Learning Physical Common Sense as Knowledge Graph Completion via BERT Data Augmentation and Constrained Tucker Factorization

Zhenjie Zhao[1] **Evangelos E. Papalexakis**[2] **Xiaojuan Ma**[1]
[1] Department of Computer Science & Engineering, HKUST
[2] Department of Computer Science & Engineering, University of California, Riverside
zzhaoao@connect.ust.hk, epapalex@cs.ucr.edu, mxj@cse.ust.hk

Abstract

Physical common sense plays an essential role in the cognition abilities of robots for human-robot interaction. Machine learning methods have shown promising results on physical commonsense learning in natural language processing but still suffer from model generalization. In this paper, we formulate physical commonsense learning as a knowledge graph completion problem to better use the latent relationships among training samples. Compared with completing general knowledge graphs, completing a physical commonsense knowledge graph has three unique characteristics: training data are scarce, not all facts can be mined from existing texts, and the number of relationships is small. To deal with these problems, we first use a pre-training language model BERT to augment training data, and then employ constrained tucker factorization to model complex relationships by constraining types and adding negative relationships. We compare our method with existing state-of-the-art knowledge graph embedding methods and show its superior performance.

1 Introduction

Physical common sense means understanding the physical properties of objects and how they can be manipulated (Forbes et al., 2019). Empowering natural language processing (NLP) methods with physical common sense is important when dealing with tasks that are related to the physical world, such as physical commonsense reasoning (Bisk et al., 2020), grounded verb semantics (She and Chai, 2017), and the more general human-robot interaction problem.

Generally, there are currently three methods of learning physical common sense: manual annotation, text mining, and machine learning. Manual annotation is difficult for human annotators due to inconsistent perceptions and the challenge of enumerating all physical facts. Mining text data is also challenging because some physical facts are not written in texts explicitly. Machine learning is a promising method to discover new physical facts using existing data. Forbes et al. (2019) formulate physical commonsense learning as three separate machine learning tasks: 1) given an object and a property, predicting whether they follow an object-property (OP) relationship, *e.g., an apple is edible*; 2) given an object and an affordance, predicting whether they follow an object-affordance (OA) relationship, *e.g., he drove the car*; and 3) given an affordance and a property, predicting whether they follow an affordance-property (AP) relationship, *e.g., if you can eat something, then it is edible*. However, it is difficult for a machine learning model to generalize through the use of the latent relationships among samples. For example, even if we have a training sample *an apple is edible*, it is hard to say that the trained model can generalize to predict a testing sample *an apple is red* correctly.

In this paper, we propose to model physical commonsense learning as a knowledge graph completion problem to better use the latent relationships among samples. An knowledge graph can be represented as a 3-way binary tensor, and each entry is in triple form (e_h, r, e_t) (Nickel et al., 2016; Wang et al., 2017), where e_h denotes the head entity, e_t denotes the tail entity, r denotes the relationship between e_h and e_t, $(e_h, r, e_t) = 1$ denotes the fact is true in the training data, and $(e_h, r, e_t) = 0$ denotes the fact does not exist or is false in the training data. The goal of knowledge graph completion is to predict the real value of (e_h, r, e_t) when it is missing or its label is wrong in the training data. In terms of physical common sense, entities come from the set of all objects, properties, and affordances, and relationships come from the set of OP, OA, and AP.

Compared with general knowledge graphs such

Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, pages 3293–3298,
November 16–20, 2020.

as DBpedia (Auer et al., 2007) and Freebase (Bollacker et al., 2008), a physical commonsense knowledge graph has at least three characteristics: 1) Training facts are scarce. For example, when labeling the properties of an object, people usually name the ones that are easiest to think of but cannot enumerate all properties. 2) Not all facts can be mined from existing texts. For example, the relationships between affordances and properties usually do not appear in texts explicitly and need to be reasoned. 3) The number of relationships is small and all are n-to-n relationships, which makes modeling relationships between entities more complicated.

