A Closer Look at Knowledge Base Completion

Corbin Rosset
Department of Computer Science
Johns Hopkins University, Baltimore, MD, 21218
crosset2@jhu.edu

Abstract

Knowledge bases are an effective tool for structuring and accessing large amounts of multi-relational data, but they are often woefully incomplete, especially in broader domains. We consider the task of learning low dimensional embeddings for Knowledge Base Completion and make the following contributions: 1) a novel embedding model, ModelE-X, that uses few parameters yet outperforms many state-of-the-art, more complex algorithms, 2) the realization that the often-unreported metric of relation ranking yields valuable insights into algorithms' behavior and 3) we scrutinize macro vs. micro-averaging of ranking metrics and discuss which is a better indicator of generalizability.

1 Introduction

Knowledge Bases (KBs) represent structured information in a way that is convenient for inference and computation. Internet companies leverage KBs to improve search results on factoid queries, such as Google's Knowledge Vault (Dong et al., 2014), Microsoft's Satori (Nickel et al., 2015) and are indispensable for tasks like question answering (Fader et al., 2014; Yih and Ma, 2016; Yao and Van Durme, 2014; Xu et al., 2016)

Personalized agents, "infobots", and reading comprehension engines perform complicated, multi-hop inference over vast amounts of facts. Researchers have approached this challenge with deep learning models which rely on vector encodings of facts, making knowledge representation learning a critical and relevant task (Shen et al., 2016; Hermann et al., 2015; Weston et al., 2014).

A KB defined over a schema of E entities from a set $\mathcal{E}$ and R relations from a set $\mathcal{R}$ is a set of triples $\mathcal{T} = \{(h, r, t)\}$ where $h, t \in \mathcal{E}, r \in \mathcal{R}$, which can be interpreted as a knowledge graph (KG) with edge labels from $\mathcal{R}$ and node labels from $\mathcal{E}$. An embedding for h is denoted $\boldsymbol{e_h} \in \mathbb{R}^d$, which are used interchangeably. Knowledge bases over any useful domain are incomplete, as they are often constructed by hand or semi-automatically (Socher et al., 2013; West et al., 2014). Knowledge Base completion (KBC) is a well-studied task of determining which triples ought to be in the KB, which is inherently a ranking problem.

There are three main categories of KBC algorithms: those which learn from only one-hop triples in $\mathcal{T}$ directly, those using deeper structure or multi-hop paths in the KG, and those leveraging textual resources with entity linkers and other NLP machinery. Deep learning has been applied to all three categories.

2 Knowledge Base Completion

Here we primarily discuss embeddings or latent feature models for KBC, which learn dense parameterizations of entities and relationship operators and are adept at modeling global patterns in large, noisy KGs (Nickel et al., 2015). Modern KB schemas define potentially millions of entities and billions of facts. Having a fixed d-dimensional embedding of each entity 1) simplifies storage requirements, 2) allows for natural comparisons between entities and 3) a mathematical structure for interpreting relationships as linear algebra operators over entities, which allows for capturing long-range structural information in the KG (Bordes et al., 2011).

The framework for traditional automatic KBC algorithms use the same machinery: in addition to learning embeddings and operators, each model typically defines a scoring function $f(h, r, t)$ of a triple over those embeddings, a pairwise margin ranking objective[1], and a method of sampling "false" triples under the somewhat unrealistic "closed world" assumption that any triple not in the overall KB is untrue. The following loss function contrasts positive and negative triples (regularization term omitted)

$$\mathcal{L}(ALG) = \sum_{p \in \mathcal{T}, n \notin \mathcal{T}} [\gamma + f_{ALG}(p) - f_{ALG}(n)]_+$$

and is minimized by mini-batch gradient descent over (p, n) pairs where $p = (h, r, t)$ is a sampled positive triple, $n \in \{(h', r, t), (h, r', t), (h, r, t')\}$ is a sampled negative triple with exactly one slot corrupted s.t. $n \notin \mathcal{T}$, and $[\cdot]_+ = max(0, \cdot)$ In our case, whenever we sample p, we also add all three corrupted versions of it independently and uniformly at random.

