



ResE: A Fast and Efficient Neural Network-Based Method for Link Prediction

Xuexiang Li, Hansheng Yang and Cong Yang *

School of Cyber Science and Engineering, Zhengzhou University, Zhengzhou 450001, China * Correspondence: wangyuanyc@zzu.edu.cn; Tel.: +86-1870-679-2446

Abstract: In this study, we present a novel embedding model, named ResE, for predicting links in knowledge graphs. ResE employs depth-separable convolution and residual blocks, integrated with channel attention mechanisms. ResE surpasses previously published models, including the closely related TransE model, by achieving the satisfactory mean rank (MR) and the excellent Hits@10 scores on both WN18RR and FB15K-237 benchmarks. ResE is a promising model for knowledge graph completion tasks, with potential for further investigation and extension to new applications such as user-oriented relationship modeling. Although comparatively shallow compared to computer vision convolutional architectures, future work may explore deeper convolutional models. ResE exhibits remarkable performance and outperforms existing approaches, thus setting a new benchmark for knowledge graph completion. The outcomes of our study illustrate the effectiveness of incorporating depth-separable convolution and residual blocks, accompanied by channel attention mechanisms, in modeling knowledge graphs. These findings highlight ResE's potential to push the boundaries of cutting-edge in this domain.

Keywords: link prediction; residual blocks; knowledge graph completion



Citation: Li, X.; Yang, H.; Yang, C. ResE: A Fast and Efficient Neural Network-Based Method for Link Prediction. *Electronics* **2023**, *12*, 1919. https://doi.org/10.3390/ electronics12081919

Academic Editor: George A. Tsihrintzis

Received: 10 March 2023 Revised: 12 April 2023 Accepted: 16 April 2023 Published: 19 April 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https:// creativecommons.org/licenses/by/ 4.0/).

1. Introduction

Knowledge graphs refer to graph-structured knowledge bases that represent factual information through relationships (edges) established between entities (nodes). These graph-based representations are widely used in various applications, including search, analytics, recommendation, and data integration. However, a common issue with knowledge graphs is incompleteness, which means that there are missing links in the graph.

Knowledge bases of significant scale, DBpedia [1] YAGO [2] and other related instances, consist of databases of triples that capture the relationships between entities in the form of (*Subject_entity*, *relation*, *Object_entity*) represented by (h, r, t). These knowledge bases are instrumental in several applications, such as semantic searching and ranking, question answering, and machine reading. Nonetheless, they are often incomplete, lacking many valid triples. As a solution to this issue, considerable research efforts have been invested in knowledge base completion or link prediction, aimed at forecasting the validity of a triple (h, r, t).

Link prediction aims to detect missing links in knowledge graphs, and it is crucial that a link predictor exhibits efficient scalability in terms of parameter count and computational cost to be applicable in practical settings. Convolution operator is a widely-used technique in computer vision, which meets these prerequisites due to its parametric efficiency, computational speed, and optimized GPU implementation. In contrast to fully connected neural networks, convolutional neural networks (CNNs) possess the ability to learn nonlinear features and capture intricate relationships using fewer parameters. Furthermore, CNNs have yielded robust techniques for training multi-layer convolutional networks, to counter overfitting, as proposed in 2015 [3,4], 2014 [5,6] and similar references.

This paper introduces ResE, a multi-layer CNN model that utilizes two-dimensional convolution for link prediction in KGs (Knowledge Graphs). ResE addresses the challenges

of overfitting, model complexity, and parameter redundancy by incorporating a residual network, depth-separable convolution, and channel attention mechanism. The architecture of ResE consists of several depth separable convolution blocks and a residual block, which harnesses the properties of convolution operators for effective link prediction.

The main contributions in our works are:

- ResE is a newly proposed entity and relation embedding model for knowledge graph completion that is based on deep separable convolutional neural networks. By leveraging the power of deep separable convolutional networks to capture relevant features, ResE effectively enhances the generalization of transition-based embedded models. This approach represents a significant contribution to the field, as it offers a novel solution for improving the performance of knowledge graph completion tasks.
- Our study involved an evaluation of ResE using two widely-accepted benchmark datasets, WN18RR [7] and FB15k-237 [8], to assess its effectiveness in link prediction. Our experimental results demonstrate that ResE outperforms previous embedding models, and it achieves the highest Mean Reciprocal Rank score in most cases when compared to several other state-of-the-art models on these datasets.

