Neural Nets and Knowledge Bases

By Kushal Arora
karora@cise.ufl.edu

Why not Deep Learning

- For a model to be called a 'deep model' it must have more than 2-3 layers
- All the models discussed are shallow networks with a single hidden layer.
- The first layer projects the entities(and in some cases relations) in latent/embedding space.
- The second layer then optimizes the objective function to endow certain desirable properties to embedding and learn scoring functions. More on this later

What can it be used for?

- Link Prediction(Internal Expansion)
- Semantic Parsing
- Question Answering?
- Knowledge Base Expansion(External)
- Ontology Alginment?

Problem Formulation

- All the models use a similar general approach with changes in relation representation, entity representation and initialization and loss function definition definition.
- General approach is as follows:
  - Initialize entity (in some cases relation too) embedding in latent space
  - Define a scoring function over triplet.
  - Define a training objective that minimizes the score for valid triplets and maximizes the score for invalid triplets
- Scoring function S should be such that, given a set of triplets X

$$S(e_i^l, r_i, e_i^r) < S(e_j^l, r_i, e_i^r), \forall j : (e_j^l, r_i, e_i^r) \notin X \quad (1)$$

$$S(e_i^l, r_i, e_i^r) < S(e_i^l, r_i, e_j^r), \forall j : (e_i^l, r_i, e_j^r) \notin X \quad (2)$$

Training Objective

- As with the problem formulation, almost all papers follow the similar max-margin training objective defined by [4].
- Let S be the scoring function, T be a valid triplet. We first define a invalid triplet T'

$$T' = (e_j^l, r_i, e_i^r) \vee (e_i^l, r_i, e_j^r)$$

where

$$(e_j^l, r_i, e_i^r) \notin D \wedge (e_i^l, r_i, e_j^r) \notin X$$

- Loss function L can be defined as

$$L = max(0, 1 + S(T) - S(T'))$$

Training

- Let D be the set of all entities, E be $\mathbb{R}^{n \times d}$ matrix of entity embedding .
- Select a positive training triplet at random $x_i \in X$.
- Select at random either constraint type (1) or (2).
  - We select an entity $e^{neg} \in D$ at random and construct a negative training triplet based on chosen constraint.

$$x^{neg} = (e^{neg}, r_i, e_i^r) or x^{neg} = (e_i^l, r_i, e^{neg})$$

- If $S(x_i) > S(x^{neg}) - 1$ then make a gradient step to minimize objective.
- Enforce the constraints that each column $||E_i|| = 1 \forall i$

Semantic Embedding[1]

- Published by Bordes et.al. In 2011.
- First work to use neural energy based methods for embedding
- The objective was to embed KB's in continous space for use by ML models.
- Datasets
  - Freebase (entities with Freebase type deceased people)
  - Wordnet (a subset of relation types)
- Evaluation
  - Mean-Rank for inference
  - Nearest Neighbor for Embedding and Extraction

Semantic Embedding

- Scoring Function
  - Maps left and right entities in a common latent space.
  - Uses L1 distance as a score.

$$S_k(E_i, E_j) = ||R_k^{lhs} E_i - R_k^{rhs} E_j||_p$$

  - E_i, E_j are embedding of i^{th}, j^{th} entity and $E_* \in \mathbb{R}^d$ where d is the dimension of embedding space.
  - Every relation is represented as a tuple $R_k = (R_k^{lhs}, R_k^{rhs})$ and $R_k^{lhs}, R_k^{rhs} \in \mathbb{R}^{dxd}$

Semantic Embedding

- Intuition behind such scoring function is to capture the essence that relations maybe non-symmetric.
- Single embedding used per entity for both subject and object role.
- The biggest drawback of this model is that entities never interact directly which was rectified in other models.
- Applications
  - Inference
  - Entity Embedding to be used in ML related task that use KB's.
  - Knowledge Extraction from Raw Text