We evaluate the model on the Link Prediction task, which is to predict, say, the best h given r and t to yield a triple that is most likely to belong the KB[2]. A list of scores is generated by applying $f_{ALG}(h', r, t)\ \forall h' \in \mathcal{E}$, which, upon sorting, hopefully yields a highly-ranked h that makes the triple true. The loss attempts to concentrate mass on observed triples, but it provides only a rough approximation to the ideal list (specifically, it might not score obviously incorrect triples any lower than partially incorrect triples, as long as the positive triple is scored above both). We report several ranking metrics: Mean Rank, Median Rank, Mean Reciprocal Rank (MRR), and Hits@10 (Hang, 2011). In practice not all $h' \in \mathcal{E}$ should be included in the list, only those for which it is known $(h', r, t) \notin \mathcal{T}$ so that the model isn't penalized for ranking one correct answer over another; this is known as the "filtered" (as opposed to "raw") metric.

3 Prior and Related Work

There are a number of models that approach KBC as a tensor or matrix factorization problem (Nickel et al., 2011, 2012; Wang and Cohen, 2016) which inspired others to define entity and relation specific embeddings (Bordes et al., 2011; Socher et al., 2013; Garcia-Duran et al., 2015). We discuss only those models we re-implement as part of this work, and we refer the reader to the canonical works of Riedel et al. (2013); Bordes et al. (2013).

ModelE defines two vectors in $\mathbb{R}^d$ for each relation to allow only certain entities in the head and tail position of a triple. That is, for the score of a triple to be high, both the head and tail entities must align with their respective relation embeddings components $\boldsymbol{r_h}$ and $\boldsymbol{r_t}$ (Riedel et al., 2013):

$$f_{ModelE}(h, r, t) = \boldsymbol{e_h}^T \boldsymbol{r_h} + \boldsymbol{e_t}^T \boldsymbol{r_t}$$

It has $Ed + 2Rd$ parameters, and aims to give high scores to true triples. Inspired by the semantically meaningful translations of word embeddings, the TransE model on a positive triple (h, r, t) learns embeddings such that $\boldsymbol{e_h} + \boldsymbol{r} \approx \boldsymbol{e_t}$ (Bordes et al., 2013)

$$f_{TransE}(h, r, t) = \|\boldsymbol{e_h} + \boldsymbol{r} - \boldsymbol{e_t}\|$$

It has $Ed + Rd$ parameters and seeks a low score for positive triples. For Bilinear (DistMult) and BilinearDiag (Yang et al., 2014), each relation r is parameterized by $W_r \in \mathbb{R}^{d \times d}$ (which is constrained to be diagonal for BilinearDiag).

$$f_{Bilinear}(h, r, t) = \boldsymbol{e_h}^T W_r \boldsymbol{e_t}$$

Bilinear has $Ed + Rd^2$ parameters, which can be quite slow and prone to overfitting.

A parallel approach to KBC utilizes features in the structure of the graph, particularly paths therein, and train on multi-hop path queries, where, for example, the appropriate tail entity is sought after starting at a head entity and traversing a path of relations (Guu et al., 2015), sometimes with a notion of a path probability (Lao et al., 2011; Lin et al., 2015). Some models like TransE and Bilinear are naturally compositional and suited for this setting (Garcıa-Durán et al.), other times heavier compositional models like LSTMs are employed (Neelakantan et al., 2015). There is another very important community devoted to leveraging textual mentions of KB triples (Toutanova et al., 2015; Han et al., 2016; Wang and Li, 2016; Wang et al., 2014; Weston et al., 2013). More recently, sophisticated deep learning approaches such as ReasoNet(Shen et al., 2016) inspired by memory neural networks(Weston et al., 2014) have been applied to KBC(Shen et al., 2017). And, log-linear models have shown outstanding performance considering their simplicity (Toutanova et al.)

