

Knowledge Reasoning Based on Neural Tensor Network

¹Jian-Hui Huang, ²Jiu-Ming Huang, ³Ai-Ping Li, ⁴Yong-Zhi Tong

¹Massive Data Processing Lab National University of Defense Technology, Changsha, China

²Massive Data Processing Lab National University of Defense Technology, Changsha, China

³Massive Data Processing Lab National University of Defense Technology, Changsha, China

⁴Massive Data Processing Lab National University of Defense Technology, Changsha, China

¹nudt_huangjianhui@163.com

Abstract: Knowledge base (KBs) is a very important part of applications such as Q&A system, but the knowledge base is always faced with incompleteness and the lack of inter-entity relationships. Knowledge reasoning is an important part of the construction of knowledge base, and is intended to find a way to supplement these missing relationships. This paper attempts to explore the model complexity of neural tensor network, a very important method of knowledge reasoning, and the reasoning accuracy. By increasing the number of slices in the tensor network layer, the number of parameters to be trained by the model is increased, thereby increasing the complexity of the model. The experimental results show that the number of slices is improved, which is helpful to increase the reasoning accuracy of the model, while the time consumption does not show obvious growth. The accuracy of the model on WordNet and FreeBase increased 2% and 3.2% respectively.

1. Introduction

The knowledge base is intended to store a large amount of information in a structured way, and the data is organized in a knowledge-dependent manner in which the nodes represent the entities and the edges correspond to the relationships among the entities. Knowledge bases are considerably useful resource for many nature natural language processing tasks such as information retrieval, recommendation systems and so on. However, the knowledge base, including some well-known knowledge base such as Yago [3], WordNet [4] or the Google Knowledge Graph, suffer from incompleteness and a lack of auto reasoning capability. Hence, learning new facts based on the knowledge bases is an essential way to improve them. Much previous work (probability graph model and inductive logic programming and Markov Logic Network et al.) has focused on completing existing knowledge bases using patterns or classifiers applied to external large text corpora. However, not all common knowledge that is obvious to people which we called common sense, is expressed in Text. For example, we find a special bird, without special information, we know that it should have wings and will fly and other information, because these can be obtained through our common sense, without the need for additional text information. Obtain. Therefore, learning new relations (triples) based on knowledge bases has been increasingly popular. Mukherjee et al (2013)[5] used a matrix tri-factorization approach to reasoning new facts

in knowledge bases. Socher et al.(2013)[1] introduced a neural tensor network to extracting common sense, which is the base of this paper. Guoliang Ji et al(2014)[2] applied a neural tensor network to reasoning over relations based on Chinese knowledge bases.

In this paper, I will explore the relationship between the number of slices and the reasoning accuracy of the model through specific experiments. NTN transforms the entities in the knowledge base into corresponding vectors, which can capture facts and their certainty about that entity, to complete the initialization of the entity vectors. For sharing statistical strength among the entities that contain similar substrings, each entity is represented as the average of its word vectors. Each relation corresponds to a group of parameters of neural tensor networks. The entities and relationships can interact well through the tensor in the neural tensor layer of networks.

The main contribution of this paper is to explore the relationship between model complexity and reasoning accuracy through experiment, and to make a reference to the choice of model complexity when using this model to reason. The paper is organized as follows. Section 2 and section 3 introduce the Neural Tensor Network model and some related work respectively. Section 4 analyzes the model complexity. Section 5 reports the results of experiments. Section 6, I summarize my contribute and consider the further work directions.

2. Related Work

Here are the two parts involved in neural tensor networks and natural language processing: semantic vector space and depth learning.

2.1 Semantic Vector Space

Mapping the corresponding words to the low-dimensional vector semantic space is an important technique for natural language processing related tasks. It is also an important basis for other natural language processing tasks, such as POS tagging, naming entity recognition. Word vector related technology has been very mature. Neural language models (Bengio et al., 2003; Collobert and Weston, 2008) have been shown to be very powerful at language modeling, a task where models are asked to accurately predict the next word given previously seen words [7]. From the language model in 2003 to the 2008 word2vec and so on a series of word vector training technology development, makes the implied semantics of the word quite perfect.

