International Journal of Engineering Research and General Science Volume 2, Issue 6, October-November, 2014
ISSN 2091-2730

Extending Knowledge Bases using various Neural Network Models - A Survey

[1]Sagarika Sahoo, [2]Avani Jadeja

[1]M.E. Scholar, Computer Engineering, HashmukhGoswami College of Engineering, GTU

[2]Assistant Professor, Computer Engineering, HashmukhGoswami College of Engineering, GTU

Abstract— Knowledge bases are an important resource for easily accessible, systematic relational knowledge. They provide the benefit oforganizing knowledge in the relational form but suffer from incompleteness of new entities and relationships. However, each of them (eg. Freebase, Wordnet etc)is based on different severe symbolic framework which makes them hard to use their data for different other purposes in the field of Artificial Intelligence(AI) such as natural language processing (word-sense disambiguation, natural language understanding, ...), vision (scene classification, image semantic annotation, ...) or collaborative filtering. Much work has been done on relation extraction from knowledge bases and extending knowledge bases using various Neural Network models. This paper provides a survey on extending knowledge bases using various Neural Network models and the ideas and strength of those models.

Key Words—Knowledge bases, semi-supervised learning, score function, entity vectors.

I. INTRODUCTION

The fundamental challenge for AIhas always been to be able to gather, organize and make intelligent use of the colossal amounts of information generated daily. Recent developments and collaborative processes has accomplished the task by creating ontologies and knowledge bases such as WordNet [1], Yago [2] or the Google Knowledge Graph are extremely useful resources for query expansion [3], coreference resolution [4], question answering (Siri), information retrieval or providing structured knowledge to users.Much work focused on extending knowledge bases using pattern or classifiers on large text bodies. However, the knowledge that is recognizable is not expressed in the large text corpora. Due to this, they suffer from incompleteness and a lack of reasoning capability.

To take the advantage of these knowledge bases much work has been focused on relation extraction from these knowledge bases by extending them using various Neural Network Models. The aim of this paper is to provide an overview of various methodologies used in extending the knowledge bases.

II. STRUCTURAL EMBEDDING OF KNOWLEDGE BASES

The main idea [8] behind the structural embedding of KBs is the following:

(i) Entities can be modelled in a d-dimensional vector space, termed the “embedding space”. The i^{th}entity is assigned a vector $\mathbf{E_i \in R_d}$.

(ii) Within that embedding space, for any given relation type, there is a specific similarity measure that captures that relation between entities. For example, the *part of* relation would use one measure of similarity, whereas *similar to*would use another. Note that these similarities are not generally symmetric, as e.g. *part of* is not a symmetric relation. It is modelled by assigning for the k_{th}given relation a pair $\mathbf{R_k = (R_{lh,\,k}, R_{rhs,\,k})}$, where $\mathbf{R_{lhs,\,j}}$and $\mathbf{R_{rhs,\,j}}$are both $\mathbf{d \times d}$matrices. The similarity function for a given entity is thus defined as:

$$\mathbf{S_k(E_i,E_j) = ||R_{lhs,\,k}\,E_i - R_{rhs,\,k}\,E_j||_p}$$

using the p-norm. In this work we chose $\mathbf{p = 1}$due to the simplicity of the gradient learning in that case. That is, we transform the entity embedding vectors $\mathbf{E_i}$and $\mathbf{E_j}$by the corresponding left and right hand relation matrices for the relation $\mathbf{R_k}$and then similarity is measured according to the 1-norm distance in the transformed embedding space.

International Journal of Engineering Research and General Science Volume 2, Issue 6, October-November, 2014
ISSN 2091-2730

III. VARIOUS MODELS

The different models, compute a score of how probable it is that two entities are in certain relationship. Let e1, e2 $\in R_d$ be the vector representation of two entities then, the different models of Neural Network based function predicts the relationship of two entities in the knowledge base. Each model assigns a score to a triplet using a function "**g**" measuring how likely the triplet is correct.We now introduce several related models in increasing order of expressiveness and complexity.