2. Related Works

To complete a knowledge graph, one important task is link prediction, which involves predicting the missing relationships between entities. This task is accomplished through learning a scoring function $\Psi(x) = \mathcal{E} \times \mathcal{R} \times \mathcal{E} \mapsto \mathbb{R}$ that set a loss to evaluate an input triple x = (h, r, t) representing an entity h with a relationship r to another entity t. The score reflects the likelihood of the fact represented by x being true. The approach is to learn to predict the likelihood of a given triple being valid, and then rank the possible triples to determine the most probable fact. This can be viewed as a pointwise learning-to-rank task, in which the goal is to predict the score for a single triple rather than comparing pairs of triples. The performance of the scoring function is evaluated using standard metrics such as Mean Reciprocal Rank (MRR) and Hits@N, where N is the number of possible triples to rank.

Several neural models have been proposed for link prediction in knowledge graphs, including TransE [9], which represents entities and relations as k-dimensional vector embeddings and uses a transitional characteristic to model relationships between entities. TransE has served as the basis for several other transition-based models, such as TransH [10], TransR [11], TransD [12], STransE [13], and TranSparse [14]. Another model, DistMult [15], generates the same number of relational parameters as TransE and uses a simplified bilinear formula that has the same scalability as TransE but outperforms it on the link prediction task. ComplEx [16] is an extension of DistMult that utilizes a complex vector space and the Hermitian dot product to represent relations, with the head and tail embeddings conjugated to capture both symmetric and antisymmetric relations. Finally, HolE [17] is a model that combines the expressiveness of tensor product with the efficiency and simplicity of TransE, utilizing the cyclic correlation of head and tail entity vectors to represent pairs of entities in the entire knowledge graph.

The proposed model offers a significant advantage over HolE's architecture since it incorporates multiple layers of non-linear features, thereby increasing its expressiveness. On the other hand, Graph Convolutional Networks (GCNs) [18] have also been investigated in relation to link prediction. GCNs use the convolution operator to exploit the locality information in graphs. However, the GCN framework is limited to undirected graphs, whereas knowledge graphs are inherently directed. Moreover, the memory requirements of GCNs may be prohibitively high.

Convolutional models have been proposed for a variety of natural language processing (NLP) tasks, including semantic parsing [19], sentence classification [20], search query retrieval [21], sentence modeling [22], and several other NLP applications. Originally designed for computer vision [23], convolutional neural networks (CNN) have recently gained significant interest in NLP [24] due to their ability to learn nonlinear features and

capture complex relationships while using fewer parameters compared to fully connected neural networks. They can also be used to learn deep expression features.

Dettmers et al. introduced ConvE [7] as the first neural link prediction method to employ a two-dimensional convolutional layer to model the relationship between input entities and relations. The model utilizes a convolutional layer in combination with a fully connected layer, enabling the representation of semantic information via multi-layer nonlinear feature learning. Additionally, ConvE has a high parameter utilization rate. The success of CNNs in computer vision inspired the development of ConvE for link prediction.

This paper presents a novel approach to knowledge graph link prediction that harnesses the power of neural networks. The method uses both structural and textual information in the knowledge graph to predict new relationships between entities. Specifically, it extracts features from the graph structure and text information of entities, and uses them to make predictions. To address parameters redundancy and overfitting, the current study proposes an innovative knowledge graph link prediction approach that leverages both depth-separable convolution and a channel attention mechanism. The utilization of these techniques facilitates the model to effectively capture intricate associations and accomplish superior performance.

3. Materials and Methods

3.1. Depthwise Separable Convolution

The use of depthwise separable convolution [25] is an effective approach in convolutional neural networks (CNNs) that can reduce the model's parameters while improving its computational efficiency. Unlike traditional convolutional layers, depthwise separable convolution involves using a set of learnable filters to perform the convolution operation on individual input channels, thereby significantly reducing the number of parameters. This technique proves particularly useful when dealing with high-dimensional inputs.

