Commonsense Knowledge Base Reasoning

Mohammad Asif Khan

- Facts are not stated explicitly in text.

  “If you stick a pin into a carrot, it leaves a hole.”

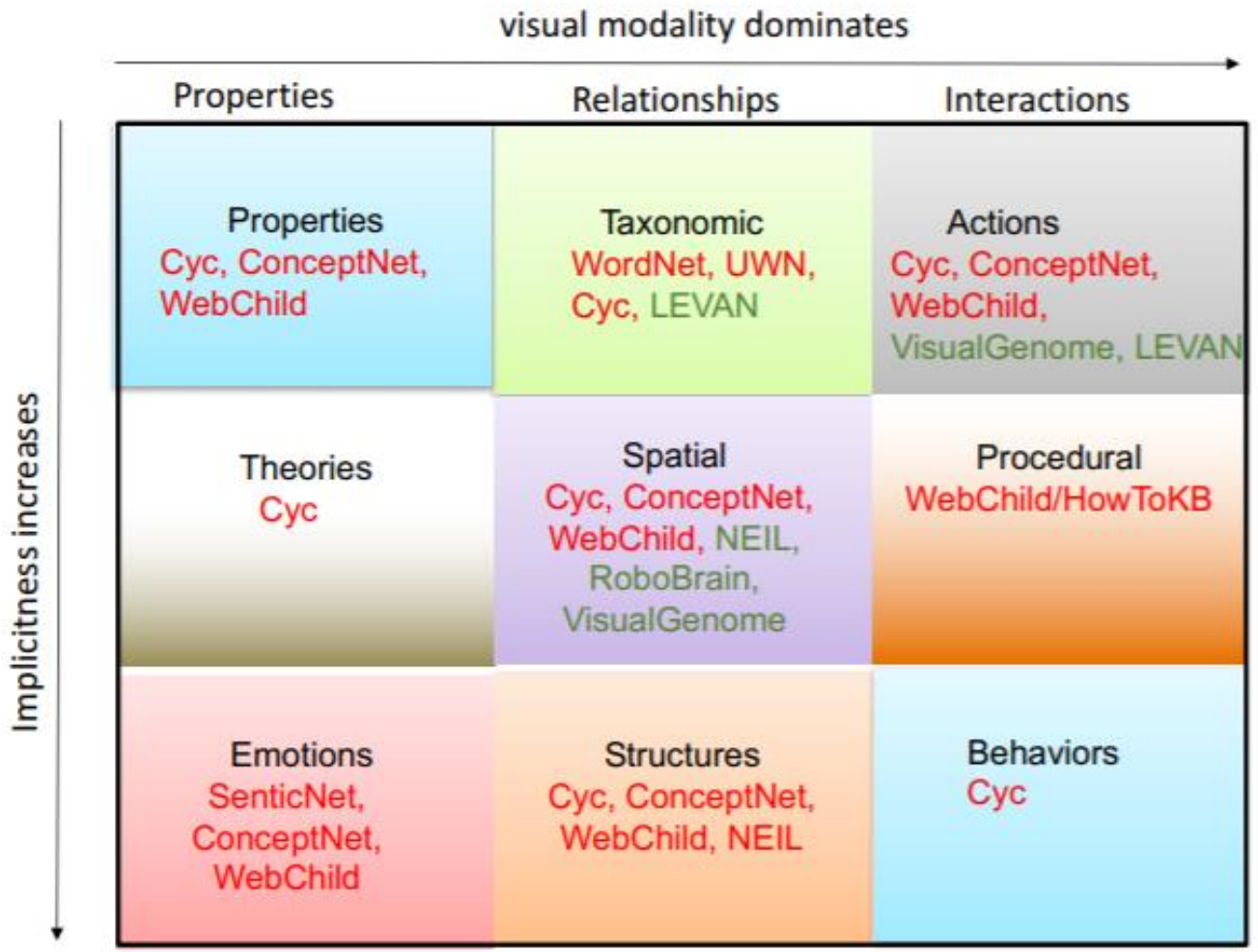
- Facts have to be combined.