		WordNet		Freebase
		rank e^l	rank e^r	rank e^r
COUNTS	*Train*	5.0%	5.0%	0.4%
	Test	0.3%	1.3%	1.7%
EMB	*Train*	76.4%	75.7%	–
	Test	4.0%	4.1%	–
EMB_{MT}	*Train*	83.9%	82.0%	95.8%
	Test	**71.7%**	**76.7%**	14.0%
EMB_{MT}+KDE	*Train*	**88.1%**	**85.8%**	**99.2%**
	Test	64.2%	68.3%	**17.0%**

Precision @ Top 10

Semantic Energy Matching Approach

- First introduced by Bordes et.al in [2] in 2012 and further expanded in [3] .
- Embeds relation too as a vector in latent space, hence useful in open-domain semantic parsing.
- In the formulation of [2] the objective is to map WordNet and text corpus in the same embedding domain and learn an energy function that is small for likely triplets.
- This setting is done to achieve two goals
  - Knowledge Base expansion using raw text
  - Word Sense Disambiguation for the text corpus

Algorithm

- Two Step Process

  Step 1: Structure Inference

  - Uses SENNA toolkit[4] for triplet extraction by applying POS, Chunking, Lemmatization and SRL.
  - Only sentences with structure of (subject, verb, direct object) were considered.
  - These parsed sentence are called lemma.

  Step 2: Entity Detection

  - In this step all-words word-sense disambiguation is done by assigning each word to a synset.

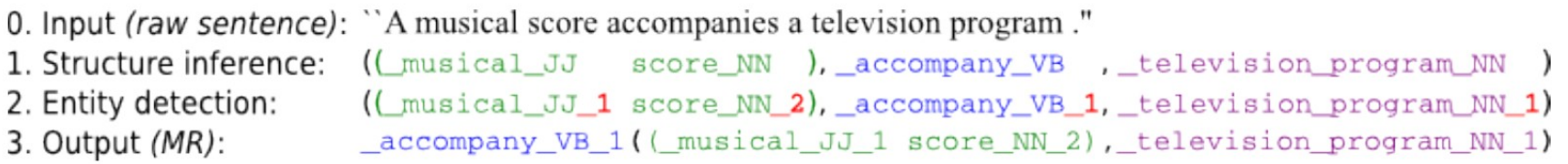


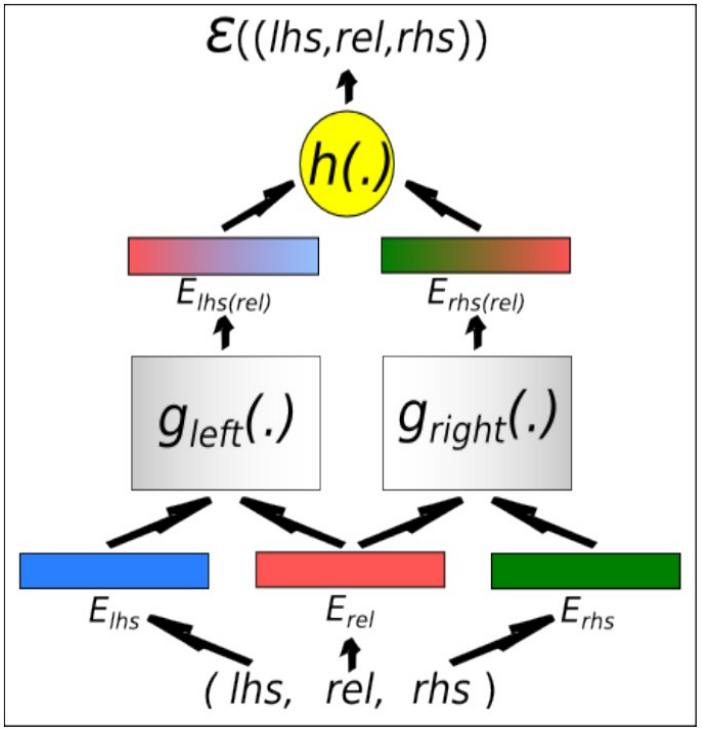
Figure 1: **Open-text semantic parsing.** To parse an input sentence (step 0), a preprocessing (lemmatization, POS, chunking, SRL) is first performed (step 1) to clean data and uncover the MR structure. Then, to each lemma is assigned a corresponding WordNet synset (step 2), hence defining a complete meaning representation (step 3).