[1] Toutanova et al. (2015); Toutanova and Chen (2015) instead maximize the conditional log likelihood, but the pairwise ranking loss is more scalable.

[2] Simiarly we also predict the best r or best t.

4 Our Approach

We introduce a novel vector parameterization of entities and relations inspired by ModelE, which we name ModelE-X (Model-E Extended)[3]. Like ModelE, ModelE-X also defines two relation vectors, but provides three enhancements that improve expressiveness and flexibility:

$$f_{ModelE-X}(h,r,t) = \|\boldsymbol{e_h} \odot \boldsymbol{r_h} - \boldsymbol{e_t} \odot \boldsymbol{r_t}\|$$

where $\odot$ is elementwise vector multiplication (again a true triple should score high).

1. ModelE-X modulates the response of a relation component (e.g. $\mathbf{r_h}$) to its argument at a much finer granularity than a simple dot product due to the lack of a sum in element-wise product (inspired by the gates of an LSTM)

2. Whereas ModelE can mistakenly give a high score to a false triple if one of the two dot products is high enough, ModelE-X requires the two response vectors to be similar, which is much less likely to occur spuriously.

3. We allow any choice of dissimilarity metrics, like ℓ_1 or ℓ_2, between response vectors (ℓ_1 works better in practice).

These advantages come without any sacrifice to complexity or runtime ($Ed + 2Rd$ parameters).

Along a similar vein, we questioned whether an element-wise relation operator was too simple, and perhaps a separate matrix for the head and tail arguments was needed. We introduce "ModelE-XL" (Model-E Extended, Linear)

$$f_{ModelE-XL}(h,r,t) = \|\boldsymbol{W_r^h} \boldsymbol{e_h} - \boldsymbol{W_r^t} \boldsymbol{e_t}\|$$

as the natural generalization of ModelE-X (it reduces to ModelE-X if W_r^h and W_r^t are diagonal). Both ModelE-X and ModelE-XL give high scores to true triples.

5 Experiments

See Bordes et al. (2013) for a description of the FB15k dataset we use. We re-implemented many canonical algorithms to eliminate sources of error when comparing results.

We tuned the margin γ over $\{0.2, 0.5, 1.0, 1.5, 2.0\}$, we set the dimension d of entity and relation embeddings to be 100 in all the models we implement except Bilinear, which was 50, but we acknowledge that nearly all models improve with more dimensions[4] We choose between ℓ_1 and ℓ_2 dissimilarity metrics, but in nearly every case ℓ_1 was superior (and faster).

We ran minibatch gradient descent with a batch size of one one-hundredth the size of the training set, with constant step size for up to 500 epochs, with early stopping if MRR did not improve after 30 epochs. We ran the dev set every 10 epochs. We re-normalized all entity embeddings to unit ℓ_2 norm after every minibatch, and we regularized relation-specific parameters with ℓ_2 norm (a regularization coefficient of 0.01 was satisfactory).

6 Results and Discussion

The "Micro" partition of Table 1 shows that ModelE-X outperforms or is competitive with state-of-the-art embedding models such as STransE (Nguyen et al., 2016) and more sophisticated algorithms that consider expensive path training (PTransE) (Lin et al., 2015) or hand-crafted log-linear features (Node+LinkFeat) (Toutanova et al.), and deep models (IRN) (Shen et al., 2017). These models are more expensive to train as they require more parameters, more samples, or human intervention.

In addition to our new model, this paper calls attention to how the community evaluates KBC algorithms. Virtually no publications report relation ranking metrics, which we argue is an oversight because it is so easy to obtain. For example, Table 1 suggests that in the usual micro average setting, weaknesses in TransE and BilinearDiag become more readily apparent based on relation metrics rather than entity ranking metrics alone, since they attain relatively higher Mean Ranks and lower MRRs, while their *entity* Hits@10 and MRRs do not raise concern. These weaknesses become even more apparent in the macro case. Relation ranking also reveals that Bilinear and ModelE (and ModelE-X) behave more similarly than previously thought, as they have similar MRR and Hits@10 on relations in both the macro and micro

[3] We make our code, experiments, and logs available at www.ANONYMIZEDLINK.com.