  “If X is a father of Y, then Y is a child and X is older than Y.”

Why?

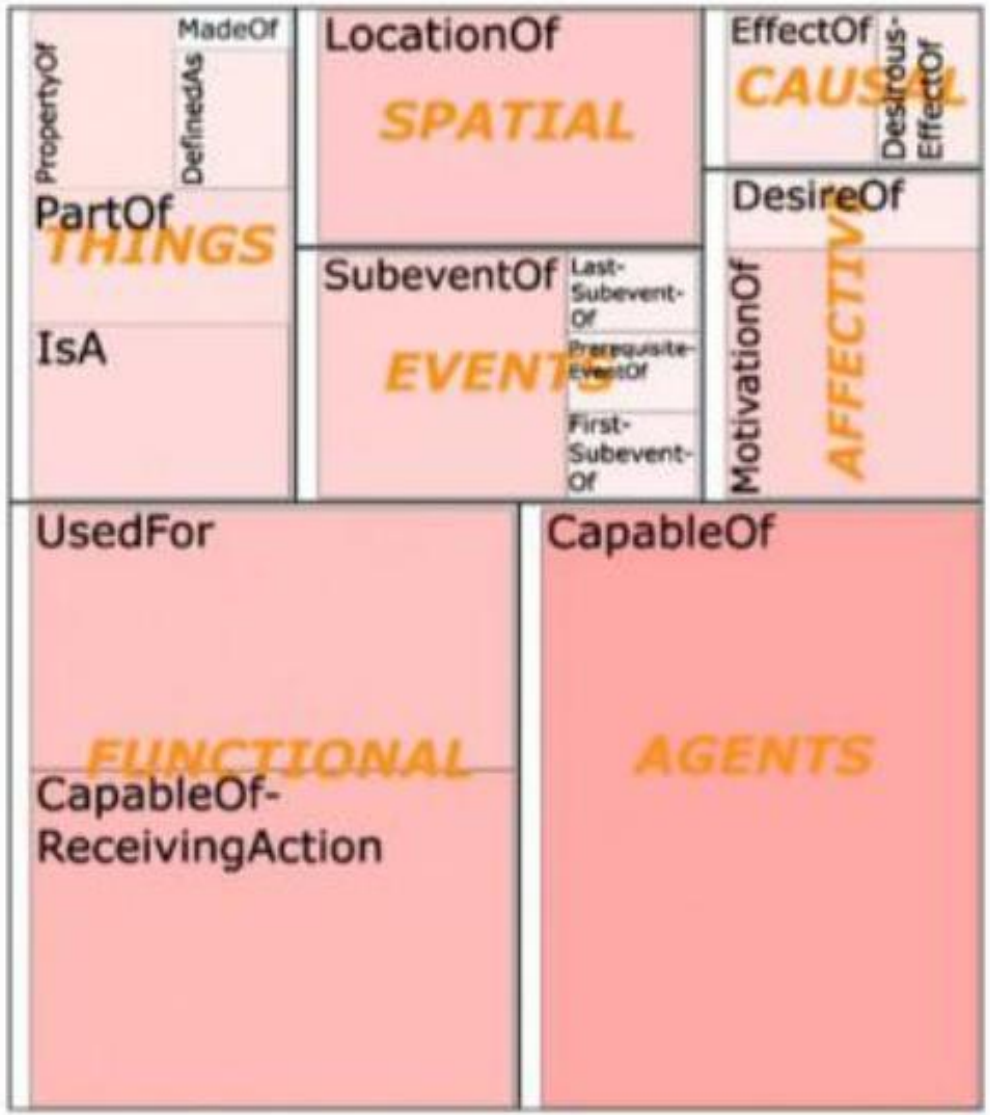
How we can represent it?

