Embedding Multimodal Relational Data

Pouya Pezeshkpour
University of California
Irvine, CA
pezeshkp@uci.edu

Liyan Chen
University of California
Irvine, CA
liyanc@uci.edu

Sameer Singh
University of California
Irvine, CA
sameer@uci.edu

1 Introduction

Knowledge bases (KB) are an essential part of many computational systems with applications in variety of domains, such as search, structured data management, recommendations, question answering, and information retrieval. However, KBs often suffer from incompleteness, noise in their entries, and inefficient inference. Due to these deficiencies, learning the relational knowledge representation has been a focus of active research [1, 2, 32, 9, 19, 26, 3]. These approaches represent relational triples, consisting of a subject entity, relation, and an object entity, by estimating fixed, low-dimensional representations for each entity and relation from observations, thus encode the uncertainty and infer missing facts accurately and efficiently. The subject and the object entities come from a fixed, enumerable set of entities that appear in the knowledge base.

Knowledge bases in the real world, however, are rich with a variety of different data types. Apart from a fixed set of entities, the relations often not only include numerical attributes (such as ages, dates, financial, and geoinformation), but also textual attributes (such as names, descriptions, and titles/designations) and images (profile photos, flags, posters, etc.). Although these different types of relations cannot directly be represented as links in a graph over a fixed set of nodes, they can be crucial pieces of evidences for knowledge base completion. For example the textual descriptions and images might provide evidence for a person's age, profession, and designation. Further, this additional information still contains similar limitations as the conventional *link* data; they are often missing, may be noisy when observed, and for some applications, may need to be predicted in order to address a query. There is thus a crucial need for relational modeling that goes beyond just the link-based, *graph* view of knowledge-base completion, is able to utilize all the observed information, and represent the uncertainty of multimodal relational evidence.

In this paper, we introduce a multimodal embedding approach for modeling knowledge bases that contains a variety of data types, such as textual, images, numerical, and categorical values. Although we propose a general framework that can be used to extend many of the existing relational modeling approaches, here we primary apply our method to the DistMult approach [32]. We extend this approach that learns a vector for each entity and relation by augmenting it with additional neural encoders for different evidence data types. For example, when the object of a triple is an image, we encode it into a fixed-length vector using a CNN, while the textual attributes are encoded using sequential embedding approaches like LSTMs. The scoring module remains identical; given the vector representations of the subject, relation, and object of a triple, this module produces a score indicating the probability that the triple is correct. This unified model allows for flow of the information across the different relation types, enabling more accurate modeling of relational data.

We provide an evaluation of our proposed approach on two relational databases. Since we are introducing a novel formulation in the relational setting, we introduce two benchmarks, created by extending the existing YAGO-10 and MovieLens-100k datasets to include additional relations such as textual descriptions, numerical attributes, and images of the original entities. In our evaluation, we demonstrate that our model utilizes the additional information effectively to provide gains in link-prediction accuracy, and present a breakdown of how much each relation benefits from each type of the additional information. We also present results that indicate the learned multimodal

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

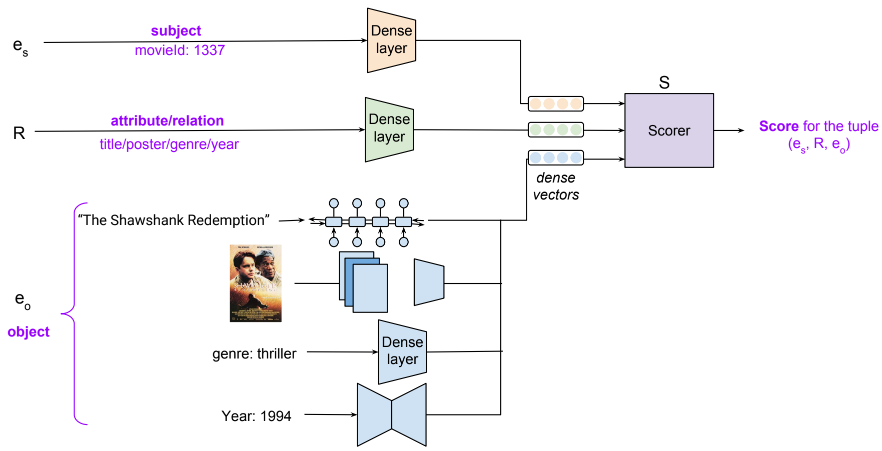


Figure 1: **Multimodal Embeddings:** Architecture of the proposed work that, given any movie and any of its attributes, like the title, poster, genre, or release year, uses domain-specific encoders to embed each attribute value. The embeddings of the subject entity, the relation, and the object value are then used to score the *truth* value of the triple by the *Scorer*, using the DistMult operation.

embeddings are capable of predicting the object entities for different types of data which is based on the similarity between those entities.

