilla spectral clustering implementation the complexity would be dominated by the $\mathcal{O}(N_e^3)$ cost of computing the SVD matrix of $\mathbf{P}$. Alternatively, a simpler K-Means would take $\mathcal{O}(tkN_eN_c)$ time, where t is the number of iterations and k the embedding size.

For the cost of training and evaluating neural link prediction models, under certain losses, we refer the reader to their respective papers [2, 37, 39, 8, 20]. We refer to [33] for a comparison of different

Algorithm 3 Generate Random Paths

function GENERATERANDOMPATHS($\mathcal{G}$)
  for $e \in \mathcal{E}$ **do**
    for $i \in \{1, \ldots, n\}$ **do**
      for $j \in \{1, \ldots, L\}$ **do**
        outgoingEdges $\leftarrow$ `GetEdges`(e)
        if previousEdge is not **None then**
          outgoingEdges.Remove(`Inverse`(previousEdge))
        newEdge $\leftarrow$ `RandomChoice`(outgoingEdges)
        paths.`Append`(newEdge)
        previousEdge $\leftarrow$ newEdge
        if `Length`(path > 6) and `DetectLoops`(path) **then**
          Break

function DETECTLOOPS(path)
  return True if loops in *path*, **else False**

function INVERSE(edge : $(e_1, r, e_2, \text{direction})$)
  return $(e_1, r, e_2, -\text{direction})$

function GETEDGES(e)
  return a set of all outgoing edges from entity e, $\{(e, r_i, e_j, \text{direction})\}$, where direction $\in \{-1, 1\}$ specifies whether the relation r is forward or reciprocal.

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. For instance, inference in ComplEx for all entities and relations would take $\mathcal{O}\left(N_e + N_c k + N_r k\right)$ time.

Concerning CONFORMAE, the space complexity of its M-step is simply the complexity of the score function in a neural link prediction model training procedure, as discussed above. In the E-step, on the other hand, we need to evaluate the score and loop through all of the entities and all the concepts, which results in $\mathcal{O}\left(N_e N_c\right)$ iterations. Note that, in practice, this step can be easily and efficiently parallelized on a GPU.

C Implementation Details

Here, we describe the experimental setup required to replicate the results in Table 3. The parameter ranges and propositionalization choices were guided by the preliminary results described in section D.

C.1 Baselines

To obtain the baselines, we ran the same grid search for all of the datasets on both, ComplEx [37] and DistMult [39], with the ranks set to [50, 100, 500] for UMLS, [500, 1000, 2000, 4000] for WN18RR and FB15K-237 and [100, 200] for Hetionet. 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 $\{10^{-3}, 5\times10^{-3}, \ldots, 10^{-1}, 5\times10^{-1}\}$. We considered two regularisers – the Frobenius norm [40, 37] and the nuclear N3 norm [20] – and consistently found the N3 superior, as suggested in [20]. For UMLS, WN18RR and FB15K-237 we used the provided train/validation/test splits, while for Hetionet, which is a relatively new biomedical dataset and does not have an established split, we held out 50k triples for validation and a further 50k for testing. We trained each model till convergence for 100 epochs and every 3 epochs computed the filtered Mean Reciprocal Rank (MRR) and HITS@K [3] on the validation and test sets. The highest validation performance was extracted and the corresponding test performance was reported. For all experiments we used the standard multi-class loss proposed by [20], and the AdaGrad optimizer [10].

Table 2: Bins for categorizing entities and relations into sub-populations based on their frequency, N, in the training set.

	WN18RR		FB15K-237	
Sub-population	Entities	Predicates	Entities	Predicates
Rare	$N \leq 3$	$N < 10^3$	$N \leq 20$	$N < 10^2$
Medium	$3 < N \leq 15$	$10^3 < N \leq 10^4$	$20 < N \leq 100$	$10^2 < N \leq 10^3$
Common	$N > 15$	$N > 10^4$	$N > 100$	$N > 10^3$

C.2 ConFormA

The ConFormA results reported in Table 3 were obtained by generating and clustering a random paths representation (see section A.2). To generate the representations we used the values suggested in literature [29] to guide our choice of parameters range, using a *minimum path length* of 2, *maximum path length* in [3, 5, 10, 20, 30] and two *number of paths* parameters: 32 and 64. This set of ranges resulted in a large number of parameter combinations. We randomly sampled six parameter combinations for UMLS, WN18RR and FB15K-237 and limited the number to three for Hetionet. We clustered the representations using the Spectral Clustering algorithm [27] , using the default parameters, and the number of clusters was in [50, 100, 500, 1000] for WN18RR, FB15K-237 and Hetionet, while for UMLS the range was reduced to [30, 50, 100]. For training the KGE model we used the corresponding baseline hyperparameters for the given embedding size. As before, we extracted the highest validation performance and reported the corresponding performance on the test set.