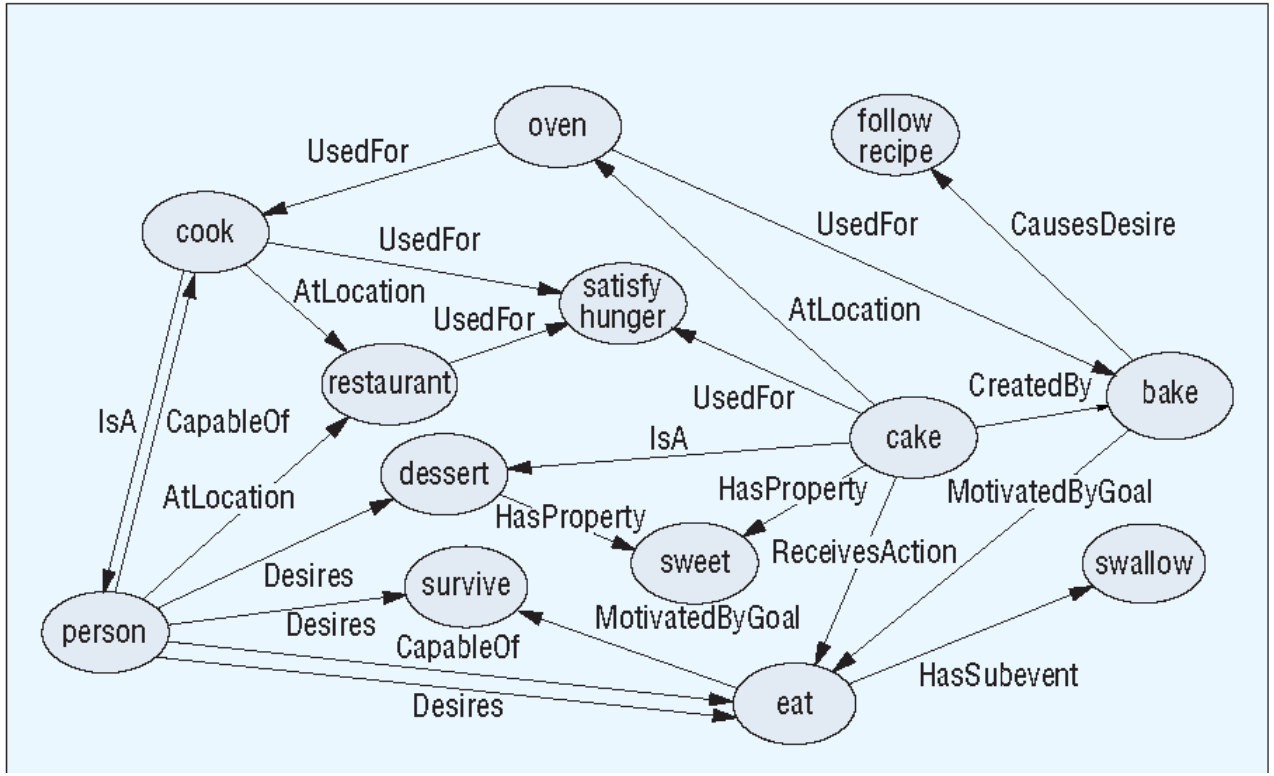
Commonsense KBs



ConceptNet

- Freely available commonsense knowledge base.
- Supports many practical textual reasoning tasks over real-word documents.
- 1.6 million assertions, 34 relation types.




oven
follow recipe
UsedFor
cook
UsedFor
UsedFor
CausesDesire
AtLocation
satisfy hunger
AtLocation
UsedFor
restaurant
CreatedBy
bake
UsedFor
IsA
CapableOf
cake
IsA
dessert
HasProperty
MotivatedByGoal
AtLocation
HasProperty
sweet
ReceivesAction
swallow
Desires
survive
MotivatedByGoal
person
Desires
CapableOf
HasSubevent
eat
Desires

ConceptNet

Project

Commonsense ↔ ConceptNet KB.

Objective: Increase coverage of ConceptNet by inferring missing entries.

How: Score tuples in ConceptNet using KB Completion methods.

