StarSpace: Embed All The Things

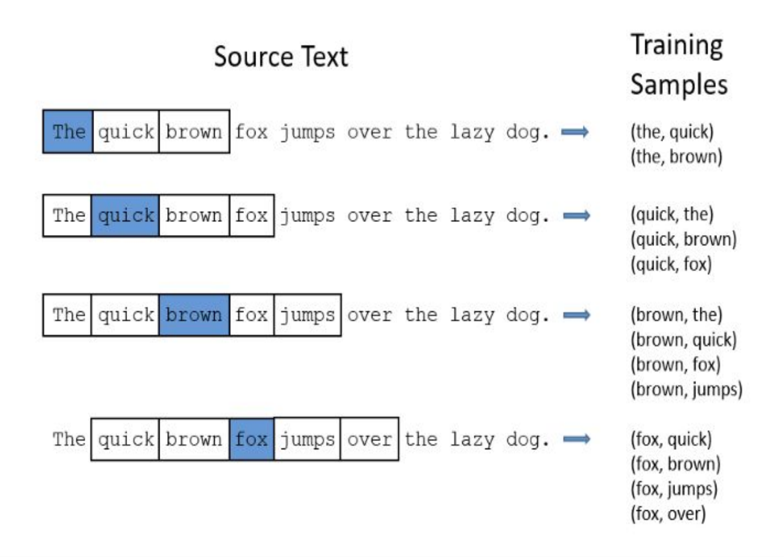
Jan Pichl

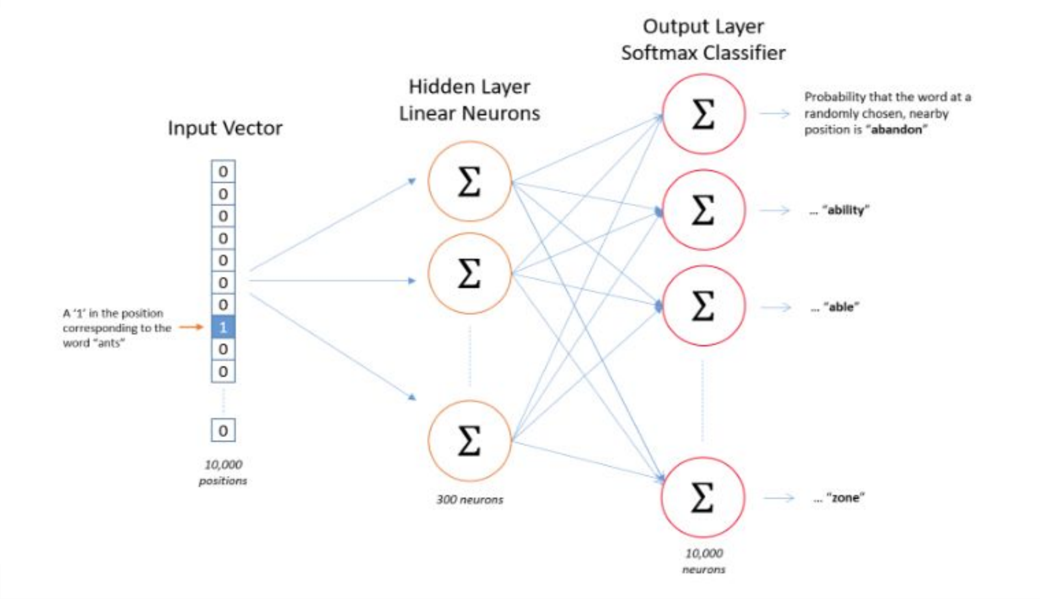
Motivation

- Dimensionality reduction
- Incorporating additional knowledge
- One embedding for multiple problems:
    - Text classification, ranking, image labelling, embedding of graphs (words, sentences, documents)
- One embedding for various types of entities
    - Comparing different types of entities - sentences, graphs, images, words, ...