By contrast, depthwise separable convolution divides the convolution operation into two separate stages: depthwise convolution and pointwise convolution. During the depthwise convolution stage, each input channel is convolved with a separate filter, resulting in a set of output feature maps. Subsequently, during the pointwise convolution stage, the output feature maps from the depthwise convolution are combined using a 1×1 convolution, resulting in the final output feature maps. The architecture of depthwise separable convolution is illustrated in Figure 1.



Figure 1. Firstly, depthwise convolution is performed on each channel, after which the output is concatenated. Following this, a 1×1 convolution kernel is applied to the concatenated output to perform pointwise convolution and obtain feature maps.

With the ability to effectively capture spatial correlations between features, this approach significantly solving parameter redundancy in the network. This makes depthwise

separable convolution particularly useful in scenarios where computational resources are limited, or when dealing with high-dimensional data, while still maintaining good accuracy.

3.2. Channel Attention Mechanisms

To improve the model's capacity to attend to significant features, our proposed approach integrates channel attention mechanisms [26] into the architecture. This mechanism facilitates the learning of which channels or features are more informative for the given task and assigns weight to each channel according to its relevance. The attention mechanism was used to assign weights to the feature channels that were extracted by the deep separable convolution layers. Channel attention mechanisms are a technique commonly used in deep neural networks to selectively amplify or suppress specific channels of feature maps. By incorporating channel attention mechanisms, the model is able to effectively enhance informative features in the input data and improve the overall network performance.

A set of attention weights for each channel is produced by a separate neural network module, which takes the feature maps as input to learn the attention weights. These weights are then multiplied with the original feature maps, scaling the channels in a way that emphasizes the most informative ones. Figure 2 summarizes the architecture.



Figure 2. A channel attention mechanism is employed by this approach, which utilizes global average pooling on the input feature layer, followed by two fully connected layers. The number of neurons in the first fully connected layer is reduced, while the second fully connected layer has the same number of neurons as the input feature layer. Next, the output of the second fully connected layer passes through a sigmoid activation function that generates a weight value between 0 and 1 for each channel in the input feature layer. The original input feature layer is then multiplied by these learned channel weights, resulting in a refined feature representation.

Channel attention mechanisms have been widely recognized as an effective means of improving the expressive power of neural networks and enhancing their performance on a range of tasks, including image classification, object detection, and semantic segmentation, etc.

3.3. Residual Network

The Residual Network (ResNet) is an ingenious deep neural network architecture that was proposed in 2015 [27] to address the degradation problem in very deep networks. As the number of layers increases, the accuracy of the network declines due to the challenges posed by optimization with the increasing network depth, which makes it arduous for the network to learn the identity mapping between layers. The primary objective of ResNet is to overcome this phenomenon.

To address this issue, ResNets employ skip connections that enable information to bypass one or more layers in the network. A ResNet's essential unit is the residual block, which consists of several convolutional layers and a skip connection. By adding the input of the residual block to its output, the skip connection ensures that the information does not get lost as it passes through the network's layers. A visual overview of the ResNet architecture is depicted in Figure 3.



Figure 3. Residual Learning: a building block.

ResNets have demonstrated the capability to train deep neural networks with hundreds of layers and maintain high accuracy by utilizing skip connections. Moreover, they have achieved state-of-the-art results on various benchmark datasets, and are widely adopted in computer vision tasks including image classification, object detection, and semantic segmentation.

A Residual network architecture was employed in the paper, allowing for the effective training of neural networks with a large number of convolutional layers. Residual connections introduced shortcut connections that bypass multiple layers, allowing for more efficient propagation of gradients and preventing vanishing gradients, thus improving convergence speed and overall performance of the network. Residual connections were utilized to construct a neural network with multiple convolutional layers that effectively learned features for link prediction.

3.4. Our Model

Our work introduces a novel neural model for link prediction that utilizes deep learning techniques to accurately capture the complex interactions between input entities and relations. Specifically, we introduce a combination of depth-separable convolution, residual network, and channel attention mechanism to capture complex features and patterns in the input data. The core of our model lies in the use of depth-separable convolution, which enables efficient processing of high-dimensional data while resolving the parameters redundancy.