[4] For instance, Toutanova and Chen (2015) achieve Hits@10 = 79.7 for Bilinear with $d = 500$ and Yang et al. (2014) achieve Hits@10 of 57.7 for $d = 100$, which we corroborate.

Micro Averages For Each Method	**Mean Rank** Rel	Entity	**Median Rank** Rel	Entity	**MRR** (×100) Rel	Entity	**Hits@10** (%) Rel	Entity
Unstructured	NA	1074 / 979	NA	-	NA	-	NA	4.5 / 6.3
UnstructuredDot*	NA	1172 / 1077	NA	377 / 312	NA	2.9 / 3.83	NA	7.02 / 9.14
ModelE*	**2.01 / 1.66**	460 / 363	1 / 1	84 / 50	79.2 / 89.9	13.8 / 22.1	99.1 / 99.2	24.7 / 34.0
Bilinear* (50)	3.09 / 2.75	182 / 84.4	1 / 1	21 / 7	**83.3 / 94.4**	20.9 / 36.1	**99.5 / 99.5**	38.7 / 56.1
BilinearDiag*	7.06 / 6.74	229 / 129	2 / 2	33 / 17	54.1 / 56.8	15.0 / 23.3	89.9 / 90.8	30.8 / 42.5
TransE*	5.3 / 4.9	800 / 725	2 / 2	79 / 46	54.1 / 58.9	13.6 / 19.6	94.1 / 94.6	24.9 / 32.4
IRN (Shen 2017)	-	**- / 38**	-	-	-	-	-	**- / 92.7**
PTransE	-	207 / 58	-	-	-	-	-	51.4 / 84.6
STransE	-	219 / 69	-	-	-	**25.2 / 54.3**	-	51.6 / 79.7
Node+LinkFeat	-	-	-	-	-	- / 82.2	-	- / 87.0
Our ModelE-X*	2.89 / 2.56	186 / 82.5	1 / 1	16 / 5	77.6 / 86.4	23.6 / 40.6	97.1 / 97.3	43.3 / 62.9
ModelE-X* (200)	2.68 / 2.34	188 / 80	1 / 1	11 / 2	79.7 / 89.4	27.1 / 53.8	98.1 / 98.2	49.8 / 76.6
ModelE-X* (500)	2.37 / 2.04	**168 / 55**	1 / 1	9 / 1	79.5 / 88.4	**29.3 / 64.4**	98.4 / 98.5	**53.4 / 83.4**

Macro Averages For Each Method	**Mean Rank** Rel	Entity	**Median Rank** Rel	Entity	**MRR** (×100) Rel	Entity	**Hits@10** (%) Rel	Entity
ModelE*	**4.72 / 4.08**	151 / 129	**4 / 3**	13.5 / 9	54.7 / 66.6	25.9 / 34.5	93.7 / 94.3	47.9 / 56.5
Bilinear*	14.2 / 13.6	172 / 151	13 / 12	11 / 6	**60.4 / 74.5**	30.1 / 41.3	**92.3 / 92.8**	52.2 / 62.0
BilinearDiag*	53.7 / 53.1	284 / 264	50 / 49	41 / 33	14.8 / 15.9	15.9 / 19.8	35.1 / 36.2	30.6 / 35.3
TransE*	61.7 / 61.1	745 / 725	58 / 57	23 / 19	24.3 / 27.6	22.0 / 27.6	59.9 / 61.5	39.3 / 45.1
Our ModelE-X*	37.2 / 36.6	121 / 100	36 / 35	8 / 3.5	30.5 / 36.3	35.4 / 48.4	58.7 / 60.3	59.3 / 70.4
ModelE-X* (200)	38.9 / 38.2	151 / 129	37 / 37	6 / 2	38.5 / 46.6	38.7 / 56.9	69.1 / 70.2	64.2 / 78.7
ModelE-X* (500)	24.8 / 24.1	**88 / 65**	24 / 23	**4.5 / 1**	40.4 / 48.9	**44.3 / 66.9**	74.4 / 75.1	**68.7 / 84.1**