2.2 Deep Learning

Deep learning technology in the visual and language technology has been quite mature, but in terms of text, relatively new. Deep learning technology in other aspects of the major achievements also makes the relevant personnel pay more attention to its natural language processing direction of the study. Nickle [6] applied a tensor factorization method for multi-relational learning, where a knowledge base was regarded as a three dimensional tensor. Bowman [7] introduced a recursive neural tensor networks model on a new corpus of constructed examples of logical reasoning in short sentences. Bordes et al [13] deal with the issue of weak interaction between entity vectors through multiple matrix products followed by Hadamard products.

3. Neural Tensor Network

This section introduces the neural tensor network model (NTN). NTN reasons over database entries which existed in knowledge base already by learning vector representations for each entity. We can see from Fig. 1, each relation triple is described as (e1,R,e2) and database entities are given as input to that relation's model. The model returns a high score if the relationship exist in database and a low one otherwise. This will answer any fact, whether implicit or explicitly mentioned in the database, with a certainty confidence.

Next, I will introduce the model from two aspects: model structure, training goal.

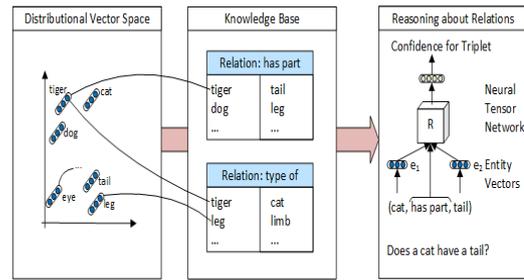


Figure 1. This figure shows how words in a knowledge base are mapped to a low-dimension vectors and averaged to construct entity vectors. Entity relation triples are inputted into a neural tensor network, which calculates the confidence that the two entities are in a relationship.

3.1 Model Structure

In this section, we introduce the NTN model structure. Firstly, we need to define some crucial parameters indexed by R for each relation's scoring function. If we use e1,e2 be the vector representations of any two entities in the database. Then the Neural Tensor Network (NTN) computes the confidence of a given triple (e1,R,e2) through a bilinear tensor layer which relates two entity vectors across multiple dimensions and the formula as follow.

$$score(e1, R, e2) = u_R^T f(e1^T W_R^{[1:k]} e2^T + V_R \begin{bmatrix} e1 \\ e2 \end{bmatrix} + b_R) \quad (1)$$

where f is a activation function, $W_R^{[1:k]} \in R^{d*d*k}$ is a tensor, the entity vectors $e1, e2 \in R^d, V_R \in R^{k*2d}, b_R \in R^k, u_R \in R^k$.

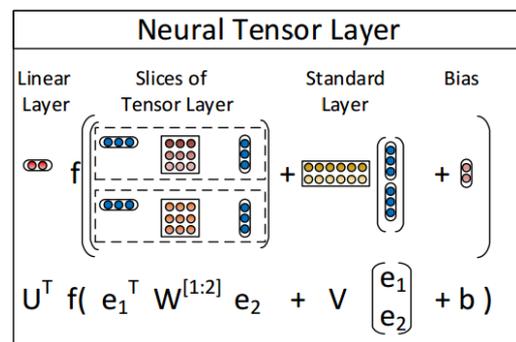


Figure 2. A visualization of the parameters of Richard Socher's Neural Tensor Network Model with k = 2 slices.

3.2 Training Objective

In the data set, the samples are represented in the form of triples. triple described as (e1,R,e2), where ei(i = 1, 2),R represent entities and relationship respectively, corresponds a tensor network whose inputs are ei(i = 1, 2). The main idea is that each triplet in the training set which be regard as positive samples and represent it with $T^R = (e1^R, R, e2^R)$, should receive a higher score than a triplet in which one of the entities is replaced with a random entity, where there triplets are regards as negative samples

and represent it as $T_c^R = (e1^R, R, ec^R)$. We minimize the objective as follow:

$$J = \sum_{i=1}^N \sum_{c=1}^C \max \left(0, 1 - \text{score}(T^{(i)}) + \text{score}(T_c^{(i)}) \right) + \lambda \|\Omega\|_2^2 \quad (2)$$

The loss function is a hinge-loss function and is followed by a regular term to prevent over-fitting. We denote the set of all relationships' NTN parameters by $\Omega = U, V, W, b, E$, where U, V, W, b are model parameters and E is word vectors.

