Entity relationship extraction using neural embedding approach *

Sahil Manchanda
154101019
sahil.manchanda@iitg.ernet.in

Patchigolla V.S.S Rahul
154101020
rahul.2015@iitg.ernet.in

Vinayak Jadhav
154101017
vinayak.jadhav@iitg.ernet.in

Manasi Sant
154101022
manasi@iitg.ernet.in

ABSTRACT

The aim of our project is to create a model representing the entities and relations of a knowledge base.

We have used neural embedding approach to score the facts of the KB and rank the facts according to the score obtained from the model. Further task includes identification of hidden facts present in the knowledge base.

1. INTRODUCTION

In the world of Internet in which data is growing at a very rapid rate, there is a need to maintain data in a structured format for better understanding , retrieval , updating and inference of information. One way of doing this is maintaining databases . The limitation of such a system will be the difficulty in finding information and developing complex relation between entities.

Another way of solving this problem is by using Knowledge Bases(KB). Knowledge Bases such as FreeBase and WordNet store information in the form of RDF Triples*(subject, predicate ,object)*. They help in improving tasks like information retrieval and biological data mining.

Our work focuses on modelling these facts and scoring the unseen facts in the database.

2. MODEL

The KB (knowledge base) stores the real world facts as triples. Where each triplet is represented as (e_1, r, e_2) where e_1 and e_2 are entities and r is the relationship between the entities.We used a NTN(Neural Tensor Network) model where entities are represented as d dimensional vectors and relation as combination of bilinear and linear tensor operators.

The first layer of neural network converts the input entities into low dimensional feature vectors. The second layer takes the input entities(feature vectors) and performs the comparison of the entities with relation specific parameters by using a scoring function. Then the learnt representations are used for scoring our test triples to know whether the triplet is a valid triplet or not.[1]

2.1 Representation of Entities

Each input entity is represented as a d dimensional vector. This vector is initialised using *gensim word2vec* model where, wikipedia article of the entity is provided as a corpus for training. These entities are later on updated when the neural network is trained.

2.2 Representation of Relationships

We formulate relations in the form of scoring functions. The scoring function of NTN model is a combination of linear and bilinear tensor operators, which is defined as follows:

$$g(e_1, R, e_2) = u_r^T f(e_1^T B_r^{[1:k]} e_2 + A_r \begin{bmatrix} e_1 \\ e_2 \end{bmatrix}) \quad (1)$$

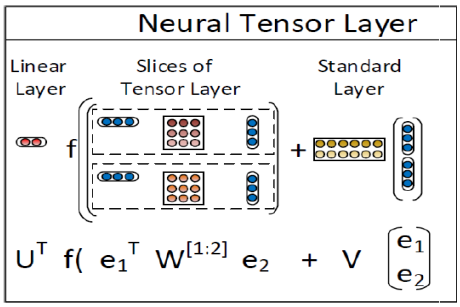

Figure 1: Block diagram of scoring function with two tensor layers

where u_r and A_r are linear tensor operators and B_r is a bilinear operator of a specific relation r. The scoring function returns a high score if the triplet is valid and lower score for invalid triples.

2.3 Training

The objective of the training procedure will be to increase the score of triplets which are present in the KB ($T^{(i)}$) and on the other hand to decrease score of triplets which are not present in the KB ($T_c^{(i)}$). To do this, we corrupt one of the entity in the triple e_1 or e_2 and generate a corrupted triplet (e_1, r, e_c) or (e_c, r, e_2) by randomly sampling from the set of possible entities for that relation.Then we minimize the following objective function:

$$L(\Omega) = \Sigma_{i=1}^{N} \Sigma_{c=1}^{C} max(0, 1 - g(T^{(i)}) + g(T_c^{(i)})) + \lambda ||\Omega||_2^2 \quad (2)$$

*This template is adapted from http://www.acm.org/publications/article-templates/SIG%20Proceedings%20Template-May2015%20Zip.zip

where $g(T)$ is the scoring function and N refer to the no of training triples of a specific relation and C refers to the no of corrupted triples and λ is the regularisation parameter and $||\Omega||_2^2$ refer to standard L_2 regularisation .