We also incorporate a residual network architecture to effectively train very deep neural networks, allowing our model to capture and learn intricate patterns in the data. Additionally, our model utilizes a channel attention mechanism to assign importance weights to each channel in the input feature layer, improving the model's ability to focus on relevant information. Overall, our model presents a powerful approach for link prediction that achieves better results by leveraging the benefits of convolution, residual network, and channel attention mechanism. Figure 4 summarizes the architecture.

During the feedforward pass, our model uses k as the embedding dimension, where we have two embedded executive row vector matrices: one for the entities, represented as $E^{|\varepsilon| \times k}$, and another for the relationships, represented as $R^{|Y| \times k}$. Here, $|\varepsilon|$ and |Y| denote the entity and relation cardinalities, respectively.

After connecting the embedded *E* and *R*, the input is passed through a Dropout layer before being fed into a 2D depth-separable convolution layer with a filter ω , as well as a residual network layer.

The channel attention mechanism is applied to transform the output of the input vector v_{in} in the residual network, yielding a representation of the output as the sum of the input vector and the transformed vector:

$$v_{R-out} = \sigma(W_2\delta(W_1v_{in})) \tag{1}$$

The expression provided above denotes the outcome of implementing the channel attention mechanism on the input vector v_{in} in the residual network. The σ function refers to the Sigmoid function, and the δ function refers to the ReLU function. The variables $W_1 \in R^{\frac{C}{r} \times C}$ and $W_2 \in R^{C \times \frac{C}{r}}$ are weight matrices used in the full connection layer. To simplify the process, a parameter r is introduced to decrease the dimension of the fully connected layer.



Figure 4. In our model, entity and relationship embeddings are first processed with Dropout and Concatenation, followed by a Depth Separable Convolution layer. This results in the generation of feature maps which are subsequently compressed and mapped into a k-dimensional subspace, which are then embedded and compared to all candidate objects.

For feature extraction of the input vector v_{in} , we utilize depth-separable convolution. After each layer of standard convolution is performed for feature extraction, an output channel is generated. Once the convolution operation is completed, a 1×1 convolution is utilized to modify the channel count. The resulting output is denoted as $V_{DC_{out}}$, which can be expressed as:

$$\mathcal{V}_{DC_{out}} = \varphi(v_{in}) \tag{2}$$

We perform feature extraction on the input vector v_{in} using depth-separable convolution, which involves standard convolution in each layer, producing an output channel. Following the completion of the convolution operation, a 1×1 convolutional layer is employed to modify the dimensionality of the output, producing a resulting feature map labeled as $V_{DC_{out}}$. The function $\varphi(V_{in})$ combines the convolution and channel adjustment operations [25]. Finally, the scoring function can be defined as:

$$Score(e_{head}, e_{tail}) = \sigma(\varphi(v_{in}) \oplus v_{R-out})$$
(3)

We apply the logistic sigmoid function $\sigma(\cdot)$ to compute the scores $Score(e_{head}, e_{tail})$, and then minimize the binary cross-entropy loss given below to optimize the model parameters:

$$\mathcal{L} = -lab \cdot \log(Score(e_{head}, e_{tail})) - (1 - lab) \cdot \log(1 - Score(e_{head}, e_{tail}))$$
(4)

where $Score(e_{head}, e_{tail})$ is the predicted probability of the existence of a link between the head entity e_{head} and tail entity e_{tail} , and *lab* is the ground-truth label (either 0 or 1).

To improve training speed and stability, we employ rectified linear unit (ReLU) activation and a technique called batch normalization (BN) [28]. Dropout is used to regularize the model in multiple stages, including on embeddings, feature maps, and hidden units. We optimize our model using Adam [29] and apply label smoothing to mitigate overfitting. Table 1 list the main factors aimed to be optimized:

Parameters	Definition	
<i>W</i> ₁ , <i>W</i> ₂	Channel attention mechanism weight matrices	
r	Dimension of the fully connected layer	
σ	Sigmoid function	
δ	ReLU function	

Table 1. Summary of the Main Factors aimed to be optimized.

By incorporating these optimization parameters and their practical values, our model demonstrates superior performance in predicting links in knowledge graphs, outperforming existing models and providing a foundation for further research in this field.