Table 1: Our ModelE-X is as simple as ModelE, yet is competitive or outperforms more sophisticated models. ModelE-X excels when all relations are treated equally (macro averaging across relations). Each value reports "raw" / "filtered" metrics on FB15k test set. Parenthesis indicate the embedding dimension, else it is 100; the authors implemented models with an asterisk. Values reported in entity columns represent the average of left and right entity ranking metrics; NA means "not applicable", and "-" means unreported.

cases. We note that relation ranking is useful in a number of tasks related to KBC, such as relation extraction from text (Mintz et al., 2009).

Nearly all publications report the micro-averages of their metrics, that is, the weighted average w.r.t the frequency of relations in the test triples. This might be an over-simplification, as it is often unclear whether poor ranking performance is a symptom of a skewed data distribution, or a sign the model truly lacks capacity to capture relevant patterns. The macro average decouples these factors to allow for clearer insight into the model's behavior. It is common to report performance on buckets of relation types (1-to-1, 1-to-Many, etc) (Bordes et al., 2013), or the macro average across a handful of relations; however, to get the most realistic perspective on how a model will generalize to new relations, the macro average should be used to treat all relations equally.

The macro average also reveals a massive discrepancy in TransE's relation metrics. TransE has been known to struggle with relationships that aren't 1-to-1 (Bordes et al., 2013), but the micro average clearly masks this systemic weakness behind a distribution of relationships TransE can grasp. Interestingly, ModelE and ModelE-X benefit from macro over micro rankings, at least w.r.t MRR and Hits@10 (bolded bottom row), suggesting that these models are adept at general reasoning across an entire schema.

Lastly, we challenge the common assumption that models trained to rank entities will have also learned to rank relations well. We compared two ModelE-X (300)'s where the loss function of the first did not include relation ranking, but the second did, all else being equal. Contrary to the belief of many, the first model suffered horrendously when tested on relation ranking: the first model had a filtered Rel Hits@10 of 51.7% (micro), whereas the second reported 98.1%. Similarly for the macro-averaged Rel metrics: the first model was 4.45% vs. 69.7% for the second. Clearly, good entity ranking does not imply good relation ranking.

References

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

Antoine Bordes, Jason Weston, Ronan Collobert, and Yoshua Bengio. 2011. Learning structured embeddings of knowledge bases. In *Conference on artificial intelligence*. EPFL-CONF-192344.

Xin Dong, Evgeniy Gabrilovich, Geremy Heitz, Wilko Horn, Ni Lao, Kevin Murphy, Thomas Strohmann, Shaohua Sun, and Wei Zhang. 2014. Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, pages 601–610.

Anthony Fader, Luke Zettlemoyer, and Oren Etzioni. 2014. Open question answering over curated and extracted knowledge bases. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, pages 1156–1165.

Alberto Garcıa-Durán, Antoine Bordes, and Nicolas Usunier. ???? Composing relationships with translations .

Alberto Garcia-Duran, Antoine Bordes, Nicolas Usunier, and Yves Grandvalet. 2015. Combining two and three-way embeddings models for link prediction in knowledge bases. *arXiv preprint arXiv:1506.00999* .

Kelvin Guu, John Miller, and Percy Liang. 2015. Traversing knowledge graphs in vector space. *arXiv preprint arXiv:1506.01094* .

Xu Han, Zhiyuan Liu, and Maosong Sun. 2016. Joint representation learning of text and knowledge for knowledge graph completion. *arXiv preprint arXiv:1611.04125* .

LI Hang. 2011. A short introduction to learning to rank. *IEICE TRANSACTIONS on Information and Systems* 94(10):1854–1862.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems*. pages 1693–1701.

Ni Lao, Tom Mitchell, and William W Cohen. 2011. Random walk inference and learning in a large scale knowledge base. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, pages 529–539.