Algorithm: Entity Detection

- Both relation and entities are embedded in latent space.
- In first step, a relation dependent embedding $E_{lhs(rel)}, E_{rhs(rel)}$ is calculated for both left and right entities.
- In final step relation dependent right and left embeddings are combined to compute energy
- The paper [] proposes to use dot product for h
- For g_{left}, g_{right} paper uses bilinear form given by

$$E_{lhs(rel)} = (W_{ent,l} E_{lhs}) \otimes (W_{rel,l} E_{rel}) + b_l$$
$$E_{rhs(rel)} = (W_{ent,r} E_{rhs}) \otimes (W_{rel,r} E_{rel}) + b_r$$
$$h(E_{lhs(rel)}, E_{rhs(rel)}) = -E_{lhs(rel)} \cdot E_{rhs(rel)}$$

- The scoring function gives high score to valid triplet hence, we maximize it by changing sign in loss function.



Algorithm: Entity Detection

- The entity detection is done in following way
  - For each lemma $(e_{lhs}^{lemma}, r^{lemma}, e_{rhs}^{lemma})$ of the form extracted from corpus
  - Replace one entity or relation, for example e_{lhs}^{lemma} with a suitable synset.
  - Suitable synset is the one that minimizes the energy of the triplet with substitution. Formally,

$$e_{lhs}^{syn} = argmin_{k \in C(syn|lem)} S((k, r^{lemma}, e_{rhs}^{lemma}))$$

where $C(syn|lem)$ is set of all synsets that can be mapped to e_{lhs}^{lemma}

- As relations too are embedded, it uses an additional constraint, namely

$$S(e_i^l, r_i, e_i^r) < S(e_i^l, r_j, e_i^r), \forall j : (e_i^l, r_j, e_i^r) \notin X(1)$$

Datasets and Results

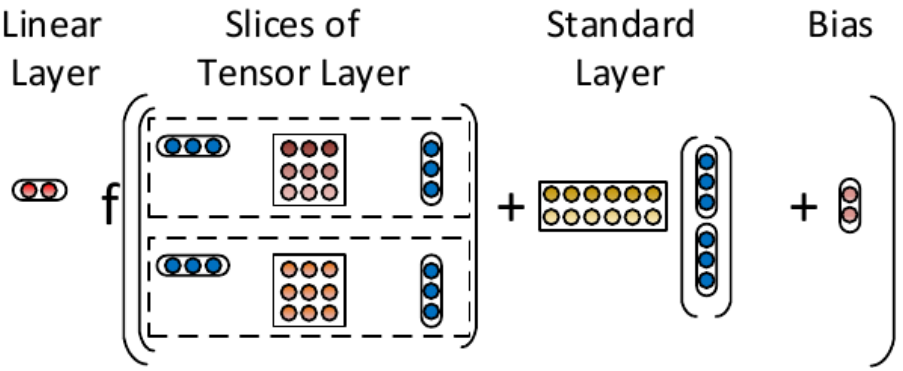
- Uses multi-task training, by training on multiple datasets at the same time.
- Dataset used are
  - WordNet, ConceptNet, Wikipedia (for raw text), Extended WordNet

Table 1: **WordNet Knowledge Acquisition (cols. 2-3) and Word Sense Disambiguation (cols. 4-5).** MFS uses the Most Frequent Sense. **All+MFS** is our best system, combining all sources of information.

Model	WordNet rank	WordNet p@10	F1 XWN	F1 Senseval3
All+MFS	–	–	**72.35%**	**70.19%**
All	139.30	3.47%	67.52%	51.44%
WN+CN+Wk	95.9	4.60%	34.80%	34.13%
WN	72.1	5.88%	29.55%	28.36%
MFS	–	–	67.17	67.79%
Gamble (Decadt *et al.*, 2004)	–	–	–	66.41%
SE (Bordes *et al.*, 2011)	53.2	7.45%	–	–
SE (no KDE) (Bordes *et al.*, 2011)	87.6	4.91%	–	–
Random	20512	0.01%	26.71%	29.55%