C.3 ConFormAE

To obtain the ConFormAE results quoted in Table 3 we randomly initialized the cluster assignments, with the initial number of clusters in [50, 100, 500, 1000] and performed the E-step after every epoch. As above, we used the same KGE hyperparameters as used for the baseline.

C.4 Entity and Relation Sub-populations

The bins used to categorize entities and relations into their frequency-based sub-populations, shown in Table 2, were constructed by considering the total number of training examples and exploring the entity and predicate distributions in each dataset.

D Clustering KGE with ConFormA

The implementations described in section C.2 were motivated by a series of preliminary experiments. We initially considered four different clustering algorithms: K-Means [24], Spectral Clustering [27], Affinity Propagation [13], and DBSCAN [11]. All of the implementations were done using the `sklearn` library[4] and the default parameters were used. It quickly became apparent that Affinity Propagation was significantly slower than the other three algorithms while producing clusters of similar quality. Moreover, despite DBSCAN being able to produce good quality clusters when carefully fine-tuned, we found that its sensitivity to parameter choice is ill-suited to experimenting with ConFormA, where we are often considering multiple propositionalization schemes. Hence, we focused primarily on Spectral Clustering and K-Means.

We began by considering four different propositionalization approaches for the ConFormA framework: the KGE embeddings, e.g., ComplEx, the fast approximation of the Weisfeiler-Lehman (WL) graph kernel [5, 35], the In-range, In-domain representation (A.1) and random paths representation (A.2). Obtaining the WL kernel representation proved intractable due to the large number of entities in benchmark datasets, such as WN18RR, which in turn resulted in huge kernel matrices. Hence, we focused on the other three approaches. The KGE representation was obtained by simply extracting the baseline KGE embeddings corresponding to the highest validation MRR. The In-range, In-domain

[4] https://scikit-learn.org/stable/modules/clustering.html

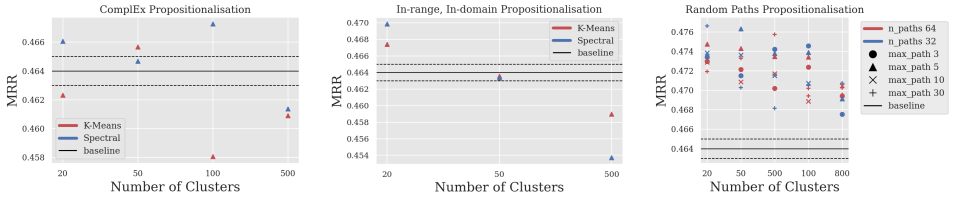

Figure 2: Different propositionalization schemes for ConFormA on WN18RR at rank 100 using ComplEx. All results were obtained using the same hyperparameters as the baseline model. For the random paths representation Spectral Clustering was used.

representation is deterministic and does not require specifying any parameters, while the random paths representations were obtained as described in section C.2.

Performance of the different propositionalization approaches on WN18RR with $k = 100$ is shown in Figure 2, where we also explored a range of values for the number of clusters parameter and explored using both, Spectral Clustering and K-Means, for the ComplEx and ILP representations while limiting random paths experiments to Spectral Clustering. To obtain the baseline, we performed an initial gridsearch as described in section C.1, but with batch-sizes limited to 500. The standard deviation was attained by training the baseline model five times using different seeds.

Firstly, we found that introducing concept assignment relationships learned from KGE representations lead to little, if any, improvement upon the baseline. One hypothesis as to why this might be the case lies in that the concepts learned in this way are not introducing any new information and only reinforcing the latent structure that can already be learned by an out-of-the-box KGE. While ILP showed some promise for low numbers of clusters, ConFormA trained on any random paths representation outperformed the out-of-the-box KGE even for large numbers of clusters. This suggests that using propositionalization approaches such as ILP and random paths might introduce information that KGE models struggle to capture otherwise. With respect to the number of clusters learned, for both ILP and random paths representations there appears to be a preference for a smaller number of clusters. Lastly, we also note that there seems to be no clear advantage of using either K-Means or Spectral Clustering.

Table 3: Mean Reciprocal Rank (MRR) and Hits (H) at 1, 3, 10 for ConFormA and ConFormAE when using DistMult or ComplEx as baseline models on WN18RR and FB15K-237 KGs for different values of embedding sizes (k). Best values for each metric and k are in bold.