Forbes et al. (2019) show that with supervised fine-tuning, Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2019) can learn the relationships OP and OA well but not AP. In this paper, we first use BERT to augment training data of OP and OA and then employ constrained Tucker factorization (Balazevic et al., 2019) to complete the knowledge graph of physical common sense. More specially, we use typed constraints to reduce the solution space and add negative relationships to leverage negative training samples. We evaluate this method on triple classification and link prediction tasks using a physical commonsense dataset (Forbes et al., 2019), and show that it can model physical common sense more effectively compared with state-of-the-art knowledge graph embedding methods.

The contributions of this paper are: 1) we formulate physical commonsense learning as a knowledge graph completion problem, 2) we propose a novel pipeline that combines pre-training models and knowledge graph embedding to learn physical common sense, and experiment results show its superior performance.

2 Related Work

2.1 Common Sense and Physical Common Sense

Common sense learning is one of the main challenges in NLP (Cambria and White, 2014). Although existing works have made significant progress on reading comprehension and question answering (Rajpurkar et al., 2016), they are still text-based and challenging to use for commonsense reasoning (Ostermann et al., 2018). In general, commonsense modeling can be classified into two categories: 1) explicitly encoding via knowledge graphs (Auer et al., 2007; Bollacker et al., 2008) and 2) implicitly encoding via language models (Bosselut et al., 2019). Building high-quality knowledge graphs usually requires expensive human annotation. There is some research on extracting facts from unstructured text (Clancy et al., 2019), but it is not flexible to build domain-specific knowledge graphs. Recent research works show that pre-training models can be good at encoding commonsense knowledge due to a large number of model parameters and text corpora, and they can be used to complete knowledge graphs (Bosselut et al., 2019).

Physical commonsense learning is a recently-proposed task (Forbes et al., 2019) that is related to language understanding with a physical world context, which is a sub-category of commonsense learning. Forbes et al. (2019) formulate physical commonsense learning as a machine learning problem, and show that a pre-training BERT model can learn the OP and OA tasks well but cannot generalize well on the AP task. In this paper, to deal with the generalization problem of BERT, we explore using knowledge graph embedding that is commonly used in commonsense modeling to deal with the issue of physical commonsense learning.

2.2 Knowledge Graph Embedding

Knowledge graphs have been shown to be useful for many NLP tasks, such as contextual word embedding (Peters et al., 2019), text classification (K M et al., 2018), and language generation (Zhou et al., 2018). In general, knowledge graph embedding can be classified into two categories: translational distance models and semantic matching models (Wang et al., 2017). Translational distance models model the score function of a factual triple (e_h, r, e_t) as the distance between e_h and e_t through the relationship r. Typical methods include TransE (Bordes et al., 2013) and its variants, such as TransD (Ji et al., 2015). Semantic matching models model the score function of a factual triple by exploiting the latent semantics between e_h and e_t, and they are usually modeled as a 3-way tensor. Typical methods include RESCAL (Nickel et al., 2011), DistMult (Yang et al., 2015), ComplEx (Trouillon et al., 2016), SimplE (Kazemi and Poole, 2018), and Tucker factorization (Balazevic et al., 2019). Compared with other methods, the Tucker factorization method learns a basis of relationship embeddings and can model more complex relationships, so it is used in this paper.

3 Method

Our method consists of two components: 1) we first augment all pairs of OP and OA tasks using BERT; 2) with the training data of OP, OA, and AP as input, we use constrained Tucker factorization to de-noise and complete the knowledge graph. In particular, we use typed constraints to reduce the solution space and add negative relationships to leverage negative training samples.