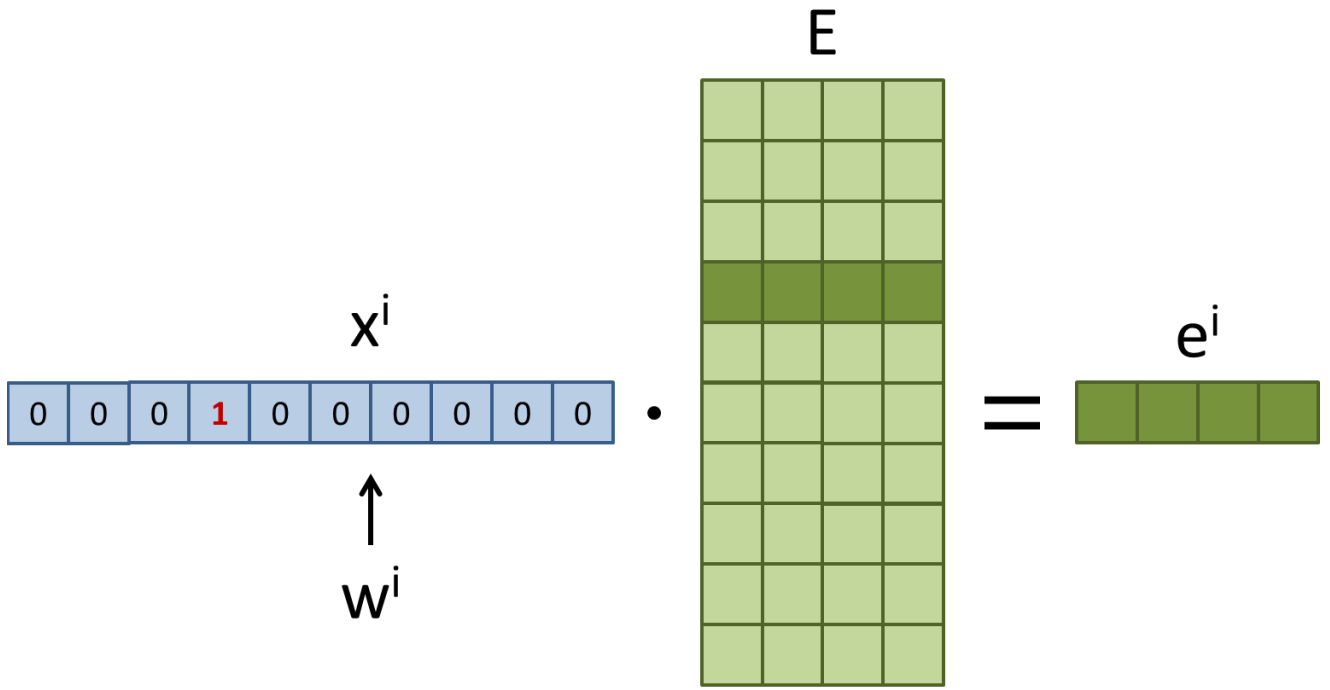
- $\forall\ (h, r, t)\ \epsilon\ ConceptNet$
  - $h\ =\ [word_1, word_2, \ldots, word_n]$
  - $\mathrm{t}\ =\ [word_1, word_2, \ldots, word_m]$