			DistMult				ComplEx			
	k	Model	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10
UMLS	50	Baseline	75.51	66.87	80.41	91.91	94.71	90.77	**98.71**	99.62
		ConFormA	**77.23**	**69.74**	**81.39**	91.45	**95.46**	**92.44**	98.18	99.62
		ConFormAE	76.53	68.68	81.32	**92.13**	95.39	92.13	98.41	**99.77**
	100	Baseline	76.19	68.08	80.48	91.38	96.00	**93.42**	98.49	99.70
		ConFormA	77.08	69.74	80.94	**91.45**	**96.30**	**93.42**	**99.17**	**99.85**
		ConFormAE	**77.42**	**70.27**	**81.01**	**91.45**	94.65	90.70	98.41	99.55
	500	Baseline	76.75	69.29	80.56	91.68	96.97	94.93	99.02	99.77
		ConFormA	**77.01**	**69.97**	**80.58**	**91.78**	**97.47**	**95.84**	**99.17**	**99.78**
		ConFormAE	76.45	69.14	80.41	91.07	97.04	95.08	98.94	99.70
WN18RR	500	Baseline	44.40	40.30	45.63	52.81	47.85	43.41	49.44	56.57
		ConFormA	44.19	39.96	45.33	52.94	48.55	43.94	50.24	**57.67**
		ConFormAE	**44.89**	**40.84**	**46.00**	**53.38**	**48.77**	**44.42**	**50.48**	57.29
	1000	Baseline	44.80	41.02	45.80	52.52	48.34	43.81	50.18	56.99
		ConFormA	**45.47**	**41.16**	**46.76**	**54.28**	49.12	44.42	**50.83**	**58.70**
		ConFormAE	45.20	40.97	46.41	54.05	**49.25**	**44.81**	50.81	58.33
	2000	Baseline	45.20	41.05	**46.39**	53.75	48.62	44.07	50.34	57.28
		ConFormA	44.93	40.60	45.98	53.67	49.40	**44.80**	**50.86**	**59.00**
		ConFormAE	**45.38**	**41.16**	**46.39**	**54.04**	**49.42**	45.20	50.41	58.42
	4000	Baseline	**45.68**	**41.35**	**47.19**	54.18	48.78	44.34	50.27	57.43
		ConFormA	45.56	41.18	46.75	54.56	49.36	44.77	**50.96**	58.38
		ConFormAE	45.66	**41.35**	46.76	**54.64**	**49.42**	**44.91**	50.83	**58.73**
FB15K237	500	Baseline	34.88	25.56	38.34	53.52	35.89	26.47	39.31	54.82
		ConFormA	34.92	25.65	38.39	53.54	36.08	**26.77**	**39.39**	55.00
		ConFormAE	**35.01**	**25.72**	**38.40**	**53.65**	**36.13**	26.76	39.46	**55.09**
	1000	Baseline	35.26	25.83	**38.82**	54.26	36.18	26.69	**39.81**	55.21
		ConFormA	35.30	25.91	38.78	**54.28**	36.26	**26.88**	39.74	55.22
		ConFormAE	**35.40**	**26.11**	38.77	**54.28**	**36.27**	26.84	**39.81**	**55.34**
	2000	Baseline	35.47	26.13	38.76	54.42	**36.37**	**27.01**	**39.89**	**55.45**
		ConFormA	35.55	26.18	**39.03**	54.32	**36.37**	26.99	**39.89**	55.19
		ConFormAE	**35.62**	**26.31**	39.02	**54.43**	36.34	26.96	39.81	55.36
	4000	Baseline	35.68	26.21	39.33	54.54	36.21	26.81	39.67	55.30
		ConFormA	**35.83**	26.42	**39.42**	**54.76**	36.46	27.04	39.98	**55.50**
		ConFormAE	35.81	**26.47**	39.16	54.61	**36.48**	**27.05**	**40.11**	55.47

E Complete results for link prediction

Table 4: Mean reciprocal rank (MRR) and Hits (H) at 1, 3, 10 for ConFormA and ConFormAE when using ComplEx as baseline models on Hetionet KG for two different values of embedding sizes (k). Best values for each metric and k are in bold.