The loss function $L(\Omega)$ inherently maximizes the difference between the scores of positive and negative triples.

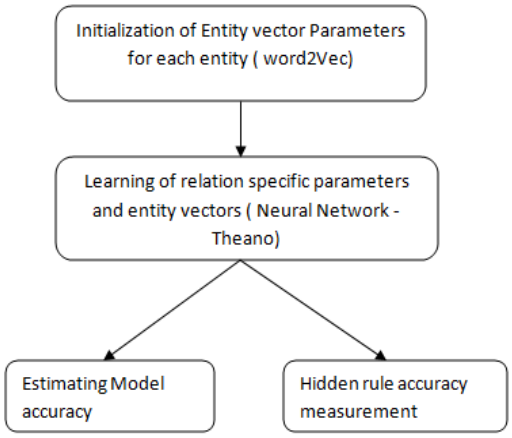

Figure 2: Block Diagram of the project

2.4 Testing

2.4.1 *Built Model accuracy*

Testing on the built model is done using valid and corrupted triples from the test data where the corrupted triplets are obtained same as that of done in training. Ranking on the triples is done based upon score given by the scoring function. Based on these scores we have measured the quality of the built model using *Mean Reciprocal Rank*(MRR) which computes average of the reciprocal of the ranks obtained for each test triple .We have also used *HITS@10* measure. This measure ranks score of test triples along with corrupted triples and a rank list is obtained. If the test triplet ranks in the top 10 then it's measure is treated as one , zero otherwise.

2.4.2 *Inference rule accuracy*

Let us consider two relations R1 and R2. Also let us assume that they participate in a transitive relationship of the form :

$$R_1(a,b) \wedge R_2(b,c) \implies R_3(a,c)$$

Here R_3 is an inferred relationship which is derived from R_1 and R_2.

We modelled this relationship R_3 by using composition of relations specific parameters of R_1 and R_2, where linear tensor parameters (A and U) are added and bilinear tensor parameter is multiplied.

In the knowledge base there can be various facts such as Place In State(IIT Guwahati,Assam) $\wedge$ State in Country(Assam,India) which can imply Place in Country(IIT Guwahati,India). These derived logical rules can help in deducing new facts which will help in completing the existing KB. It can also help in optimizing storage space as only the logical rules for these facts need to be stored.[2]

3. DATASET

We have used subset of people/person section of freebase for training and testing our model. Information of each person such as place of birth, gender, siblings, parents, profession, notable types, ethnicity , religion , sports associated etc.[3]

4. IMPLEMENTATION

We have used Theano library of python for modelling of neural network . 70 percent of the Dataset has been used for training and rest for testing part.All the parameters of the model are updated using mini batch stochastic gradient descent with appropriate learning rate for different relations.The number of tensor slices of the relation parameter we have were set to 4. The number of corrupted triples for every training triple are set to 10 . The number of mini batches are set to 10 . The number of training epochs are set to 20.The regularisation parameter is set to 0.0001.

5. RESULTS

5.1 Accuracy over various relations

The accuracy of various relations(with their entity count mentioned in braces) is shown in the following figure.We have observed that if number of training triples are less, then we got varying results for accuracy . We also have observed that as the number of entities increase the accuracy of the model decreases.

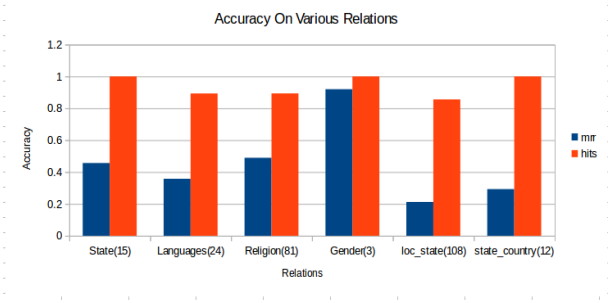

Figure 3: Accuracy over various relations

5.2 Hidden rules accuracy

We have considered the following hidden rule and observed similar results for the actual relation *loc_country* and hidden one.