3.1 Data Augmentation

Because BERT can only do well on the OP and OA tasks (Forbes et al., 2019), we only augment data of these two tasks. In particular, for each pair (o, p) of OP, where $o \in O$ is an instance of objects and $p \in P$ is an instance of properties, we compose a sentence: "A/An o is p.", and use fine-tuned BERT on OP to predict its label l_{op}. Similarly, for each pair (o, a) of OA, where $o \in O$ is an instance of objects and $a \in A$ is an instance of affordances, we compose a sentence: "He a the o.", and use fine-tuned BERT on OA to predict its label l_{oa}. We use the augmented data $\mathcal{D}_{OP}$, $\mathcal{D}_{OA}$, together with the original AP data $\mathcal{D}_{AP}$ as input to the constrained Tucker factorization model.

3.2 Constrained Tucker Factorization

All (e_h, r, e_t) tuples compose a 3-way binary tensor $\mathcal{X} \in \{0, 1\}^{n_e \times n_e \times n_r}$, where each entry $\mathcal{X}(i, j, k)$ denotes whether the i-th head entity and j-th tail entity follow the k-th relationship, n_e is the number of entities, and n_r is the number of relationships. Each slice of $\mathcal{X}$ is a $n_e \times n_e$ matrix of the relationship k. The Tucker factorization model proposed by Balazevic et al. (2019) approximates $\mathcal{X}$ as:

$$\hat{\mathcal{X}} = W \times_1 E \times_2 E \times_3 R, \tag{1}$$

where $\times_i$ denotes the i-mode product, $E \in \mathbb{R}^{d_e \times n_e}$ is entity embeddings, $R \in \mathbb{R}^{d_r \times n_r}$ is relation embeddings, $W \in \mathbb{R}^{d_e \times d_e \times d_r}$ is a core tensor, d_e is the latent dimension of entities, and d_r is the latent dimension of relationships.

3.2.1 Typed Constraints

Similar to the typed tensor decomposition method in (Chang et al., 2014), because we know that only objects and properties can potentially have the relationship OP, we can constrain the remaining entries of the OP matrix as 0. We can also constrain the OA and AP relationships in a similar way. Therefore, we optimize the following objective jointly for the three tasks:

$$\min_{E,R,W} ||\mathcal{X} - \hat{\mathcal{X}}||_F^2 + \lambda ||\hat{\mathcal{X}} \odot \mathcal{M}||_F^2 + \beta f(\hat{\mathcal{X}}), \tag{2}$$

where $||\cdot||_F$ denotes the Frobenius norm, $\odot$ denotes element-wise production, $\mathcal{M}$ is the mask tensor for the typed constraint, and $f(\hat{\mathcal{X}}) = ||E||_F^2 + ||R||_F^2 + ||W||_F^2$ is the regularization term. λ and β are coefficient weights of constraints. Because all entities are categorized and we consider the type constraint, there is only one possible relationship for a single head and tail.

3.2.2 Negative Samples

One unique challenge of a physical commonsense knowledge graph is that we have to use the open-world assumption. Namely, for unknown facts, we cannot assume that they are negative samples. In this paper, we propose encoding negative samples by adding corresponding negative relationships explicitly. For each OP, OA and AP relationship, we add a corresponding negative relationship, *i.e.*, NOT-OP, NOT-OA and NOT-AP. For example, *(person, NOT-OP, a_tool)*, *(cup, NOT-OA, twist)*, *(walk, NOT-AP, used_for_eating)*. Similar to (Balazevic et al., 2019), we also use reverse relationships. Namely, for each tuple *(h, r, t)*, we add *(t, r-reverse, h)*. Therefore, there are six negative relationships in total. For the OP and OA tasks, the negative samples are added through the data augmentation module in subsection 3.1, *i.e.*, the labels are predicted by BERT, and for the AP task, we use the negative samples from the dataset. In this way, we can not only increase the number of relationships but also leverage labeled negative samples more effectively.