Neural Tensor Network(NTN) Approach

- Proposed by Richard Socher et.al. [5], [6] using this approach
- They use this approach for Knowledge Base completion i.e. inferring new facts from existing KB.
- In this formulation, the evaluation is done by predicting the accurate triplet in test test



Neural Tensor Model

Neural Tensor Model

- The scoring function used in case of NTN is more generic one
- Contrary to other models that use projection matrices to model relations, NTN uses a tensor of depth k, which is a parameter to the model.
- A way to look at this is the each splice of tensor model is responsible for a type of entity pair
- For instance, the model could learn that both animals and mechanical entities such as cars can have parts from different parts of the semantic word vector space.
- The scoring function is as follows:

$$\mathrm{S}(\mathrm{e}_{lhs}, r, e_{rhs}) = u_R^T f\left(e_{lhs}^T W_R^{[1:k]} e_{rhs} + V_R \begin{bmatrix} e_{lhs} \\ e_{rhs} \end{bmatrix} + b_R\right)$$

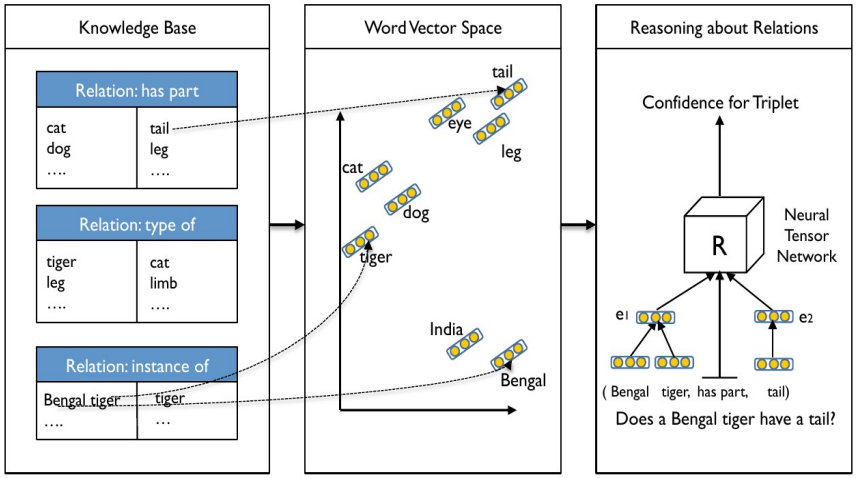
Where W_R, V_R, b_R, U are of the dimension $\mathbb{R}^{d\times d\times k}, \mathbb{R}^{2d\times k}, \mathbb{R}^k, \mathbb{R}^k$ and f is $tanh$.

Entity Representation Approach

- This paper proposes an interesting entity representation approach.
- Entities are represented as mean of vectors of words associated with the entity.
- For example

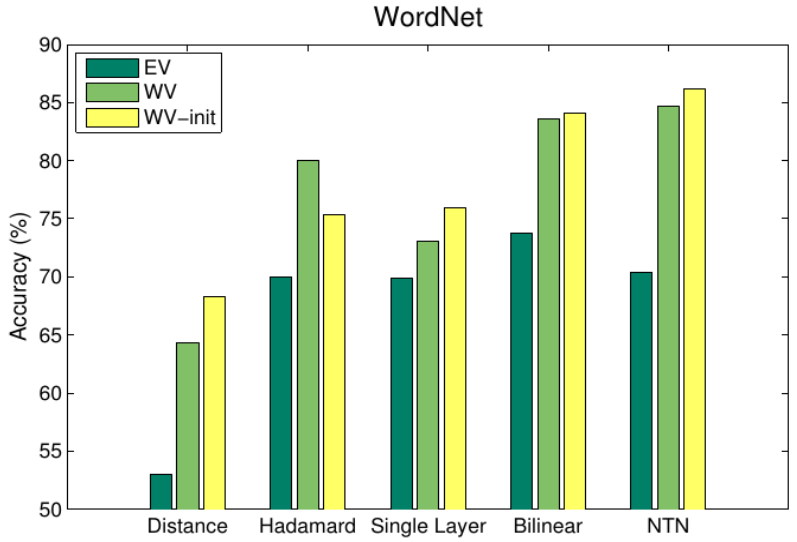
$$\mathrm{E}_{HomoSapiens} = 0.5(V_{Homo} + V_{Sapiens})$$
$$\mathrm{E}_{BengalTiger} = 0.5(V_{Bengal} + V_{Tiger})$$