4. Analysis of Neural Tensor Network complexity

Some experiences of machine learning and deep learning tell us that it is possible to increase the complexity of the model and increase the performance of the model when the model is not over-fitting. Based on this argument, here I try to explore the model complexity and the reasoning ability of the neural tensor network model through experimental comparison. Here we assume that the word vector and the dimension of the entity vector take $200, e_i \in R^{200}$. then the number of parameters of the neural tensor network layer is calculated as follows: The number of parameters of tensor $W_R^{[1:k]} \in R^{d*d*k}$ is $40000*k$, the number of parameters in $V_R \in R^{2*d}$ is $2*k$. It can be seen that the most important factor affecting model complexity lies in tensor. The number of parameters that need to be trained in the model represents the complexity of the model. Equation (3) (4) shows the relationship between the total number of model parameters and the number of slices in the tensor network

$$N(k) = N(U) + N(W) + N(b) + N(V) + N(E) \quad (3)$$

$$N(k) = k + 200*200*k + k + 2*200*k + 400 \quad (4)$$

Table 1: The parameters of the model that need to be trained with the increase of the slice

Slices	3	4	5	6
Parameter(*10000)	12	16	20	24
Slices	7	8	9	10
Parameter(*10000)	28	32	36	40

5. Experiments

Experiments are conducted on both WordNet[1] and FreeBase[2] predict whether some relations hold or not using other facts in the database. Before reasoning with Neural Tensor Network, We have a very important step is to convert the entity into the corresponding entity vector. In this paper, I will use a simple random initialization method for entity Vector initialization.

5.1 Entity to Semantic Vector

Firstly, we simply initialize each word vector $x \in R^{200}$ by sampling it from a zero mean Gaussian distribution: $x \sim N(0, \sigma^2)$. Next, the entity vector is converted to the average of the vector of the constituent words. On the right side of Figure 1, we can see that the entity vector of the input entity Bengal tiger is:

$$\zeta(\text{Bengal tiger}) = \zeta(\text{Bengal}) + \zeta(\text{tiger}) \quad (5)$$

The advantage of doing so is that entities with the same substring can share the same part of the semantics. For example, Bengal tiger and South china tiger. The two entities have the same sub-string tiger, using the formula (5), can make two entities have the same semantic part.

5.2 Datasets

Table II is the data set used in this paper. For WordNet, I use 112,581 relational triplets for training. In total, there are 11 different relations which including 38,696 unique entities. In addition, the experimental will use 2609 relational triplets for validation, and use 10544 relational triplets for final test. For FreeBase, I use 316232 relational triplets for training. In total, there are 75043 unique entities in 13 different relations. In contrast to the previous practice, some relationship of FreeBase is considered to be difficult to reason (such as place of birth, place of death, location, children, spouse, parents) has not been removed in this paper. Because the need to explore here is the relationship between model complexity and accuracy.

Table 2: The statistics for WordNet and Freebase including number of different relations #R.

Dataset	#R	#Ent	#Train	#Dev	#Test
WordNet	11	38696	112581	2609	10544
FreeBase	13	75043	316232	5908	23733

5.3 Environment of Experiment

Table III shows the operating environment on which this experiment relies. As the complexity of the model and the increase in the size of the data, the consumption of resources must be an important factor in our consideration. So it is necessary to discuss the environment in which the whole program is running. This paper will use the time as an important indicator to reflect the complexity of the model.