2 Multimodal Relational Embeddings

Knowledge bases (KB) often contain different types of information about entities including links, textual descriptions, categorical attributes, numerical values, and images. In this section, we briefly introduce the existing approaches to the embedded relational modeling that focus on modeling of the linked data using dense vectors. We then describe our model that extends these approaches to the multimodal setting, i.e., modeling the KB using all the different information.

Problem Setup: The goal of the relational modeling is to train a machine learning model that can score the *truth* value of any factual statement, represented here as a triplet of subject, relation and object, (s, r, o), where $s, o \in \xi$, a set of entities, and $r \in \Re$, a set of relations. Accordingly, the link prediction problem can be defined as learning a scoring function $\psi : \xi \times \Re \times \xi \rightarrow \mathbb{R}$ (or sometimes, $[0, 1]$). In order to learn the parameters of such a model, training data consists of the observed facts for the KB, i.e., a set of triples, which may be incomplete and noisy. In the last few years, the methods that have achieved impressive success on this task consist of models that learn fixed-length vectors, matrices, or tensors for each entity in ξ and relation in $\Re$, with the scoring function consisting of varying operators applied to these learned representations (described later in Section 4).

DistMult for Link Prediction: Although our proposed framework can be used with many of the existing relational models, here we focus on the DistMult approach [32] because of its simplicity, popularity, and high accuracy. In DistMult, each entity i is mapped to a d-dimensional dense vector ($\mathbf{e}_i \in \mathbb{R}^{d\times 1}$) and each relation r to a diagonal matrix $\mathbf{R}_r \in \mathbb{R}^{d\times d}$, and consequently, the score for any triple (s, r, o) is computed as: $\psi(s, r, o) = \mathbf{e}_s^T \mathbf{R}_r \mathbf{e}_o$. Since we cannot guarantee that the unobserved triples are *true negatives*, we use a pairwise ranking loss that tries to score existing (positive) triples higher than non-existing triples (negatively sampled), as:

$$\min_{\Theta} \sum_{i\in D_+} \sum_{j\in D_-} \max(0, \gamma + \phi_j - \phi_i) \tag{1}$$

where D_+, D_- denote the set of existing and non-existing (sampled) triples, γ is the width of margin, ϕ_i is the score of the i^{th} triple and Θ is the set of all embeddings. Following Bordes et al. [2], we generate negative samples of training triplets by replacing either subject or object entity with a random entity. DistMult thus learns entity and relation representations that encode the knowledge base, and can be used for completion, queries, or cleaning.

Multimodal Value Embeddings: Existing approaches to this problem assume that the subjects and the objects are from a fixed set of entities ξ, and thus are treated as indices into that set. However, in the most of the real-world KBs, the objects of triples (s, r, o) are not restricted to be in some indexed set, and instead, can be of any data type such as numerical, categorical, image, and text. In order to incorporate such *objects* into the existing relational models like DistMult, we propose to learn embeddings for any of these types of data. We utilize recent advances in deep learning to construct *encoders* for these objects to represent them, essentially providing an embedding $\mathbf{e}_o$ for any object value.

The overall goal remains the same: the model needs to utilize all the observed subjects, objects, and relations, across different data types, in order to estimate whether any fact (s, r, o) holds. We present an example of an instantiation of our model for a knowledge base containing movie details in Figure 1. For any triple (s, r, o), we embed the subject (movie) and the relation (such as `title`, `release year`, or `poster`) using a direct lookup. For the object, depending on the domain (indexed, string, numerical, or image, respectively), we use an appropriate encoder to compute its embedding $\mathbf{e}_o$. We use appropriate encoders for each data type, such as CNNs for images and LSTMs for text. As in DistMult, these embeddings are used to compute the score of the triple. Training such a model remains identical to DistMult, except that for negative sampling, here we replace the object entity with a random entity from the same domain as the object (either image, text, numerical or etc.).