Yankai Lin, Zhiyuan Liu, Huanbo Luan, Maosong Sun, Siwei Rao, and Song Liu. 2015. Modeling relation paths for representation learning of knowledge bases. *arXiv preprint arXiv:1506.00379* .

Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*. Association for Computational Linguistics, pages 1003–1011.

Arvind Neelakantan, Benjamin Roth, and Andrew McCallum. 2015. Compositional vector space models for knowledge base completion. *arXiv preprint arXiv:1504.06662* .

Dat Quoc Nguyen, Kairit Sirts, Lizhen Qu, and Mark Johnson. 2016. Stranse: a novel embedding model of entities and relationships in knowledge bases. *arXiv preprint arXiv:1606.08140* .

Maximilian Nickel, Kevin Murphy, Volker Tresp, and Evgeniy Gabrilovich. 2015. A review of relational machine learning for knowledge graphs. *arXiv preprint arXiv:1503.00759* .

Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. 2011. A three-way model for collective learning on multi-relational data. In *Proceedings of the 28th international conference on machine learning (ICML-11)*. pages 809–816.

Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. 2012. Factorizing yago: scalable machine learning for linked data. In *Proceedings of the 21st international conference on World Wide Web*. ACM, pages 271–280.

Sebastian Riedel, Limin Yao, Andrew McCallum, and Benjamin M Marlin. 2013. Relation extraction with matrix factorization and universal schemas .

Yelong Shen, Po-Sen Huang, Ming-Wei Chang, and Jianfeng Gao. 2017. Implicit reasonet: Modeling large-scale structured relationships with shared memory. *arXiv preprint arXiv:1611.04642* .

Yelong Shen, Po-Sen Huang, Jianfeng Gao, and Weizhu Chen. 2016. Reasonet: Learning to stop reading in machine comprehension. *arXiv preprint arXiv:1609.05284* .

Richard Socher, Danqi Chen, Christopher D Manning, and Andrew Ng. 2013. Reasoning with neural tensor networks for knowledge base completion. In *Advances in neural information processing systems*. pages 926–934.

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

Kristina Toutanova, Danqi Chen, Patrick Pantel, Hoifung Poon, Pallavi Choudhury, and Michael Gamon. 2015. Representing text for joint embedding of text and knowledge bases. In *EMNLP*. Citeseer, volume 15, pages 1499–1509.

Kristina Toutanova, Xi Victoria Lin, and Wen-tau Yih. ???? Compositional learning of embeddings for relation paths in knowledge bases and text .

William Yang Wang and William W Cohen. 2016. Learning first-order logic embeddings via matrix factorization. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI 2015), New York, NY, July. AAAI*.

Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge graph and text jointly embedding. In *EMNLP*. Citeseer, volume 14, pages 1591–1601.

Zhigang Wang and Juanzi Li. 2016. Text-enhanced representation learning for knowledge graph. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*. AAAI Press, pages 1293–1299.

Robert West, Evgeniy Gabrilovich, Kevin Murphy, Shaohua Sun, Rahul Gupta, and Dekang Lin. 2014. Knowledge base completion via search-based question answering. In *Proceedings of the 23rd international conference on World wide web*. ACM, pages 515–526.

Jason Weston, Antoine Bordes, Oksana Yakhnenko, and Nicolas Usunier. 2013. Connecting language and knowledge bases with embedding models for relation extraction. *arXiv preprint arXiv:1307.7973* .

Jason Weston, Sumit Chopra, and Antoine Bordes. 2014. Memory networks. *arXiv preprint arXiv:1410.3916* .

Kun Xu, Siva Reddy, Yansong Feng, Songfang Huang, and Dongyan Zhao. 2016. Question answering on freebase via relation extraction and textual evidence. *arXiv preprint arXiv:1603.00957* .

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

Xuchen Yao and Benjamin Van Durme. 2014. Information extraction over structured data: Question answering with freebase. In *ACL (1)*. Citeseer, pages 956–966.

Wen-tau Yih and Hao Ma. 2016. Question answering with knowledge base, web and beyond. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*. ACM, pages 1219–1221.