ex:
$(listen\ to\ radio, HasPrerequisite, tune\ in\ station)$

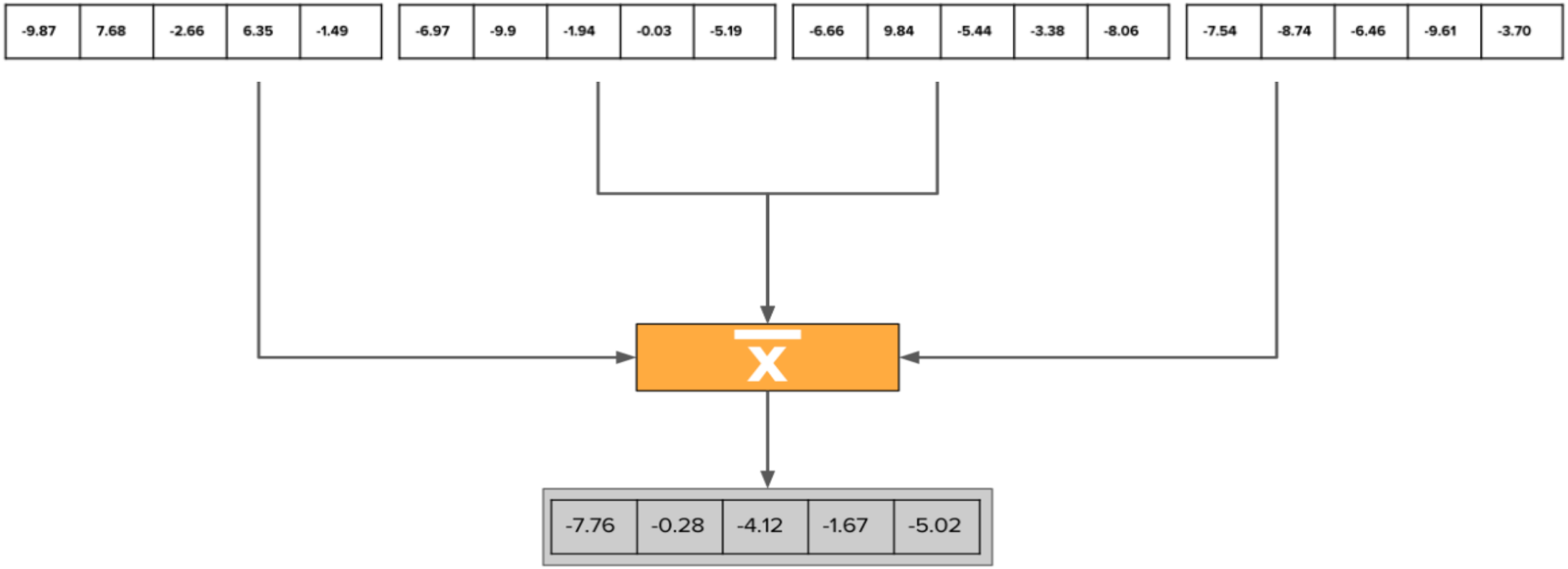
KB Completion: ConceptNet

Bilinear Averaging

- $\forall\ (h, R, t)\ \epsilon\ ConceptNet$
  - $h\ =\ [h_1, h_2, \ldots, h_n]$
  - $\mathrm{t}\ =\ [t_1, t_2, \ldots, t_m]$
  - $\mathrm{R}\ =\ [r\ ]$



Word Averaging

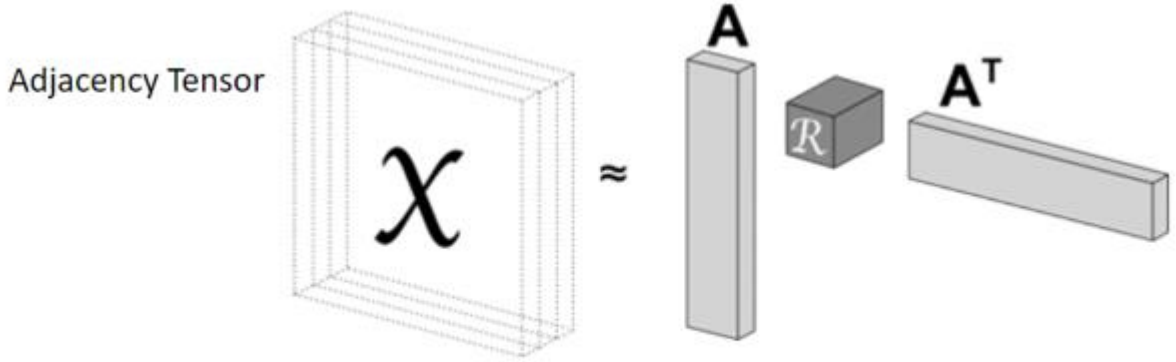


Bilinear Model

$$f_{ijk}^{\text{RESCAL}} := \mathbf{e}_i^\top \mathbf{W}_k \mathbf{e}_j = \sum_{a=1}^{H_e} \sum_{b=1}^{H_e} w_{abk} e_{ia} e_{jb}$$

$$A = [e_i^1, e_i^2, e_i^3, \dots, e_i^n] \in E^m$$

$$W = R = [r_i^j] \in E^{m \times m}$$



- Problem:
  - R grows quadratically with term vectors.