4 Experiments

To evaluate the method, we conducted experiments with a physical commonsense dataset (Forbes et al., 2019) on triple classification and link prediction. To simplify the problem, we only used the situated OP, OA, and AP data, which contains 80 objects, 50 properties, and 504 affordances. The statistics are shown in Table 2. With the data augmentation component, we generated 4000 OP samples and 40320 OA samples. We compared the method with state-of-the-art knowledge graph embedding methods, including TransE, TransD, RESCAL, Dist-

	OP				OA				AP			
Method	acc	$F1_{obj}$	$F1_{prop}$	$\mu F1$	acc	$F1_{obj}$	$F1_{aff}$	$\mu F1$	acc	$F1_{aff}$	$F1_{prop}$	$\mu F1$
TransE	0.71	0.14	0.18	0.13	0.50	0.47	0.54	0.43	0.62	0.24	0.29	0.21
TransD	0.65	0.25	0.30	0.20	0.56	0.52	0.44	0.51	0.58	0.23	0.24	0.21
RESCAL	0.60	0.21	0.21	0.20	0.47	0.37	0.46	0.35	0.63	0.22	0.24	0.22
DistMult	0.62	0.24	0.22	0.22	0.52	0.45	0.47	0.45	0.64	0.21	0.23	0.21
ComplEx	0.63	0.23	0.22	0.20	0.52	0.40	0.48	0.45	0.62	0.21	0.24	0.20
SimplE	0.61	0.18	0.21	0.17	0.51	0.38	0.49	0.42	0.61	0.23	0.24	0.21
Tucker	0.77	0.21	0.14	0.17	0.62	0.54	0.45	0.55	0.18	0.28	0.30	0.26
Ours (w/o DA)	0.77	0.12	0.12	0.08	0.50	0.11	0.07	0.09	0.80	**0.44**	0.41	0.47
Ours (w/o CSTR)	0.17	0.29	0.30	0.26	0.55	0.69	**0.82**	0.69	0.18	0.27	0.30	0.26
Ours	**0.91**	**0.61**	**0.48**	**0.62**	**0.85**	**0.83**	0.67	**0.84**	**0.81**	0.43	**0.42**	**0.47**

Table 1: Experimental results of triple classification, including macro $F1$ scores per category, *i.e.*, object (obj), property ($prop$), affordance (aff), and micro $F1$ score ($\mu F1$).

	training			testing		
	positive	negative	total	positive	negative	total
OP	6188	34712	40900	1654	9446	11100
OA	2454	2454	4908	666	666	1332
AP	18564	104136	122700	4962	28338	33300

Table 2: The statistics of the physical commonsense dataset from Forbes et al. (2019).

Method	MRR	Hits@10	Hits@3	Hits@1
TransE	0.629	0.691	0.636	0.596
TransD	0.631	0.701	0.637	0.596
RESCAL	0.019	0.036	0.017	0.006
DistMult	0.598	0.619	0.605	0.582
ComplEx	0.605	0.615	0.603	0.597
SimpleIE	0.603	0.619	0.606	0.591
Tucker	0.650	0.723	0.648	0.620
Ours (w/o DA)	0.733	0.815	0.764	0.687
Ours (w/o CSTR)	0.514	0.727	0.549	0.416
Ours	**0.826**	**0.931**	**0.863**	**0.768**

Table 3: Link prediction results, where MRR denotes Mean Reciprocal Rank.

Mult, ComplEx, SimplE, and Tucker [1]. We optimized equation 2 with Adam in PyTorch and did not optimize the regularization explicitly. λ was set to 0.1 through a 5-fold cross validation. d_e and d_r were set to 200 by default.

4.1 Triple Classification

Triple classification needs to predict whether a fact (e_h, r, e_t) is correct or not. With the learned E, R, and W, we calculated the probability that two entities e_h, e_t follow a relationship r as:

$$\sigma(W \times_1 e_h \times_2 e_t \times_3 r), \tag{3}$$

where σ is the sigmoid function. With the typed constraint, we then selected the relationship with the maximal probability. The results are shown in Table 1. For other methods, we only input the original training data without data augmentation.