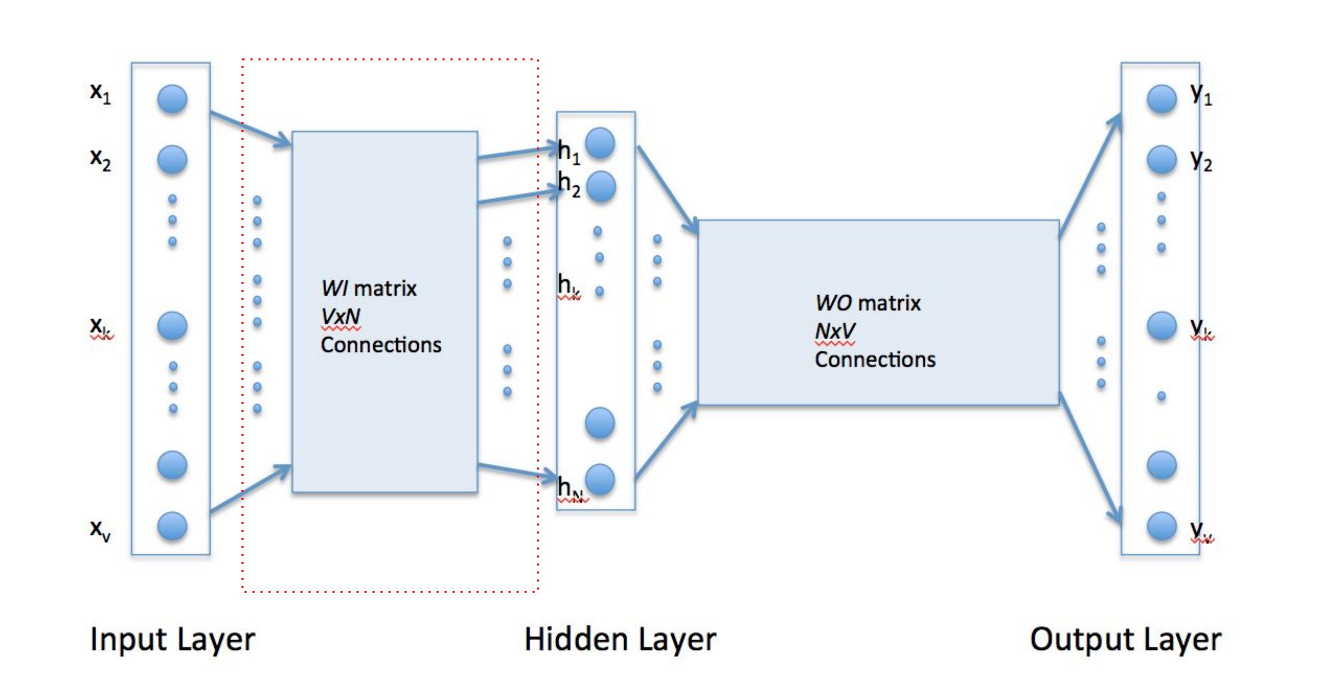
Unsupervised method for words - word2vec

- We need to embed a word into a lower dimensional space
- Skip-gram neural network
- Arithmetic operations show some interesting relations