Bilinear Model

Bilinear Model

- Problem:
  - R grows quadratically with term vectors.

- Solution:
  - Apply non-linear transform on each term.
    - $u_i = \tanh(W^{(B)} v_i + b^{(B)})$
    - score $(h, R, t) = u_h^T W_R u_t$

ConceptNet Tuples

```
[['ReceivesAction', 'hockey', 'play on ice', '3.4594316186372978'],
 ['AtLocation', 'restroom', 'rest area', '3.4594316186372978'],
 ['HasPrerequisite', 'eat breakfast', 'make breakfast', '3.4594316186372978'],
 ['AtLocation', 'book', 'table', '3.4594316186372978'],
 ['AtLocation', 'dust', 'refrigerator', '3.4594316186372978'],
 ['UsedFor', 'letter', 'communication', '3.4594316186372978'],
 ['UsedFor', 'dynamite', 'blow thing up', '3.4594316186372978'],
 ['AtLocation', 'shade', 'window', '3.4594316186372978'],
 ['HasA', 'cat', 'eye', '3.4594316186372978'],
 ['AtLocation', 'cat', 'bag', '3.4594316186372978']]
```

Preprocessing

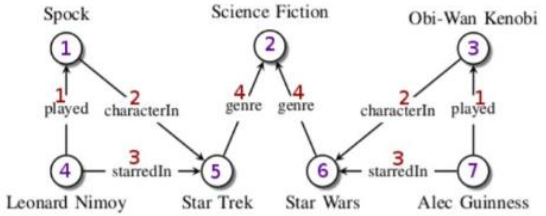
- Word to index

```
[[13, [630], [64, 36, 317]],
 [1, [3333], [391, 314]],
 [5, [80, 434], [75, 434]],
 [1, [105], [166]],
 [1, [1223], [668]],
 [2, [235], [1104]],
 [2, [5351], [1340, 46, 74]],
 [1, [2103], [469]],
 [8, [227], [676]],
 [1, [227], [334]]]
```

Preprocessing

- Padding

```
(array([[ 630,    0,    0, ...,    0,    0,    0],
        [3333,    0,    0, ...,    0,    0,    0],
        [  80,  434,    0, ...,    0,    0,    0],
        ...,
        [  44,    0,    0, ...,    0,    0,    0],
        [1588,  805, 1936, ...,    0,    0,    0],
        [ 189,    0,    0, ...,    0,    0,    0]]),
 array([[   64,    36,   317, ...,     0,     0,     0],
        [  391,   314,     0, ...,     0,     0,     0],
        [   75,   434,     0, ...,     0,     0,     0],
        ...,
        [68029,     0,     0, ...,     0,     0,     0],
        [  332,     9,   805, ...,     0,     0,     0],
        [    3,    54,   738, ...,     0,     0,     0]]),
 array([13,  1,  5, ...,  1,  2,  9]))
```



(e_4, r_1, e_1) exists (e_1, r_1, e_4) does not
(e_1, r_2, e_5) exists (e_1, r_3, e_5) does not
(e_4, r_3, e_5) exists (e_6, r_3, e_5) does not
... and so on

$$BCE = -\frac{1}{N}\sum_{i=0}^{N} y_i \cdot log(\hat{y}_i) + (1 - y_i) \cdot log(1 - \hat{y}_i)$$