With the data augmentation (**DA**) and typed constraints (**CSTR**), we achieved the best classification accuracy. In particular, we achieved relatively high micro and macro F1 scores for the three tasks, indicating that our method can predict positive samples more accurately.

4.2 Link Prediction

Link prediction predicts the tail entity with one head and one relationship, *i.e.*, $(e_h, r, ?)$. With the learned E, R, and W, we calculated probabilities of all candidate entities as:

$$\sigma(W \times_1 e_h \times_3 r). \tag{4}$$

Similarly, we compared our results with typical knowledge graph embedding methods. For the Tucker method, we trained 2000 epochs, and for our method, we trained 50 epochs. The results are shown in Table 3. Compared with other methods, our method usually had relatively higher performance, indicating its potential in discovering new physical commonsense facts.

4.3 Discussion

To evaluate the effectiveness of the data augmentation (**DA**) and typed constraint (**CSTR**) components, we also conducted ablation studies on triple classification and link prediction separately, and the results are shown in Tables 1 and 3, from which we can see that **DA** and **CSTR** can help improve the performance of Tucker factorization.

Compared with knowledge graph embedding methods, the pre-training BERT model can perform better on OP and OA, but it is more difficult to generalize well on AP because such facts are not written in existing texts explicitly and BERT does not encode them as well as the OP and OA tasks (Forbes et al., 2019). For example, in terms of AP triple classification, the results of BERT are: a mi-

[1]The implementations of TransE, TransD, RESCAL, DistMult, ComplEx and SimplE are from OpenKE (Han et al., 2018). The implementation of Tucker is from Balazevic et al. (2019). Without any explicit statement, we used their default parameters.

cro F1 score of 0.37, an affordance macro F1 score of 0.36, and a property macro F1 score of 0.25. Our results for triple classification outperform them by a large margin, although our results are still worse in terms of OP and OA classification.

From the perspective of multi-task learning, one explanation of the improvement on the AP task is that the core tensor W can be viewed as parameter sharing among the three tasks and through the parameter sharing, the OP and OA tasks help improve the performance of AP. In a separate experiment, we used a multi-task BERT model (Stickland and Murray, 2019), and got a micro F1 score of 0.46, an affordance macro F1 score of 0.37, and a property macro F1 score of 0.48 for the AP task, which was similar to the result with our model.

5 Conclusion

In this paper, we formulate physical commonsense learning as a knowledge graph completion problem. We first use BERT to augment training data of OP and OA, and then employ constrained Tucker factorization to complete the knowledge graph. We constrain types to reduce the solution space and add negative relationships to leverage negative training samples. Compared with typical knowledge graph embedding methods, our results show good performance on triple classification and link prediction. Our method also has the potential to be a generic approach to benefit performance on the knowledge graph completion problem.

Acknowledgments

The authors thank Dr. Mingfei Sun for helpful discussions, as well as all anonymous reviewers for insightful comments. E. Papalexakis was supported by a UCR-China collaboration grant by the Bourns College of Engineering at UCR, and by the National Science Foundation CDSE Grant no. OAC-1808591.

References

Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. DBpedia: A nucleus for a web of open data. In *Proceedings of the 6th International The Semantic Web and 2nd Asian Conference on Asian Semantic Web Conference*, page 722735, Berlin, Heidelberg. Springer-Verlag.

Ivana Balazevic, Carl Allen, and Timothy Hospedales. 2019. TuckER: Tensor factorization for knowledge graph completion. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5185–5194, Hong Kong, China. Association for Computational Linguistics.

Yonatan Bisk, Rowan Zellers, Ronan LeBras, Jianfeng Gao, and Yejin Choi. 2020. PIQA: reasoning about physical commonsense in natural language. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 7432–7439. AAAI Press.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: A collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, SIGMOD 08, page 12471250, New York, NY, USA. Association for Computing Machinery.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Durán, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS13, page 27872795, Red Hook, NY, USA. Curran Associates Inc.

Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Celikyilmaz, and Yejin Choi. 2019. COMET: Commonsense transformers for automatic knowledge graph construction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4762–4779, Florence, Italy. Association for Computational Linguistics.

Erik Cambria and Bebo White. 2014. Jumping NLP curves: A review of natural language processing research. *IEEE Computational intelligence magazine*, 9(2):48–57.

Kai-Wei Chang, Wen-tau Yih, Bishan Yang, and Christopher Meek. 2014. Typed tensor decomposition of knowledge bases for relation extraction. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1568–1579, Doha, Qatar. Association for Computational Linguistics.

Ryan Clancy, Ihab F. Ilyas, and Jimmy Lin. 2019. Scalable knowledge graph construction from text collections. In *Proceedings of the Second Workshop on Fact Extraction and VERification (FEVER)*, pages 39–46, Hong Kong, China. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language*

Technologies, Volume 1 (Long and Short Papers), pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Maxwell Forbes, Ari Holtzman, and Yejin Choi. 2019. Do neural language representations learn physical commonsense? *Proceedings of the 41st Annual Conference of the Cognitive Science Society*.

Xu Han, Shulin Cao, Xin Lv, Yankai Lin, Zhiyuan Liu, Maosong Sun, and Juanzi Li. 2018. OpenKE: An open toolkit for knowledge embedding. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 139–144, Brussels, Belgium. Association for Computational Linguistics.

Guoliang Ji, Shizhu He, Liheng Xu, Kang Liu, and Jun Zhao. 2015. Knowledge graph embedding via dynamic mapping matrix. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 687–696, Beijing, China. Association for Computational Linguistics.

Annervaz K M, Somnath Basu Roy Chowdhury, and Ambedkar Dukkipati. 2018. Learning beyond datasets: Knowledge graph augmented neural networks for natural language processing. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1*, pages 313–322, New Orleans, Louisiana. Association for Computational Linguistics.

Seyed Mehran Kazemi and David Poole. 2018. Simple embedding for link prediction in knowledge graphs. In *Advances in Neural Information Processing Systems*, pages 4284–4295. Curran Associates, Inc.

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

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 International Conference on Machine Learning*, ICML'11, page 809816, Madison, WI, USA. Omnipress.

Simon Ostermann, Michael Roth, Ashutosh Modi, Stefan Thater, and Manfred Pinkal. 2018. SemEval-2018 task 11: Machine comprehension using commonsense knowledge. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 747–757, New Orleans, Louisiana. Association for Computational Linguistics.

Matthew E. Peters, Mark Neumann, Robert Logan, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A. Smith. 2019. Knowledge enhanced contextual word representations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 43–54, Hong Kong, China. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Lanbo She and Joyce Chai. 2017. Interactive learning of grounded verb semantics towards human-robot communication. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1634–1644, Vancouver, Canada. Association for Computational Linguistics.

Asa Cooper Stickland and Iain Murray. 2019. BERT and PALs: Projected attention layers for efficient adaptation in multi-task learning. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 5986–5995, Long Beach, California, USA. PMLR.

Tho Trouillon, Johannes Welbl, Sebastian Riedel, Eric Gaussier, and Guillaume Bouchard. 2016. Complex embeddings for simple link prediction. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 2071–2080, New York, New York, USA. PMLR.

Quan Wang, Zhendong Mao, Bin Wang, and Li Guo. 2017. Knowledge graph embedding: A survey of approaches and applications. *IEEE Transactions on Knowledge and Data Engineering*, 29(12):2724–2743.

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

Hao Zhou, Tom Young, Minlie Huang, Haizhou Zhao, Jingfang Xu, and Xiaoyan Zhu. 2018. Commonsense knowledge aware conversation generation with graph attention. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 4623–4629. International Joint Conferences on Artificial Intelligence Organization.