A. *DISTANCE MODEL*

The model of Bordes et al. [8] scores relationships by mapping the left and rightentities to a common space using a relationship specific mapping matrix and measuring the L1distance between the two. The scoring function for each triplet has the following form:

$$g(e_1; R; e_2) = \|W_{R,1}\, e_1 - W_{R,2}\, e_2\|_1$$

where$W_{R,1}; W_{R,2} \in \mathbb{R}^{dxd}$are the parameters of relation R's classifier. This similarity-based model scores correct triplet lower (entities most certainly in a relation have 0 distances). All other functions are trained to score correct triplets higher. The function fis trained to rank the training samples below all other triplets in terms of 1-norm distance. It is parameterized by the following neural network:

$$f(e^l_i, r_i, e^r_i) = \|R^{lhs}_{ri} Ev(e^l_i) - R^{rhs}_{ri} Ev(e^r_i)\|_1$$

R^{lhs} and R^{rhs} are both $d \times d \times D_r$tensors, where e.g. R^{lhs}_imeans to select the ith component along the third dimension of R^{lhs}, resulting in a $d \times d$ matrix. E is a $d \times D_e$matrix containing the embeddings of the $D_e$entities and the function $v\ (n): \{1, . . . ,D_e\} \longrightarrow \mathbb{R}^{De}$ maps the entity dictionary index n into a sparse vector of dimension$D_e$consisting of all zeros and a one in the nth dimension.

B. *SINGLE LAYER MODEL*

The Single Layer model [20] tries to alleviate the problems of the distance model by multitask learning and semi supervised learning. This model tries to alleviate the problems of the distance model by connecting the entity vectors implicitly through the nonlinearity of a standard neural network. The architecture deals with raw words, the first layer has to map words into real-valued vectors for processing by subsequent layers of the Neural Network.

Look Up Table Layer: Each word i $\in$ D is embedded into a d-dimensional space using a lookup table$LT_W(\cdot)$:

$$LT_W(i)=W_i$$

where$W \in \mathbb{R}^{d\times|D|}$ is a matrix of parameters to be learnt, $W_i \in \mathbb{R}^d$ is the ith column of W and d is the word vector size (wsz) to be chosen by the user. In the first layer of our architecture an input sentence $\{s_1, s_2, . . . s_n\}$ of n words in D is thus transformed into a series of vectors $\{W_{s1}, W_{s2}, . . . W_{sn}\}$ by applying the lookup-table to each of its words. The structure of the model is shown in figure 1. A word i is then embedded in a $d = \sum_k d^k$ dimensional space by concatenating all lookup-table outputs:

$$LT_{W1,\ldots,WK}(i)^T = (LT_{W1}\,(i^1)^T, . . . , LT_W{}^K(i^K)^T)$$

When a word is decomposed into K elements (features), it can be represented as a tuple $i = \{i_1, i_2, . . . i_K\} \in D_1 \times \cdots \times D_K$, where $D_k$is the dictionary for the kth-element. We associate to each element a lookup-table $LT_{Wk}(\cdot)$, with parameters $W^k \in R^{dk\times|Dk|}$ where $d_k \in N$ is a user-specified vector size.The scoring function has the following form:

$$g(e1;R; e2) = u^T f\,(W_{R,1}e1 + W_{R,2}e2)$$

where $f = \tanh$, $W_{R,1}, W_{R,2} \in \mathbb{R}^{kxd}$ **and** $u \in \mathbb{R}^{kx1}$ are the parameters of relation R's scoring function.

C. HADAMARD MODEL

This model was introduced by Bordes et al. [10] and tackles the issue of weak entity vector interaction through multiple matrix products followed by Hadamard products. It is different to the other models in our comparison in that it represents each relation simply as a single vector that interacts with the entity vectors through several linear products all of which are parameterized by the same parameters.