Encoding Multimodal Data: Here we describe the encoders we use for multimodal objects.

Structured knowledge: Consider a triplet of information in the form of (s, r, o). To represent the subject entity s and the relation r as independent embedding vectors (as in previous work), we pass their one-hot encoding through a dense layer. Furthermore, for the case that the object entity is categorical, we embed it through a dense layer with a recently introduced selu activation [13], with the same number of nodes as the embedding space dimension.

Numerical: Objects in the form of real numbers can provide a useful source of information and are often quite readily available. We use a feed forward layer, after applying basic normalization, in order to embed the numbers into the embedding space. Note that we are projecting them to a higher-dimensional space, from $\mathbb{R} \rightarrow \mathbb{R}^d$. It is worth contrasting this approach to the existing methods that often treat numbers as distinct *entities*, i.e., learning independent vectors for numbers `39` and `40`, relying only on data to learn that these values are similar to each other.

Text: Since text can be used to store a wide variety of different types of information, for example names versus paragraph-long descriptions, we create different encoders depending on the lengths of the strings involved. For attributes that are fairly short, such as names and titles, we use character-based stacked, bidirectional LSTM to encode these strings, similar to Verga et al. [28], using the final output of the top layer as the representation of the string. For strings that are much longer, such as detailed descriptions of entities consisting of multiple sentences, we treat them as a sequence of words, and use a CNN over the word embeddings, similar to Francis-Landau et al. [5], in order to embed such values. These two encoders provide a fixed length encoding that has been shown for multiple tasks to be an accurate semantic representation of the strings [4].

Images: Images can also provide useful evidence for modeling entities. For example, we can extract person's details such as gender, age, job, etc., from image of the person [15], or location information such as its approximate coordinates, neighboring locations, and size from map images [30]. A variety of models have been used to compactly represent the semantic information in the images, and have been successfully applied to tasks such as image classification, captioning [11], and question-answering [33]. To embed images such that the encoding represents such semantic information, we use the last hidden layer of VGG pretrained network on Imagenet [21], followed by compact bilinear pooling [6], to obtain the embedding of the images.

3 Experiments

In this section we first demonstrate the capability of our model in comparison to DistMult method through a variety of link prediction tasks on our datasets, and present the analysis of the relations that most benefite from each type of data. Then we present some examples of recovery of missing multimodal information to demonstrate the kinds of dependencies the model is capturing.

Evaluation Benchmarks[1]: To evaluate the performance of our mutimodal relational embeddings approach, we provide two new benchmarks by extending existing datasets. The first one is built by

[1]The datasets available at: `https://github.com/pouyapez/multim-kb-embeddings`

#Relations	13
#Users	943
#Movies	1682
#Posters	1651*
#Ratings (train)	80,000
#Ratings (test)	20,000

(a) MovieLens-Plus

#Relations	45
#Total Entities	123,182
#Subjects	112,981
#Link Triples	1,079,040
#Numerical Attributes	111,406*
#Descriptions	107,326*
#Image Attributes	61,246*

(b) Yago10-Plus

Figure 2: Statistics of the two benchmark datasets we are using. The numbers in * are our contributions to the datasets, i.e. the reason for the `Plus` suffix.

adding posters to movie recommendation dataset, *MovieLens 100k*, and the second one by adding image and textual information for YAGO-10 dataset from *DBpedia* and numerical information from YAGO-3 database. We will release the datasets publicly for future research on multimodal relation modeling. Tables 2a and 2b provide the statistics of these datasets[1].

MovieLens-100k-Plus: We start with the *MovieLens-100k* dataset [2] [10], a popular benchmark for recommendation system for predicting user ratings with contextual features, containing $100,000$ ratings from around 1000 users on 1700 movies. MovieLens already contains rich relational data about occupation, gender, zip code, and age for users and genre, release date, and the titles for movies. We consider the genre attribute for each movie as a binary vector with length 19 (number of different genres provided by MovieLens). We use this representation because each movie genre is a combination of multiple, related categories. Moreover, we collect the movie posters for each movie from TMDB[3]. We treat the 5-point ratings as five different relations in KB triple format, i.e., $(\text{user}, r = 5, \text{movie})$, and evaluate the rating predictions as data for other relations is introduced into to the model. We use 10% of rating samples as the validation data.