Table 3: The environment of experiments

Operating System	Win7_64bit
Language	Python2.7
Open Source Package	Numpy, Scipy
Memory of Computer	16G
Cpu	Core-i5-4460

5.4 Experimental Results

In this section, I will use the chart comparison and the use of python open source package matplotlib drawing way to visualize the results. In the first part, I will illustrate the complexity of the model through the entire program run time. Next, observe the change in the reasoning accuracy of the relationship reasoning in WordNet and FreeBase as the complexity of the model increases. In this paper, I will be the most common slice (slice = 3) as the baseline.

In table IV and table V ,I illustrate the complexity of the corresponding model by the number of slices, the total time consumed by the program run, and the number of model parameters .Where the reasoning accuracy is the average reasoning accuracy of all the relationships in the corresponding data set. Through the results in the table, we can see that with the increase of the number of slices, the running time of the model shows a linear increase rule. The accuracy of the reasoning is also changing. The reasoning accuracy of the FreeBase and WordNet are reach the peak when the slice is 6 and 7 ,respectively. the optimal reasoning results were 0.8242, 0.8532, compared to the baseline, respectively, increased 0.02 and 0.032. Then with the increase of the slice, the reasoning accuracy is reduced, which indicates that the model has been fitted.

Fig. 3 and Fig. 4 visualize the results by drawing, showing the specific changes in the reasoning accuracy of the various data relations in the two datasets as the model complexity changes .From Fig. 3 we can see that "similar to" and "domain topic" fluctuate the most, affected by the complexity of the model. It can be seen from Fig. 4 that the reasoning of "children", "place of birth", "place of death" and so on is still the least accurate. The greatest impact on model complexity is "children " and "place of birth".

Table 4: The relations between number of slices running time and reasoning accuracy (WordNet)

Slice	3	4	5	6	7	8	9	10
Parameters	120000	160000	200000	240000	280000	320000	360000	400000
Time (min)	1253	1475	1800	2329	2413	2536	2880	3020
Average Accuracy	0.8045	0.8081	0.8036	0.8242	0.8532	0.8101	0.7918	0.7803

Table 5: The relations between number of slices ,running time and reasoning accuracy (FREEBASE)

Slice	3	4	5	6	7	8	9	10
parameter	120000	160000	200000	240000	280000	320000	360000	400000
Time (min)	2320	2578	2866	3004	3159	3400	3625	3785
Average Accuracy	0.8213	0.8236	0.8320	0.8403	0.8538	0.8440	0.8220	0.8102

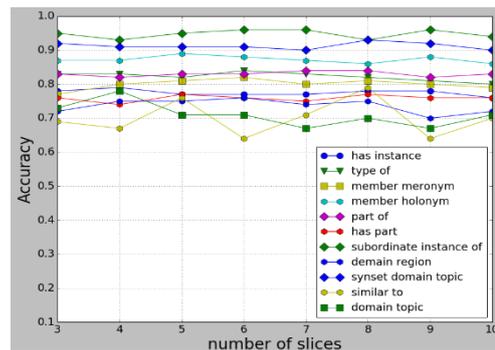


Figure 3. Changes of 11 relationships in WordNet

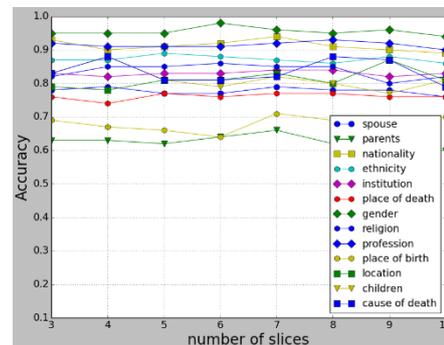


Figure 4 .Changes of 13 relationships in FreeBase

6. Conclusion

The complexity of the model can be determined by choosing the appropriate super-Parameters, and the appropriate complexity is helpful to improve the performance of the model. In the neural tensor network, the number of slices in tensor has the greatest influence on the complexity of the model. Hence, I started from the number of slices to explore the relationship between model complexity and model performance. Finally, through experiments, we know that the appropriate increase in the number of slices, is indeed conducive to improving the model of reasoning accuracy, of course, this also inevitably increases the program running time . The theme of this paper is to explore the effect of the complexity of the neural tensor network model on the reasoning accuracy ,So all of the entity vector initialization using a random initialization method .