4. Experiments

4.1. Datasets

ResE was evaluated on two standard datasets, namely WN18RR and FB15k237, both of which are subsets of commonly used benchmarks, WN18 and FB15k. These datasets are often considered straightforward due to the high proportion of relations that are reversible, which facilitates the prediction of the majority of test triples. WN18RR and FB15k237 were developed to address the issue of reversible relations in knowledge base completion tasks, making them more challenging and realistic. Table 2 summarizes the statistical information of both datasets.

Table 2. Summary of Experimental Dataset Statistics.

Benchmarks	Entities	Relations
WN18RR	40,943	11
FB15k-237	14,541	237

4.2. Evaluation Criteria

The task of Knowledge Graph Completion, also known as link prediction, aims to predict a missing entity in a triple, given the other two entities and the relation. This involves inferring either h or t when given (r, t) or (h, r), respectively. The performance of a model is evaluated by ranking the scores produced by a score function f on the test data.

To assess the model's performance, we adopt the "Filtered" setting procedure. For each valid test triple (h, r, t), we generate a set of corrupted triples by replacing either h or t with each of the other entities in E. Afterward, we rank the valid test triple and the corrupted triples based on their scores in ascending order. The procedure excludes any corrupted triples that exist in the KB.

We employ three widely used evaluation metrics, namely, mean rank (MR), mean reciprocal rank (MRR), and Hits@10, to assess the performance of the model. Better model performance is indicated by a decrease in MR or an increase in MRR or Hits@10.

4.3. Training Regime

In this study, we adopt the widely used Bernoulli sampling method, which has been commonly employed in the literature [10,11], to generate negative triples. To generate invalid triples, we substitute either the subject or object entity in a valid triple with other entities in the knowledge base.

To optimize the model parameters, we employ the Adam optimization algorithm and use an activation function that rectifies negative inputs and leaves positive inputs unchanged. We conduct a hyperparameter tuning process for the ResE model through a grid search method, where the performance metric used is the mean reciprocal rank (MRR) on the validation set.

Our experimental results indicate that the ResE model performs optimally on the WN18RR and FB15K-237 datasets with the following hyperparameters: embedding dropout rate of 0.2, embedding size of 200, batch size of 256, 0.005 for learning rate and 0.2 for label smoothing.

5. Results

In this study, Tables 3 and 4 presents a thorough comparison of the ResE's performance with that of previously published methods, holding experimental settings constant across all experiments. Our findings show that the ResE model surpasses other models in performance, exhibiting the lowest mean rank (MR) and the highest Hits@10 scores on the WN18RR dataset, in addition to the rewarding Mean Reciprocal Rank and Hits@10 scores on the FB15K-237 dataset. Specifically, the ResE model demonstrates significant improvement over the closely related TransE model on both datasets, with a relative improvement of approximately 43% in MR and 23% in MRR on FB15K-237, in addition to a 14.8% absolute improvement in Hits@10. Notably, the ResE model also outperforms other models, including DistMult and ComplEx.

Table 3. Experimental results on WN18RR datasets.

Methods	MR	MRR	Hit@10
TransE	3385	0.226	50.1
DisMult	5110	0.431	48.9
ComplEX	5261	0.429	51.2
ConvE	5277	0.462	48.3
ConvKB	2554	0.248	52.5
ResE	3485	0.512	58.4

Table 4. Experimental results on FB15K-237 datasets.

Benchmarks	Entities	Relations	
TransE	347	0.294	46.5
DisMult	254	0.241	41.9
ComplEX	339	0.247	42.8
ConvE	246	0.316	49.1
ConvKB	257	0.396	51.7
ResE	198	0.363	53.4

Prior research has demonstrated the competitive performance of TransE, a popular model for knowledge graph completion tasks [13,16,17]. However, recent studies have suggested that the CNN-based embedding model ConvE may outperform TransE under certain conditions. Nevertheless, our experiments indicate that TransE remains a strong baseline model, as it exhibits superior performance to ConvE on the WN18RR dataset with respect to both indicators. These results suggest that ResE has the potential to excel in tasks related to knowledge graph inference and warrants further exploration.

6. Conclusions

In this study, we introduced ResE, a novel embedding model designed to predict links in knowledge graphs. The model leverages multiple layers of depth-separable convolution and incorporates residual blocks with channel attention mechanisms. Our experimental results demonstrate that ResE surpasses previously published models, including the closely related TransE model, achieving outstanding performance on both WN18RR and FB15K-237 datasets with satisfactory metrics.