YAGO-10-Plus: Even though MovieLens has a variety of data types, it is still quite small, and is over a specialized domain. We also consider a second dataset that is much more appropriate for knowledge graph completion and is popular for link prediction, the YAGO3-10 knowledge graph [23, 18]. This graph consists of around 120,000 entities, such as people, locations, and organizations, and 37 relations, such as kinship, employment, and residency, and thus much closer to the traditional information extraction goals. We extend this dataset with the textual description (as an additional relation) and the images associated with each entity (we have collected images for half of the entities), provided by *DBpedia*[4] [14]. We also identify few more additional relations such as `wasBornOnDate`, `happenedOnDate`, etc, that have dates as values.

Experimental setup: To facilitate a fair comparison we implement all methods using the identical loss and optimization for training, i.e., AdaGrad and the ranking loss. We tune all the hyperparameters on the validation data and use grid search to find the best hyperparameters. For evaluation we use two metrics of MRR and Hits@ which are commonly used in previous works.

Link Prediction: In this section, we evaluate the capability of our model in the link prediction task. The goal is to calculate MRR and Hits@ metric (ranking evaluations) of recovering the missing entities from triples in the test dataset, performed by ranking all the entities and computing the rank of the correct entity. Figure 3a shows the link (rating) prediction evaluation on movielens dataset when test data is consisting only of rating triples. As it shown, the model *R+M+U+T* outperform other methods with a considerable gap, which shows the importance of facilitating extra information in our advantages. On the other hand, Furthermore, Hits@1 for our baseline model is *40%*, which matches existing recommendation systems [8]. Based on results it seems that adding titles information has a higher impact compared to the poster information.

The result of link prediction on YAGO is provided in Figure 3b. We see that the model that encodes all type of information consistently performs better than other models, indicating that the model is effective in utilizing the extra information. On the other hand, the model that uses only text performs the second best, suggesting the entity descriptions contain more information than others. It is notable

[1] our contributions to the datasets

[2] https://grouplens.org/datasets/MovieLens/100k/

[3] https://www.themoviedb.org/

[4] http://wiki.dbpedia.org/

Models	Predicting Rating		
	MRR	Hits@1	RMSE
R (DistMult)	0.62	0.40	1.48
R+M	0.63	0.421	1.63
R+U	0.643	0.41	1.73
R+M+U	0.646	0.423	1.37
R+M+U+T	0.650	**0.424**	**1.23**
R+M+U+P	**0.652**	0.413	1.27
R+M+U+T+P	0.644	0.42	1.3

(a) **Rating prediction for MovieLens.** The model using rating is labeled R, movie-attributes as M, user-attributes as U, titles as T, and posters as P.

Models	MRR	Hits@1	Hits@3
S (DistMult)	0.326	0.221	0.375
S+D	0.36	0.262	0.395
S+N	0.325	0.213	0.382
S+I	0.342	0.235	0.352
S+D+N	0.359	0.243	0.401
S+D+I	0.351	0.239	0.371
S+N+I	0.362	0.259	0.402
S+D+N+1	**0.372**	**0.268**	**0.418**

(b) **Link Prediction for YAGO dataset.** The model using structured relations is labeled S, description as D, numerical information as N, and images as I.

Figure 3: Accuracy of Multimodal Embeddings

Table 1: **Per-Relation Breakdown** of how much each type of data is helping for YAGO.

	Links Only		+Numbers		+Description		+Images	
Relation	MRR	Hits@1	MRR	Hits@1	MRR	Hits@1	MRR	Hits@1
isAffiliatedTo	0.364	0.259	0.370	0.271	**0.392**	**0.301**	0.368	0.254
playsFor	0.371	0.261	**0.391**	0.291	0.389	**0.296**	0.381	0.275
isLocatedIn	0.341	0.223	0.352	0.249	**0.401**	**0.317**	0.369	0.265
hasGender	0.7894	0.602	0.771	0.582	0.796	**0.627**	**0.806**	0.613
wasBornIn	0.361	0.241	0.372	0.261	**0.408**	**0.326**	0.381	0.304

that model S outperformed by all other models, demonstrate the importance of using variations of information to attain higher accuracy.