Word2vec in matrix form



- The trained matrix is used as a lookup table

Generalization

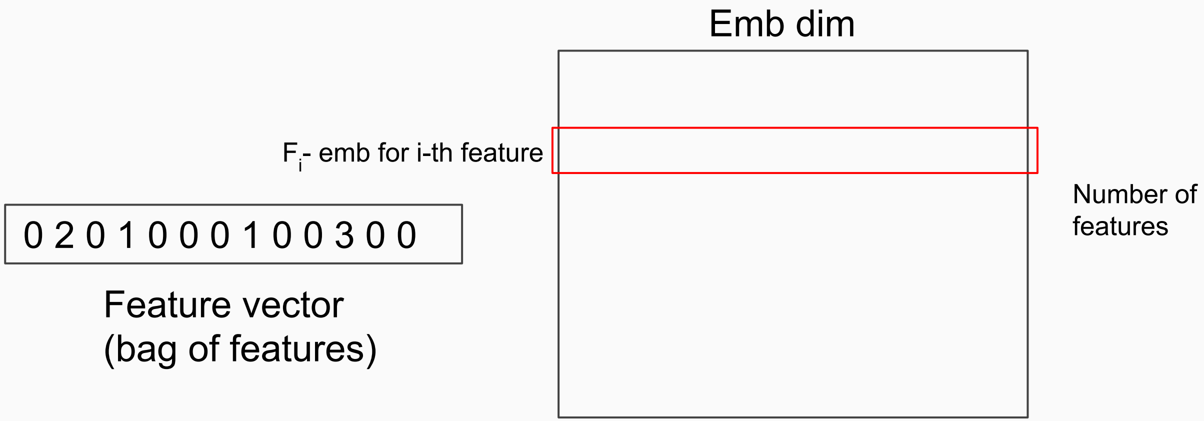
- We need to embed other entities than words
- We need to embed different kinds of entities into a same vector space
- Compare embeddings vectors of different kind of entities
- Train a lookup table similar to a word2vec
- Using a supervised task to train the representation

StarSpace

- Facebook research
- Star (*) Space - embedding for “everything” -> Star
- Main idea
  - Transform discrete features to a vector of real numbers
  - Use one simple approach for different task (avoid computationally expensive neural networks)
- Directly trained on a desired task or...
- The model can be used as an input for neural network for different tasks

StarSpace

- Input: discrete feature vector
- Each entity consist of one or more features (bag of features)
- Embedding matrix: D x d (number of features x desired dimensionality)
- Each row of a matrix is the embedding for one feature
- Entity with multiple features - sum of embeddings
- Fixed feature dictionary



Optimization

- Similarity function: cosine similarity
- Loss function:

$$\sum_{\substack{(a,b)\in E^+ \\ b^-\in E^-}} \underbrace{L^{batch}}_{\text{ranking loss}}(sim(a,b), sim(a,b_1^-), \ldots, sim(a,b_k^-))$$

E+: positive entity pairs
E-: negative entity
(negative sampling)

Embed of entity

- dot product
- cosine sim

- Positive and negative sample
- Task specific spample generation

Training

- Stochastic Gradient Descent
- Adagrad
- Each SGD step contains one positive sample
- Margin ranking loss:
  - $\max\{0, \mu - sim(a, b) + sim(a, b^{-})\}$

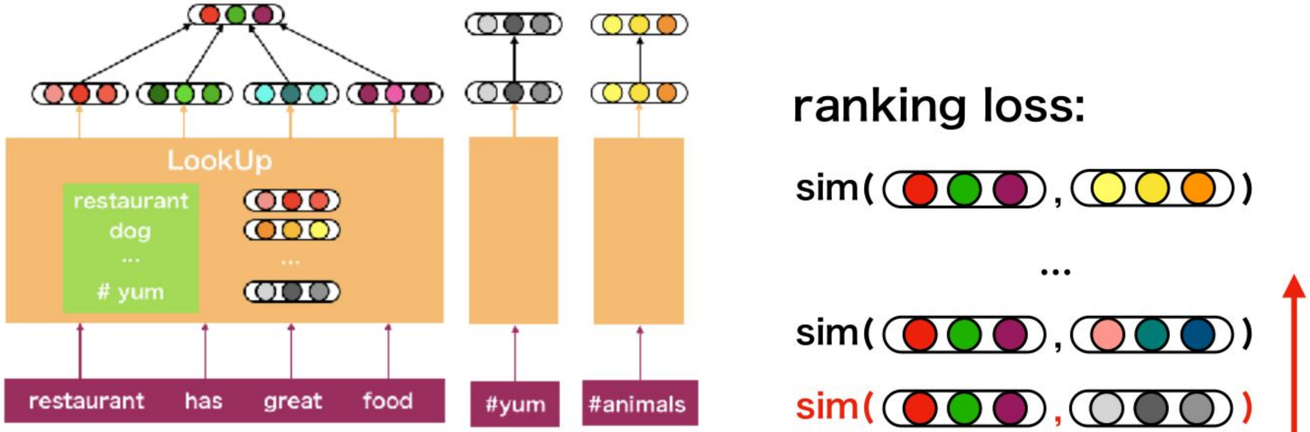
- Positive examples need to be ranked higher than negative ones
- Each batch contains k negative examples
- Testing
  - Direct usage of similarity function
  - Embedding as an input for another model