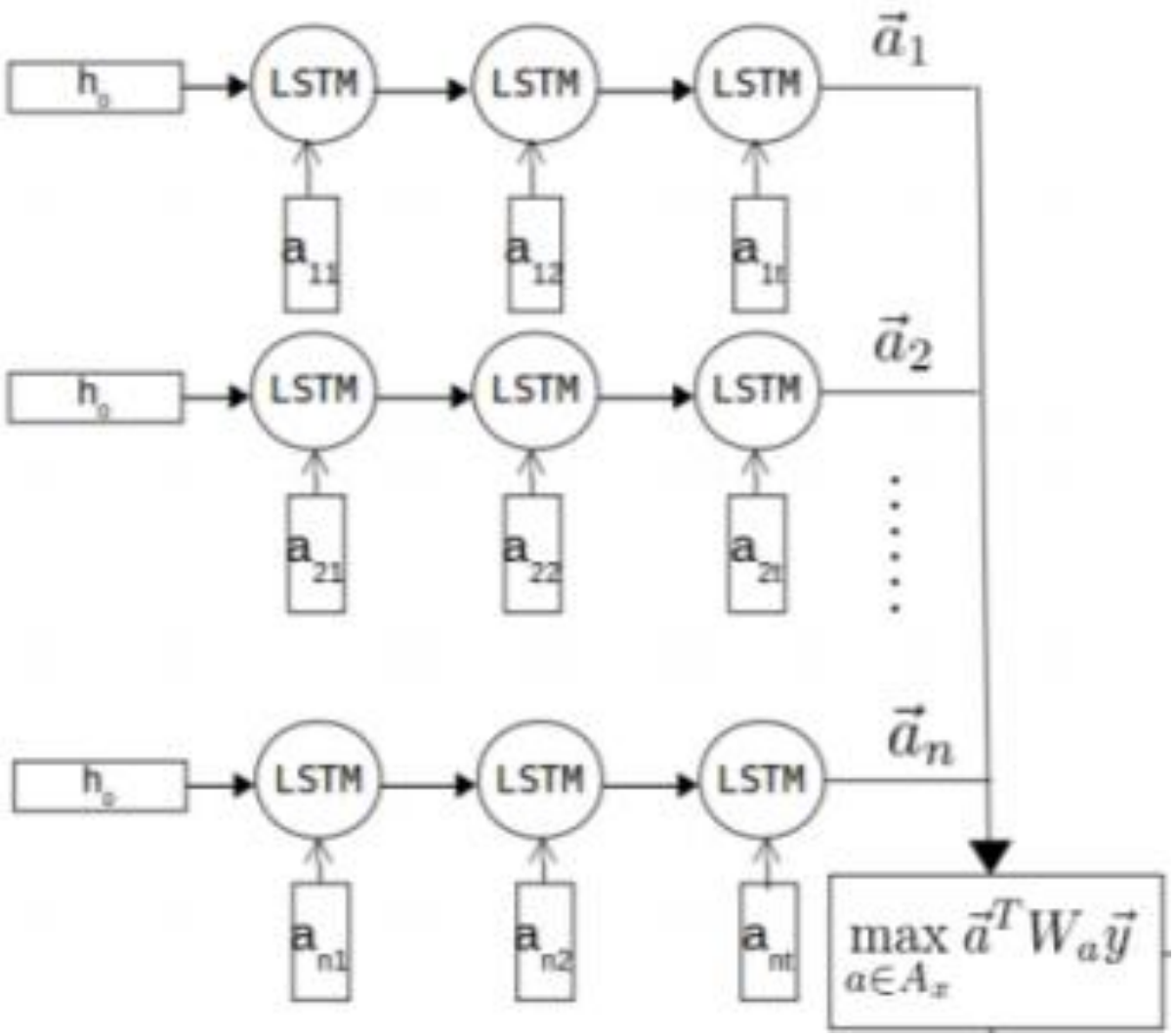
KB Completion as a Triple Classification

Negative Sampling

- Random Sampling
  - For every positive example generate negatives by:
    - randomly perturb head.
    - randomly perturb tail.