Relation Breakdown: We perform additional analysis on the YAGO dataset to gain a deeper understanding of the performance of our model. Table 1 compares our models on the top five most frequent relations. As shown, the model that includes textual description significantly benefits `isAffiliatedTo, isLocatedIn` and `wasBornIn` relations, as this information often appears in text. Moreover, images are useful to detect genders (`hasGender`), while for the relation `playsFor`, numerical (dates) are more effective than images.

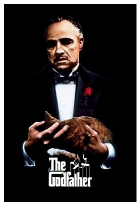
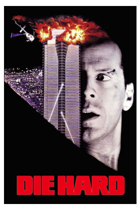
Querying Multimodal Attributes: Although we only encode multimodal data, and cannot *decode* in this setting directly, we provide examples in which we query for a multimodal attribute (like the poster), and rank all existing values (other posters) to observe which ones get ranked the highest. In other words, we are asking the model, if the actual poster is not available, which of the existing posters would the model recommend as a replacement (and same for title and genre). In Table 2 we show top-3 predicted values. We can see that the selected posters have visual similarity to the original poster in regards to the background, and placement of a face and the movie title in the poster. Along the same line, genres, though not exact, are quite similar as well (at least one of original genres appear in the predicted ones). And finally, the selected titles are also somewhat similar in meaning, and in structure. For example, two of the predicted titles for "Die Hard" have something to do with dying and being buried. Furthermore, both "The Godfather" and its first predicted title "101 dalmatians" consist of a three-character word followed by a longer word. We leave extensions that directly perform such decoding to future work.

4 Related Work

There is a rich literature on modeling knowledge bases using low-dimensional representations, differing in the operator used to score the triple. Whether they use matrix and tensor multiplication [17, 32, 22], euclidean distance [2, 29, 16], circular correlation [19], or the Hermitian dot product [26] as scoring function, the encoding component uses only the structured information for entities. We are the first to use different types of information such as text, numerical values and images in the encoding component while treating them as separate triplets of information.

A number of methods utilize an extra type of information as observed features for entities, either merging, concatenating, or averaging the entity and its features to compute its embeddings, such as

Table 2: **Querying Multimodal Values:** We find the highest scoring values, according to our trained model, for each attribute of a movie, and compare it to the true value.

True Value	Top-3 Predicted Values
“The Godfather”	“101 Dalmatians”, “Love and Death on Long Island”, “First Knight”
	
“Action, Crime, Drama”	“Drama, Romance, War, Western”, “Drama, Romance, War”, “Drama, War”
“Die Hard”	“The Band Wagon”, “Underground”, “Roseanna’s Grave”
	
“Action, Thriller”	“Drama, War”, “Action, Drama, War”, “Comedy, Drama, War”

numerical values [7], images [31], and text [24, 25, 27]. In addition to treating extra information as features, graph embedding approaches [3, 20, 12] consider fixed number of attributes as a part of encoding component to achieve more accurate embedding. The difference between our model and these mentioned approaches is three-fold: we are the first to use all varieties of information, we treat these different type of information (numerical, text, image) as relational triples of structured knowledge instead of predetermined features, and, our model represents uncertainty in them, supporting missing values and facilitating recovery of lost information, which is not possible with previous approaches.

5 Conclusions

Motivated by the need to utilize multiple source of information to achieve more accurate link prediction we presented a novel neural approach to multimodal relational learning. In this paper we introduced a universal link prediction model that uses different types of information to model knowledge bases. We proposed a compositional encoding component to learn unified entity embedding that encode the variety of information available for each entity. In our analysis we show that our model in comparison to a common link predictor, DistMult, can achieve higher accuracy, showing the importance of employing the available variety of information for each entity. Since all the existing datasets are designed for previous methods, they lack mentioned kind of extra information. In result, we introduced two new benchmarks YAGO-10-plus and MovieLens-100k-plus, that are extend version of existing datasets. Further, in our evaluation, we showed that our model effectively utilizes the extra information in order to benefit existing relations. We will release the datasets and the open-source implementation of our models publicly.

There are number of avenues for future work. We will investigate the performance of our model in completing link prediction task using different scoring function and more elaborate encoding component and objective function. We are also interested in modeling decoding of multimodal values in the model itself, to be able to query these values directly. Further, we plan to explore efficient query algorithms for embedded knowledge bases, to compete with practical database systems.