			ComplEx			
	k	Model	MRR	H@1	H@3	H@10
Hetionet	100	Baseline	29.60	22.05	32.37	44.46
		ConFormA	29.74	22.16	32.56	44.40
		ConFormAE	**29.84**	**22.24**	**32.66**	**44.65**
	200	Baseline	32.83	25.45	35.60	47.25
		ConFormA	32.96	25.39	**35.98**	47.53
		ConFormAE	**33.02**	**25.52**	35.95	**47.66**

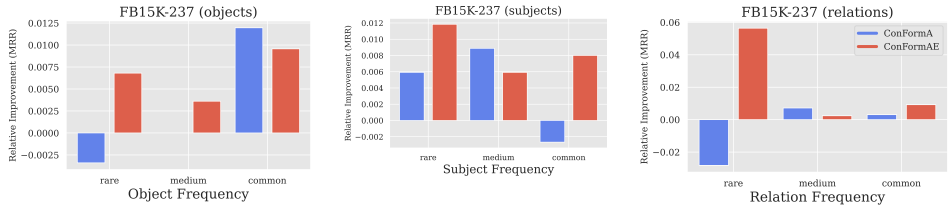

Figure 3: Relative improvement in terms of MRR of ConFormA and ConFormAE over the ComplEx baseline for FB15K-237 with $k = 100$.

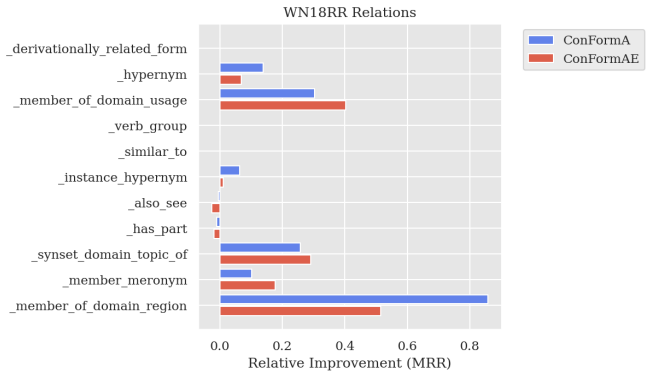

Figure 4: Relative improvement in terms of MRR of ConFormA and ConFormAE over the ComplEx baseline for relations in WN18RR with $k = 100$.

F Interpreting Concepts learned by ConFormAE

Table 5: Example of concepts learned by ConFormAE on UMLS, initialized using 50 random cluster assignments.

Concept 1	**Concept 7**	**Concept 9**
cell_or_molecular_dysfunction	activity	biologic_function
disease_or_syndrome	behavior	cell_function
experimental_model_of_disease	daily_or_recreational_activity	genetic_function
injury_or_poisoning	diagnostic_procedure	mental_process
mental_or_behavioral_dysfunction	educational_activity	molecular_function
neoplastic_process	governmental_or_regulatory_activity	natural_phenomenon_or_process
pathologic_function	health_care_activity	organ_or_tissue_function
Concept 2	health_care_related_organization	organism_function
alga	individual_behavior	physiologic_function
amphibian	laboratory_procedure	**Concept 10**
animal	machine_activity	amino_acid_sequence
archaeon	molecular_biology_research_technique	body_location_or_region
bacterium	occupational_activity	body_system
bird	organization	carbohydrate_sequence
fish	professional_society	classification
fungus	research_activity	clinical_drug
human	self_help_or_relief_organization	conceptual_entity
invertebrate	social_behavior	drug_delivery_device
mammal	therapeutic_or_preventive_procedure	entity
organism	**Concept 8**	finding
plant	amino_acid_peptide_or_protein	functional_concept
reptile	antibiotic	geographic_area
rickettsia_or_chlamydia	biologically_active_substance	group_attribute
vertebrate	biomedical_or_dental_material	idea_or_concept
virus	body_substance	intellectual_product
Concept 3	carbohydrate	laboratory_or_test_result
environmental_effect_of_humans	chemical	language
event	chemical_viewed_functionally	manufactured_object
human_caused_phenomenon_or_process	chemical_viewed_structurally	medical_device
phenomenon_or_process	eicosanoid	molecular_sequence
qualitative_concept	element_ion_or_isotope	nucleotide_sequence
quantitative_concept	enzyme	regulation_or_law
temporal_concept	food	research_device
Concept 4	hazardous_or_poisonous_substance	sign_or_symptom
acquired_abnormality	hormone	spatial_concept
age_group	immunologic_factor	**Concept 11**
anatomical_abnormality	indicator_reagent_or_diagnostic_aid	biomedical_occupation_or_discipline
congenital_abnormality	inorganic_chemical	occupation_or_discipline
family_group	lipid	**Concept 12**
group	neuroreactive_substance_or_biogenic_amine	anatomical_structure
patient_or_disabled_group	nucleic_acid_nucleoside_or_nucleotide	body_part_organ_or_organ_component
population_group	organic_chemical	body_space_or_junction
professional_or_occupational_group	organophosphorus_compound	cell
Concept 5	pharmacologic_substance	cell_component
clinical_attribute	receptor	embryonic_structure
organism_attribute	steroid	fully_formed_anatomical_structure
Concept 6	substance	gene_or_genome
physical_object	vitamin	tissue