Richard Socher in his paper [1] through the experimental show that the use of pre-training method to initialize the entity vector, is very helpful to increase of the reasoning accuracy. Using word2vec[8], glove even more complex language model, such as LSTM, for word vector pre-training, will become the focus of my next step.

Acknowledgment

The work described in this paper is partially supported by The National Key Research and Development Program of China (2016QY03D0601, 2016QY03D0603) and National Natural Science Foundation of China (No.61502517, No.61672020, No.61662069).

References

- [1] R Socher., Chen, D., Manning, C.D., Ng, Y.: Reasoning With Neural Tensor Networks for Knowledge Base Completion. In: Advances in Neural Information Processing Systems 26 2013
- [2] Guoliang Ji, Yinghua Zhang, Hongwei Hao, and Jun Zhao: Reasoning Over Relations Based on Chinese Knowledge Bases. Springer International Publishing,2014,8801:141-150
- [3] F. M. Suchanek, G. Kasneci, and G. Weikum. Yago: a core of semantic knowledge. In Proceedings of the 16th international conference on World Wide Web, 2007.
- [4] G.A. Miller. WordNet: A Lexical Database for English. Communications of the ACM, 1995.
- [5] Mukherjee, T., Pande, V., Kok, S.: Extracting New Facts in Knowledge Bases:-A matrix trifactorization approach. In: ICML Workshop on Structured Learning: Inferring Graphs from Structured and Unstructured Inputs ,2013
- [6] Nickel, M., Tresp, V., Chen, Kriegel, H.P.: A Three-Way Model for Collective Learning on Multi-Relational Data. In: Proceedings of the 28th International Conference on Machine Learning 2011
- [7] Huang, E.H., Socher, R., Manning, C.D., Ng, Y.: Improving Word Representations via Global Context and Multiple Word Prototypes. In: Annual Meeting of the Association for Computational Linguistics, ACL ,2012
- [8] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, Jeffrey Dean. Distributed Representations of Words and Phrases and their Compositionality. arXiv:1310.4546 2013
- [9] Y Goldberg , O Levy. Word2vec Explained: deriving Mikolov et al.'s negative-sampling word-embedding method. Eprint Arxi,2014
- [10] R. Socher, B. Huval, C. D. Manning, and A. Y. Ng. Semantic Compositionality Through Recursive Matrix-Vector Spaces. In EMNLP, 2012.
- [11] Danqi Chen, Richard Socher, Christopher D. Manning, Andrew Y. Ng. Learning New Facts From Knowledge Bases With Neural Tensor Networks and Semantic Word Vectors ICLR 2013
- [12] A Bordes , J Weston , R Collobert , Y Bengio. Learning Structured Embeddings of Knowledge Bases. Aaai Conference on Artificial Intelligence,2011
- [13] Xu Zenglin, Sheng Yongpan, He Lirong, Wang Yafang Knowledge Atlas Technology Journal of Electronic Science and Technology University, Vol. 45, 2016.7 page: 46-52
- [14] R. Collobert and J. Weston. A unified architecture for natural language processing: deep neural networks with multitask learning. In ICML, 2008.
- [15] Li Xiangyang, Miao Zhuang. Free text information extraction technology [J]. Information Science, 2004, 22 (7): 815-821
- [16] LIU Qiao, LI Yang, DUAN Hong, LIU Yao, QIN Zhi-guang Overview of Knowledge Map Construction Technology University of Electronic Science and Technology Information and Software Engineering Institute of Computer Research and Development 2016 page: 582-600
- [17] Lao N, Subramanya A, Pereira F, et al. Reading the web with learned syntactic-semantic inference rules[C]//Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning. Association for Computational Linguistics, 2012.page:1017-1026.
- [18] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In Proceedings of the 2008 ACM SIGMOD international conference on Management of data, SIGMOD, 2008