$loc\_state \wedge state\_country \implies loc\_country$

5.3 Changing the parameters

We took a relation(language) and changed various parameters and observed the following results. Firstly we changed the number of tensor slices and observed the following result as shown in the figure.When the tensor slices were 4 the model might have over fitted with training data and hence we observed lesser accuracy when compared with 2 tensor slices. Then we have selected two tensor slices and changed the count of corrupted triples and observed the following result as shown in the figure. As we decreased the no of corrupted triples , accuracy for MRR is observed to be almost same but HITS@10 decreased considerably. Then we have

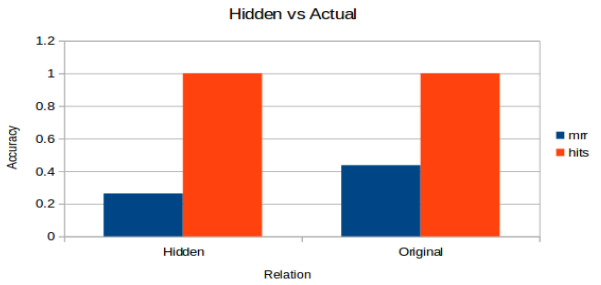

Figure 4: Comparison of hidden relation vs actual relation accuracy

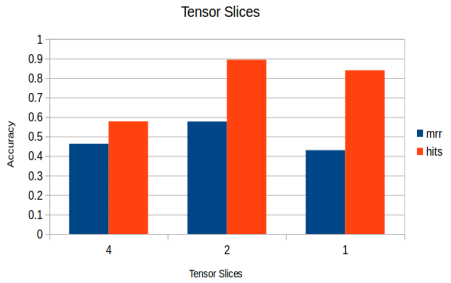

Figure 5: Tensor Slices vs Accuracy

selected 5 as corrupted triples count and observed the following result as shown in the figure . Then finally we have

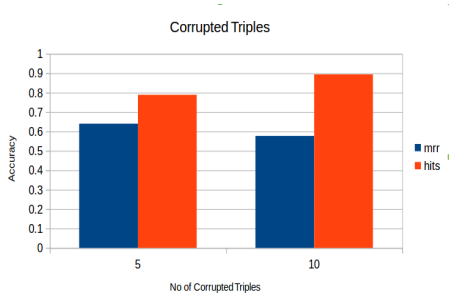

Figure 6: Corrupted triples vs Accuracy

changed the learning rate to obtain the following result as shown in figure.

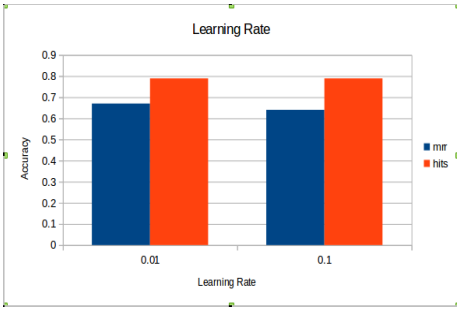

Figure 7: Learning rate vs Accuracy

6. CONCLUSION

We have successfully trained neural network on 10 relations of freebase data. We have verified the accuracy of the neural network using MRR and HITS@10 measures . Also we have been able to model a relation as a composition of existing two relations which will eventually help us to deduce new facts and optimize storage space as only logical rules for these facts need to be stored.

7. REFERENCES

[1] Socher, Richard, Huval, Brody, Manning, Christopher D., and Ng, Andrew Y. Semantic compositionality through recursive matrix-vector spaces.

[2] Embedding entities and relations for learning and inference in knowledge bases by Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao2 and Li Deng.

[3] www.freebase.com/people/person?instances=