Bilinear LSTM

- $\forall\ (h, R, t)\ \epsilon\ ConceptNet$
  - $h\ =\ [h_1, h_2, \ldots, h_n]$
  - $\mathrm{t}\ =\ [t_1, t_2, \ldots, t_m]$
  - $\mathrm{R}\ =\ [r\ ]$



Pre-training

- ConceptNet generated from OMCS.
- Normalize OMCS sentences with ConceptNet relation.

```
['If you want to eat breakfast HasPrerequisite make breakfast'],
['Something you find on book AtLocation table'],
['Something you find behind refrigerator AtLocation refrigerator']
```

- Merge ConceptNet tuples and their OMCS sentences.
- Train word2vec on merged corpus.

KBs Completion.

	Bilinear-Averaging
Validation Accuracy	0.7875
Validation AUC	0.8593
Test Accuracy	0.7941
Test AUC	0.8629

http://ttic.uchicago.edu/~kgimpel/papers/li+etal.acl16.pdf

```
Words in subject found in Vocabulary ['drive', 'fast']
Words in object found in Vocabulary ['accident']
createdby score:  0.173436
symbolof score:  0.163275
haslastsubevent score:  0.149011
causes score:  0.138195
usedfor score:  0.128927
```

```
Words in subject found in Vocabulary ['soak', 'in', 'hotspring']
Words in object found in Vocabulary ['relaxation']
causes score:  0.197455
haslastsubevent score:  0.180322
createdby score:  0.162554
hasprerequisite score:  0.154793
hasfirstsubevent score:  0.152403
```

Relation Discovery

Other Models

- DistMult Averaging.
- DistMult LSTM.
- ER-MLP Averaging.
- ER-MLP LSTM.

Implementation

- PyTorch

  https://github.com/MdAsifKhan/NLP-Project

Usage

The implementation is structured as follows.

1. `model.py`

Contains implementation of different neural networks for scoring a tuple. Currently we provide following models:

- Bilinear Averaging Model
- Bilinear LSTM Model
- DistMult Averaging Model
- DistMult LSTM Model
- ER-MLP Averaging Model
- ER-MLP LSTM Model

2. `utils.py`

Implementation of preprocessing class, negative sampling and other basic utilities. Main class and functions:

- class preprocess

Implements method to read arbitrary phrase ConceptNet triples and convert them to token representation for training neural network models.

- function sample_negatives

Implements negative sampling strategy. Sampling is done by alternatively corrupting head and tail of a triple.

- class TripleDataset

Data class to support with pytorch batch loader.

3. `evaluation.py`

Contains implementation of different evaluation metric. For this project we mainly use accuracy and auc score.

4. `pretrained_embedding.py`

The scoring of tuple is highly dependent on initial embeddings used for training. To help model to better capture commonsense knowledge of ConceptNet we use pretraining. We create training data by combining ConceptNet tuples with natural language sentences of Open Mind Common Sense.

Summary

Commonsense important for general intelligence.

ConceptNet: the largest freely commonsense database.

Common sense enriched word embeddings, can directly be used for applications like entity resolution, pos tagging etc.

Questions.

Thank you! ☺

References

- http://web.media.mit.edu/~minsky/E6/eb6.html
- https://en.wikipedia.org/wiki/Commonsense_reasoning
- https://cacm.acm.org/magazines/2015/9/191169-commonsense-reasoning-and-commonsense-knowledge-in-artificial-intelligence/fulltext
- http://ttic.uchicago.edu/~kgimpel/papers/li+etal.acl16.pdf
- https://people.cs.umass.edu/~xiangl/improved-representation-learning.pdf
- http://web.media.mit.edu/~lieber/Publications/Understanding-Stories-Commonsense.pdf
- https://www.microsoft.com/en-us/research/wp-content/uploads/2016/06/Roth-Dan_CS-NLU.pdf
- https://github.com/commonsense/conceptnet5/wiki
- http://agents.media.mit.edu/projects/makebelieve/AAAI2002-makebelieve.pdf