References

[1] Antoine Bordes, Jason Weston, Ronan Collobert, Yoshua Bengio, et al. Learning structured embeddings of knowledge bases. In *AAAI*, 2011.

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

[4] Cícero Nogueira Dos Santos and Maira Gatti. Deep convolutional neural networks for sentiment analysis of short texts. In *COLING*, pages 69–78, 2014.

[5] Matthew Francis-Landau, Greg Durrett, and Dan Klein. Capturing semantic similarity for entity linking with convolutional neural networks. *In NAACL*, 2016.

[6] Yang Gao, Oscar Beijbom, Ning Zhang, and Trevor Darrell. Compact bilinear pooling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 317–326, 2016.

[7] Alberto Garcia-Duran and Mathias Niepert. Kblrn: End-to-end learning of knowledge base representations with latent, relational, and numerical features. *arXiv preprint arXiv:1709.04676*, 2017.

[8] Roger Guimerà, Alejandro Llorente, Esteban Moro, and Marta Sales-Pardo. Predicting human preferences using the block structure of complex social networks. *PloS one*, 7(9):e44620, 2012.

[9] Nitish Gupta and Sameer Singh. Collectively embedding multi-relational data for predicting user preferences. *CoRR*, 2015.

[10] F Maxwell Harper and Joseph A Konstan. The movielens datasets: History and context. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 5(4):19, 2016.

[11] Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3128–3137, 2015.

[12] Thomas N Kipf and Max Welling. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308*, 2016.

[13] Günter Klambauer, Thomas Unterthiner, Andreas Mayr, and Sepp Hochreiter. Self-normalizing neural networks. *arXiv preprint arXiv:1706.02515*, 2017.

[14] Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick Van Kleef, Sören Auer, et al. Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web*, 6(2):167–195, 2015.

[15] Gil Levi and Tal Hassner. Age and gender classification using convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 34–42, 2015.

[16] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. Learning entity and relation embeddings for knowledge graph completion. In *AAAI*, pages 2181–2187, 2015.

[17] Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. 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, 2011.

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

[19] Maximilian Nickel, Lorenzo Rosasco, Tomaso A Poggio, et al. Holographic embeddings of knowledge graphs. In *AAAI*, pages 1955–1961, 2016.

[20] Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. *arXiv preprint arXiv:1703.06103*, 2017.

[21] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *In ICLR*, 2015.

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

[23] Fabian M Suchanek, Gjergji Kasneci, and Gerhard Weikum. Yago: a core of semantic knowledge. In *Proceedings of the 16th international conference on World Wide Web*, pages 697–706. ACM, 2007.

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

[25] Kristina Toutanova, Victoria Lin, Wen-tau Yih, Hoifung Poon, and Chris Quirk. Compositional learning of embeddings for relation paths in knowledge base and text. In *ACL (1)*, 2016.

[26] Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. Complex embeddings for simple link prediction. In *International Conference on Machine Learning*, pages 2071–2080, 2016.

[27] Cunchao Tu, Han Liu, Zhiyuan Liu, and Maosong Sun. Cane: Context-aware network embedding for relation modeling. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1722–1731, 2017.

[28] Patrick Verga, David Belanger, Emma Strubell, Benjamin Roth, and Andrew McCallum. Multilingual relation extraction using compositional universal schema. *Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2016.

[29] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. Knowledge graph embedding by translating on hyperplanes. In *AAAI*, pages 1112–1119, 2014.

[30] Tobias Weyand, Ilya Kostrikov, and James Philbin. Planet-photo geolocation with convolutional neural networks. In *European Conference on Computer Vision*, pages 37–55. Springer, 2016.

[31] Ruobing Xie, Zhiyuan Liu, Tat-seng Chua, Huanbo Luan, and Maosong Sun. Image-embodied knowledge representation learning. *arXiv preprint arXiv:1609.07028*, 2016.

[32] Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. Embedding entities and relations for learning and inference in knowledge bases. *In ICLR*, 2015.

[33] Zichao Yang, Xiaodong He, Jianfeng Gao, Li Deng, and Alex Smola. Stacked attention networks for image question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 21–29, 2016.