The superior performance of ResE on both datasets, as evidenced by the best mean rank (MR) and highest Hits@10 scores, underscores its potential as a promising model for knowledge graph completion tasks and warrants further investigation. In light of these findings, future work should focus on several directions to further enhance the performance and applicability of ResE.

Firstly, we plan to extend ResE to new applications where data can be formulated in the form of triples, such as modeling relationships between different entities in various domains.

This extension will enable researchers to explore the adaptability and generalizability of ResE to diverse contexts.

Secondly, although ResE currently employs a comparatively shallow architecture in contrast to typical convolutional models used in computer vision, our results underscore the effectiveness of depth-separable convolution and residual blocks with channel attention mechanisms in knowledge graph modeling. Consequently, future research could investigate the potential benefits of deeper convolutional models, possibly leading to even more significant improvements in performance.

Furthermore, given the rapid advancements in the field of deep learning and knowledge graph representation learning, it is essential to stay up-to-date with new techniques and developments. Incorporating state-of-the-art approaches into the ResE model will help ensure its continued success and maintain its relevance in the research community.

In conclusion, ResE is a fast, expressive, and robust model that achieves excellent results in knowledge graph completion tasks. The results of this study indicate that the proposed approach has the potential to significantly contribute to the advancement of knowledge in this domain. We anticipate that ResE will serve as a solid foundation for future research endeavors, ultimately leading to further insights and breakthroughs in the field of knowledge graph representation learning. By addressing these future directions and keeping pace with the latest developments, we expect ResE to maintain its position as a state-of-the-art tool for knowledge graph completion and facilitate the discovery of novel relationships between entities across a wide range of applications.

Author Contributions: Conceptualization, X.L. and H.Y.; methodology, H.Y.; software, H.Y.; validation, X.L., H.Y. and C.Y.; formal analysis, H.Y.; investigation, X.L.; resources, H.Y.; data curation, C.Y.; writing—original draft preparation, H.Y.; writing—review and editing, C.Y.; visualization, H.Y.; supervision, X.L.; project administration, X.L.; funding acquisition, X.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by Zhengzhou collaborative innovation major project 20XTZX06013 and Key scientific research project of colleges and universities in Henan Province 22A520042.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Acknowledgments: We express our sincere appreciation to Weixing Zhang, Nan Lin, and Lipeng Xie for their constructive feedback and insightful discussions throughout the course of this work. Additionally, we extend our gratitude to Yangjie Cao for his invaluable advice in the optimization of the model structure. With their contributions, we have achieved satisfactory results on the WN18RR and FB15K-237 datasets. Finally, we express our sincere gratitude to the Zhengzhou collaborative innovation major project (No. 20XTZX06013) and the Key scientific research project of colleges and universities in Henan Province (No. 22A520042) for the strong support of this research.

Conflicts of Interest: The authors declare no conflict of interest.

References

- Lehmann, J.; Isele, R.; Jakob, M.; Jentzsch, A.; Kontokostas, D.; Mendes, P.N.; Hellmann, S.; Morsey, M.; van Kleef, P.; Auer, S.; et al. DBpedia—A large-scale, multilingual knowledge base extracted from Wikipedia. *Semant. Web* 2015, 6, 167–195. [CrossRef]
- Suchanek, F.M.; Kasneci, G.; Weikum, G. Yago: A core of semantic knowledge. In Proceedings of the The Web Conference, Banff, AB, Canada, 8–12 May 2007.
- 3. Ioffe, S.; Szegedy, C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *arXiv* 2015, arXiv:1502.03167,
- Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the Inception Architecture for Computer Vision. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 2818–2826.
- Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.E.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 1–9.
- 6. Srivastava, N.; Hinton, G.E.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.