Tasks

- Classification:
  - Positive: training set
  - Negative: labels sampled from set of labels
- Collaborative Filtering-based Recommendation:
  - Positive: user ID embedding OR list of items users like except one, label - the left out item
  - Negative: sampled from possible items
- Content-based Recommendation:
  - User: bag of documents, Document: bag of words
  - Same as before: one left out
- Multi-Relational Knowledge Graphs:
  - Graph (head, relation, tail) e.g. (Beyonce, born-in, Houston)
  - Randomly remove head or tail -> removed item as a label

Example

E+: {('restaurant has great food', '#yum #restaurant')}
E-: {'#animals', '#donald trump'}



The higher the ranking of the relevant item the better

In Testing:

for a given input a, using $\max_{\hat{b}} sim(a, \hat{b})$

Tasks

- Document Search:
  - Input: query keywords, Output: document bag of words
  - Unsupervised data: random keywords from document
  - Negative: irrelevant documents
  - Implicit learning of document embeddings
- Word embeddings:
  - Input: windows of words (n words on both sides of selected word)
  - Output: selected word
- Sentence embeddings:
  - Positive pairs: sentences from the same document
  - Negative: from different documents
- Image labelling:
  - Input: ResNet (or another model) image features
  - Output: Image label
  - Negative: irrelevant labels

Number of k negative samples

- Dataset - collection of Freebase triplets
- Removed head and tail respectively
- Prediction of missing entity
- Reported how many entities were ranked among first ten (hit@10)
- Best result on this task: k = 50

K	1	5	10	25	50	100	250	500	1000
Epochs	3260	711	318	130	69	34	13	7	4
hit@10	67.05%	68.08%	68.13%	67.63%	69.05%	66.99%	63.95%	60.32%	54.14%

Table 5: Adapting the number of negative samples k for a 50-dim model for 1 hour of training on Freebase 15k.

Results

- Tasks: select sentence from Wikipedia article, try to find the article it came from
- Original article has sentences as a features (minus the original one)

Metric	Hits@1	Hits@10	Hits@20	Mean Rank	Training Time
Unsupervised methods					
TFIDF	56.63%	72.80%	76.16%	578.98	-
fastText (public Wikipedia model)	18.08%	36.36%	42.97%	987.27	-
fastText (our dataset)	16.89%	37.60%	45.25%	786.77	40h
Supervised method					
SVM Ranker BoW features	56.73%	69.24%	71.86%	723.47	-
SVM Ranker: fastText features (public)	18.44%	37.80%	45.91%	887.96	-
StarSpace	56.75%	78.14%	83.15%	122.26	89h

Results

- Tasks: select two sentences from a single Wikipedia article
- Based on the one sentence we want to try the other one

Metric	Hits@1	Hits@10	Hits@20	Mean Rank	Training Time
Unsupervised methods					
TFIDF	24.79%	35.53%	38.25%	2523.68	-
fastText (public Wikipedia model)	5.77%	14.08%	17.79%	2393.38	-
fastText (our dataset)	5.47%	13.54%	17.60%	2363.74	40h
StarSpace (word-level training)	5.89%	16.41%	20.60%	1614.21	45h
Supervised methods					
SVM Ranker BoW features	26.36%	36.48%	39.25%	2368.37	-
SVM Ranker: fastText features (public)	5.81%	12.14%	15.20%	1442.05	-
StarSpace (sentence pair training)	30.07%	50.89%	57.60%	422.00	36h
StarSpace (word+sentence training)	25.54%	45.21%	52.08%	484.27	69h

Thank you!

Bibliography

[1] WU, Ledell, et al. StarSpace: Embed All The Things!. *arXiv preprint arXiv:1709.03856*, 2017.

https://www.slideshare.net/akihikowatanabe3110/starspace-embed-all-the-things