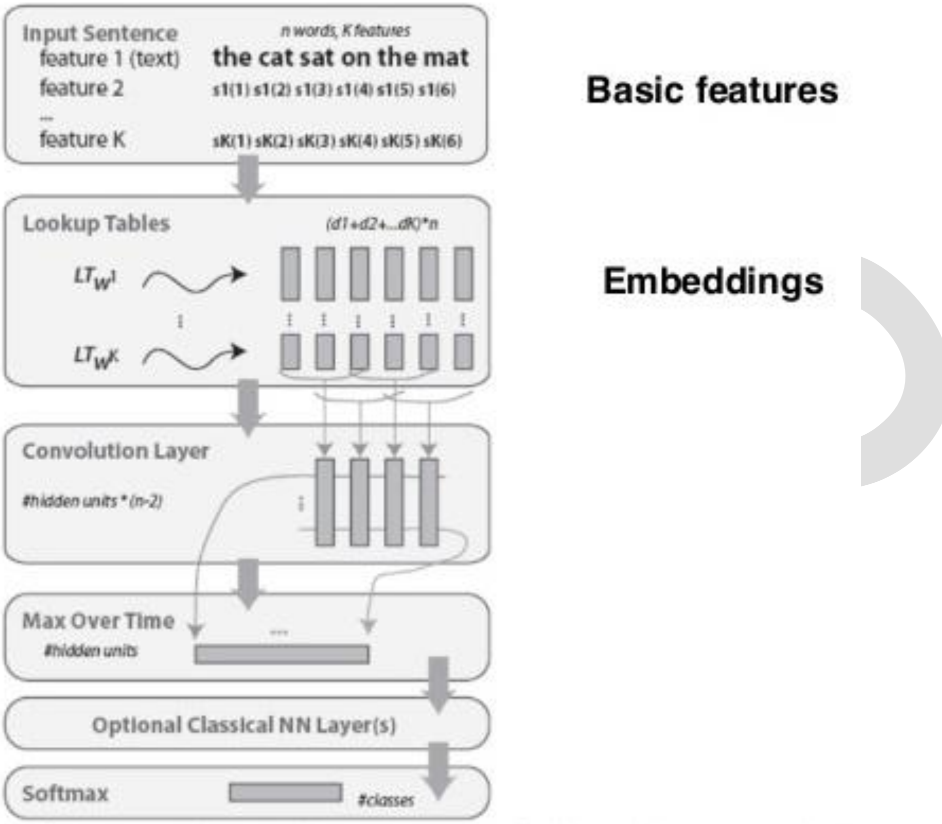

Figure1: A general deep NN architecture for NLP. Given an input sentence, the NN outputs class probabilities for one chosen word.

The semantic matching energy function has a parallel structure first, pairs (lhs, rel) and (rel, rhs) are combined separately and then, these semantic combinations are matched. More precisely the model can be represented as:

$$\mathbf{E}_{\mathbf{lhs(rel)}} = (\mathbf{W}_{\mathbf{ent,l}}\mathbf{E}_{\mathbf{lhs}}) \otimes (\mathbf{W}_{\mathbf{rel,l}}\mathbf{E}_{\mathbf{rel}}) + \mathbf{b}_{\mathbf{l}}$$

$$\mathbf{E}_{\mathbf{rhs(rel)}} = (\mathbf{W}_{\mathbf{ent,r}}\mathbf{E}_{\mathbf{rhs}}) \otimes (\mathbf{W}_{\mathbf{rel,r}}\mathbf{E}_{\mathbf{rel}}) + \mathbf{b}_{\mathbf{r}}$$

$$\mathbf{h}(\mathbf{E}_{\mathbf{lhs(rel)}}, \mathbf{E}_{\mathbf{rhs(rel)}}) = -\mathbf{E}_{\mathbf{lhs(rel)}} \cdot \mathbf{E}_{\mathbf{rhs(rel)}}$$

where $\mathbf{W}_{\mathbf{ent,l}}$, $\mathbf{W}_{\mathbf{rel,l}}$, $\mathbf{W}_{\mathbf{ent,r}}$ and $\mathbf{W}_{\mathbf{rel,r}}$ are $\mathbf{d}\times\mathbf{d}$ weight matrices, $\mathbf{b}_{\mathbf{l}}$, $\mathbf{b}_{\mathbf{r}}$ are **d** bias vectors and $\otimes$ depicts the element-wise vector product. This bilinear parameterization is appealing because the operation $\otimes$ allows encoding conjunctions between **lhs** and **rel**, and **rhs** and **rel**.

D. Bilinear Model

The fourth model [11, 9] fixes the issue of weak entity vector interaction through a relation-specific bilinear form.The scoring function is as follows: $\mathbf{g}(\mathbf{e}_1;\mathbf{R};\ \mathbf{e}_2) = \mathbf{e}^{\mathbf{T}}_{\mathbf{1}}\ \mathbf{W}_{\mathbf{R}}\mathbf{e}_2$; where $\mathbf{W}_{\mathbf{R}} \in \mathbf{R}^{\mathbf{dxd}}$ are the only parameters of relation **R**'s scoring function. This is a big improvement over the two previous models as it incorporates the interaction of two entity vectors in a simple and efficient way.