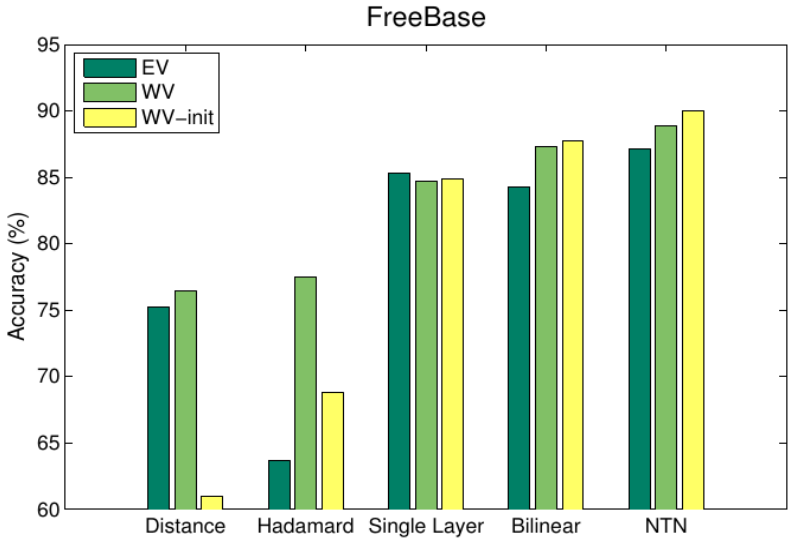
- Pre-trained unsupervised word vectors from [] are used to initialize word vectors.
- This improves generalization as statistical strength of composing words can help in improving confidence for unseen entities
- Ex Bengal Tiger and Sumatran Tiger



Dataset and Results

- Datasets used are
  - Freebase
  - WordNet
- The threshold T_d such that triplet is valid if $S(e_1, r, e_2) > T_d$ is calculated using development set.





EV: Entity Vectors, WV: Word Vectors with random initialization,
WV-Init: Word Vectos initialized with pretrained word vectors

Inferred Examples

Entity e_1	Relationship R	Sorted list of entities likely to be in this relationship
tube	type of	structure; anatomical structure; device; body; body part; organ
creator	type of	individual; adult; worker; man; communicator; instrumentalist
dubrovnik	subordinate instance of	city; town; city district; port; river; region; island
armed forces	domain region	military operation; naval forces; military officier; military court
boldness	has instance	audaciousness; aggro; abductor; interloper; confession;
peole	type of	group; agency; social group; organisation; alphabet; race

Example of right hand side entities inferred from WordNet. Entities are ranked by the assigned score.

References

1. Antoine Bordes, Jason Weston, Ronan Collobert, and Yoshua Bengio.Learning structured embeddings of knowledge bases. In Conference on Articial Intelligence, number EPFL-CONF-192344, 2011.
2. Antoine Bordes, Xavier Glorot, Jason Weston, and Yoshua Bengio. Joint learning of words and meaning representations for open-text semantic parsing. In International Conference on Articial Intelligence and Statistics, pages 127135, 2012.
3. Antoine Bordes, Xavier Glorot, Jason Weston, and Yoshua Bengio. A semantic matching energy function for learning with multi-relational data. Machine Learning, 94(2):233259, 2014.
4. Ronan Collobert and Jason Weston. A unied architecture for natural language processing: Deep neural networks with multitask learning. In Proceedings of the 25th international conference on Machine learning, pages 160167. ACM, 2008.
5. Richard Socher, Danqi Chen, Christopher D Manning, and Andrew Ng. Reasoning with neural tensor networks for knowledge base completion. In Advances in Neural Information Processing Systems, pages 926934, 2013.
6. Danqi Chen, Richard Socher, Christopher D Manning, and Andrew Y Ng. Learning new facts from knowledge bases with neural tensor networks and semantic word vectors. arXiv preprint arXiv:1301.3618, 2013.

Thank You!