- 7. Dettmers, T.; Minervini, P.; Stenetorp, P.; Riedel, S. Convolutional 2D Knowledge Graph Embeddings. In Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018.
- Toutanova, K.; Chen, D. Observed versus latent features for knowledge base and text inference. In Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality, Beijing, China, 26–31 July 2015; Association for Computational Linguistics: Beijing, China, 2015; pp. 57–66. [CrossRef]
- Bordes, A.; Usunier, N.; García-Durán, A.; Weston, J.; Yakhnenko, O. Translating Embeddings for Modeling Multi-relational Data. In Proceedings of the NIPS, Lake Tahoe, NV, USA, 5–8 December 2013.
- 10. Wang, Z.; Zhang, J.; Feng, J.; Chen, Z. Knowledge Graph Embedding by Translating on Hyperplanes. In Proceedings of the AAAI Conference on Artificial Intelligence, Quebec, QC, Canada, 27–31 July 2014.
- 11. Lin, Y.; Liu, Z.; Sun, M.; Liu, Y.; Zhu, X. Learning Entity and Relation Embeddings for Knowledge Graph Completion. In Proceedings of the AAAI Conference on Artificial Intelligence, Austin, TX, USA, 25–30 January 2015.
- 12. Ji, G.; He, S.; Xu, L.; Liu, K.; Zhao, J. Knowledge Graph Embedding via Dynamic Mapping Matrix. In Proceedings of the Annual Meeting of the Association for Computational Linguistics, Beijing, China, 26–31 July 2015.
- Nguyen, D.Q.; Sirts, K.; Qu, L.; Johnson, M. STransE: A novel embedding model of entities and relationships in knowledge bases. In Proceedings of the North American Chapter of the Association for Computational Linguistics, San Diego, CA, USA, 12–17 June 2016.
- Ji, G.; Liu, K.; He, S.; Zhao, J. Knowledge Graph Completion with Adaptive Sparse Transfer Matrix. In Proceedings of the AAAI Conference on Artificial Intelligence, Phoenix, AZ, USA, 12–17 February 2016.
- 15. Liu, H.; Wu, Y.; Yang, Y. Analogical Inference for Multi-relational Embeddings. In Proceedings of the International Conference on Machine Learning, Sydney, NSW, Australia, 6–11 August 2017.
- Trouillon, T.; Welbl, J.; Riedel, S.; Gaussier, É.; Bouchard, G. Complex Embeddings for Simple Link Prediction. In Proceedings of the International Conference on Machine Learning, New York, NY, USA, 19–24 June 2016.
- 17. Nickel, M.; Rosasco, L.; Poggio, T.A. Holographic Embeddings of Knowledge Graphs. In Proceedings of the AAAI Conference on Artificial Intelligence, Phoenix, AZ, USA, 12–17 February 2015.
- Duvenaud, D.K.; Maclaurin, D.; Aguilera-Iparraguirre, J.; Gómez-Bombarelli, R.; Hirzel, T.D.; Aspuru-Guzik, A.; Adams, R.P. Convolutional Networks on Graphs for Learning Molecular Fingerprints. *arXiv* 2015, arXiv:1509.09292.
- 19. tau Yih, W.; Toutanova, K.; Platt, J.C.; Meek, C. Learning Discriminative Projections for Text Similarity Measures. In Proceedings of the Conference on Computational Natural Language Learning, Portland, OR, USA, 23–24 June 2011.
- Kim, Y. Convolutional Neural Networks for Sentence Classification. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, Doha, Qatar, 25–29 October 2014.
- 21. Shen, Y.; He, X.; Gao, J.; Deng, L.; Mesnil, G. Learning semantic representations using convolutional neural networks for web search. In Proceedings of the 23rd International Conference on World Wide Web, Seoul, Republic of Korea, 7–11 April 2014.
- 22. Kalchbrenner, N.; Grefenstette, E.; Blunsom, P. A Convolutional Neural Network for Modelling Sentences. *arXiv* 2014, arXiv:1404.2188.
- 23. LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [CrossRef]
- 24. Collobert, R.; Weston, J.; Bottou, L.; Karlen, M.; Kavukcuoglu, K.; Kuksa, P.P. Natural Language Processing (Almost) from Scratch. *arXiv* 2011, arXiv:1103.0398.
- Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.C. MobileNetV2: Inverted Residuals and Linear Bottlenecks. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 4510–4520. [CrossRef]
- Hu, J.; Shen, L.; Sun, G. Squeeze-and-Excitation Networks. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 7132–7141.
- He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
- Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet classification with deep convolutional neural networks. *Commun. ACM* 2012, 60, 84–90. [CrossRef]
- 29. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. arXiv 2014, arXiv:1412.6980.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.