CONCLUSION

We represented four various models of Neural Networks which enhance the knowledge bases without any external data other than the knowledge bases itself. The various models considered the structural embedding of knowledge base as either objects or entities and the relationship between them and using the scoring function based on their Neural Network Model, unseen relation has been found. Much work is still in progress to improve the above models to extend the knowledge bases and make it more useful for the various purposes in the field of AI.

REFERENCES

[1] G.A. Miller. WordNet: A Lexical Database for English. Communications of the ACM, 1995.
[2] 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.
[3] J. Graupmann, R. Schenkel, and G. Weikum. The SphereSearch engine for unified ranked retrieval of heterogeneous XML and web documents. In Proceedings of the 31st international conference on Very large data bases, VLDB, 2005.
[4] Ng and C. Cardie. Improving machine learning approaches to coreference resolution. In ACL, 2002.
[5] R. Snow, D. Jurafsky, and A. Y. Ng. Learning syntactic patterns for automatic hypernym discovery. In NIPS, 2005.
[6] A. Fader, S. Soderland, and O. Etzioni. Identifying relations for open information extraction. In EMNLP, 2011.
[7] G. Angeli and C. D. Manning. Philosophers are mortal: Inferring the truth of unseen facts. In CoNLL, 2013.
[8] A. Bordes, J. Weston, R. Collobert, and Y. Bengio. Learning structured embeddings of knowledge bases. In AAAI, 2011.
[9] R. Jenatton, N. Le Roux, A. Bordes, and G. Obozinski. A latent factor model for highly multi-relational data. In NIPS, 2012.
[10] A. Bordes, X. Glorot, J. Weston, and Y. Bengio. Joint Learning of Words and Meaning Representations for Open-Text Semantic Parsing. AISTATS, 2012.
[11] I. Sutskever, R. Salakhutdinov, and J. B. Tenenbaum. Modelling relational data using Bayesian clustered tensor factorization. In NIPS, 2009.
[12] M. Ranzato and A. Krizhevsky G. E. Hinton. Factored 3-Way Restricted Boltzmann Machines For Modeling Natural Images. AISTATS, 2010.
[13] D. Yu, L. Deng, and F. Seide. Large vocabulary speech recognition using deep tensor neural networks. In INTERSPEECH, 2012.
[14] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Ng, and C. Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In EMNLP, 2013.
[15] A. Yates, M. Banko, M. Broadhead, M. J. Cafarella, O. Etzioni, and S. Soderland. Textrunner: Open information extraction on the web. In HLT-NAACL (Demonstrations), 2007.
[16] M. Nickel, V. Tresp, and H. Kriegel. A three-way model for collective learning on multirelational data. In ICML, 2011.
[17] A. Bordes, N. Usunier, A. Garca-Durn, J. Weston, and O. Yakhnenko. Irreflexive and hierarchical relations as translations. CoRR, abs/1304.7158, 2013.
[18] J. Turian, L. Ratinov, and Y. Bengio. Word representations: a simple and general method for semi-supervised learning. In Proceedings of ACL, pages 384–394, 2010.
[19] R. Socher, B. Huval, C. D. Manning, and A. Y. Ng. Semantic Compositionality Through Recursive Matrix-Vector Spaces. In EMNLP, 2012.
[20] R. Collobert and J.Weston. A unified architecture for natural language processing: deep neural networks with multitask learning. In ICML, 2008.
[21] R. Socher, E. H. Huang, J. Pennington, A. Y. Ng, and C. D. Manning. Dynamic Pooling and Unfolding Recursive Autoencoders for Paraphrase Detection. In NIPS. MIT Press, 2011.
[22] E. H. Huang, R. Socher, C. D. Manning, and A. Y. Ng. Improving Word Representations via Global Context and Multiple Word Prototypes. In ACL, 2012.
[23] Sagarika Sahoo, NishthaTripathi, AvaniJadeja. Neural Tensor Networks for capturing new facts from knowledge bases. International Journal of Computer Science and Mobile Computing, Vol.3 Issue.10, October- 2014, pg. 860-863