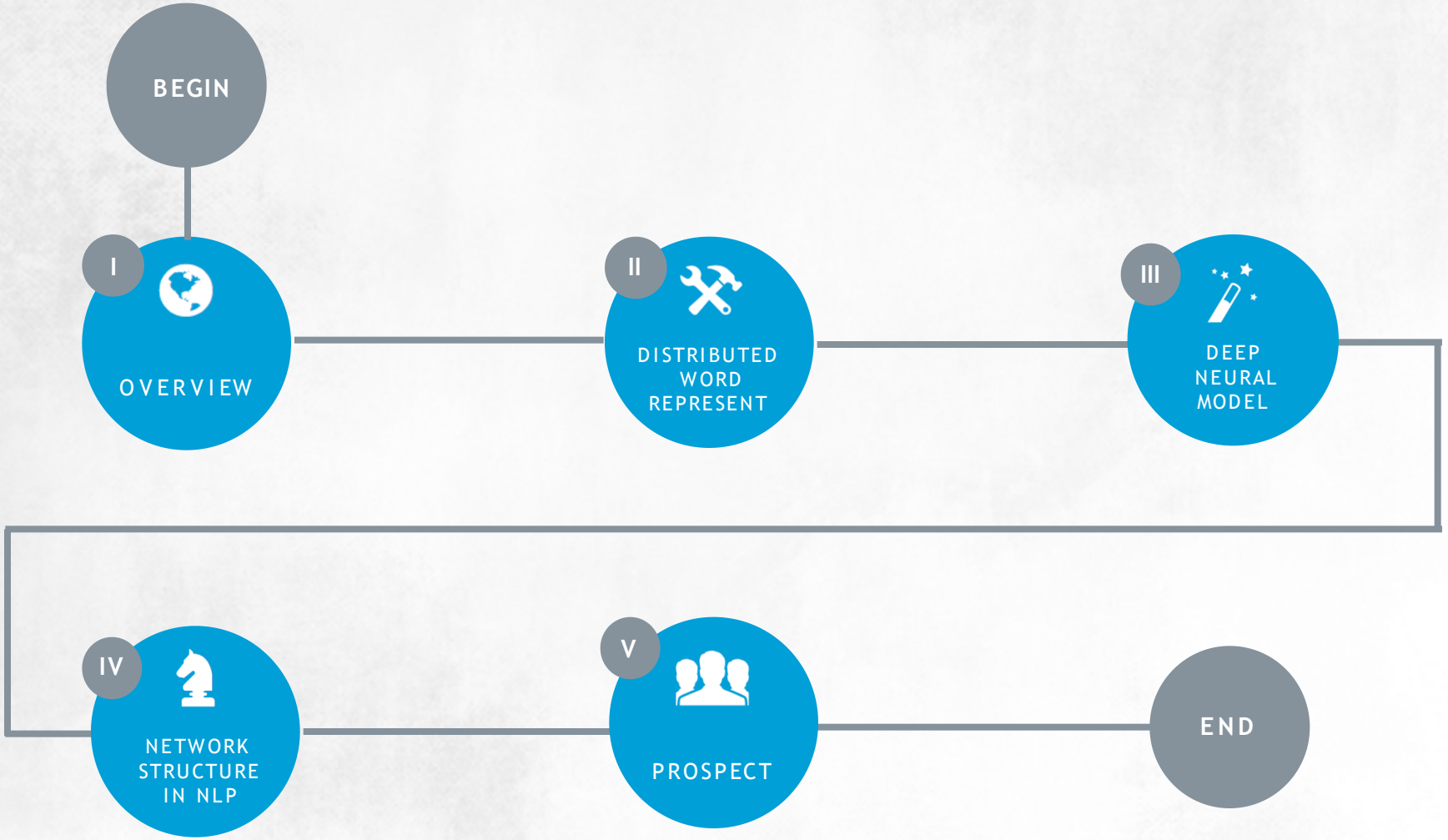
# Part 1

# Topics On Deep Learning In NLP

Hailin WANG

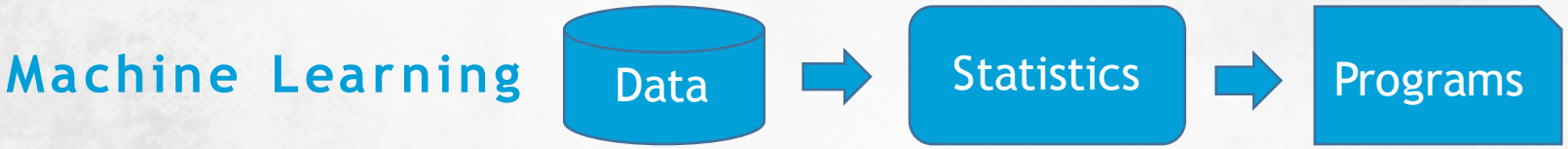
@Fido

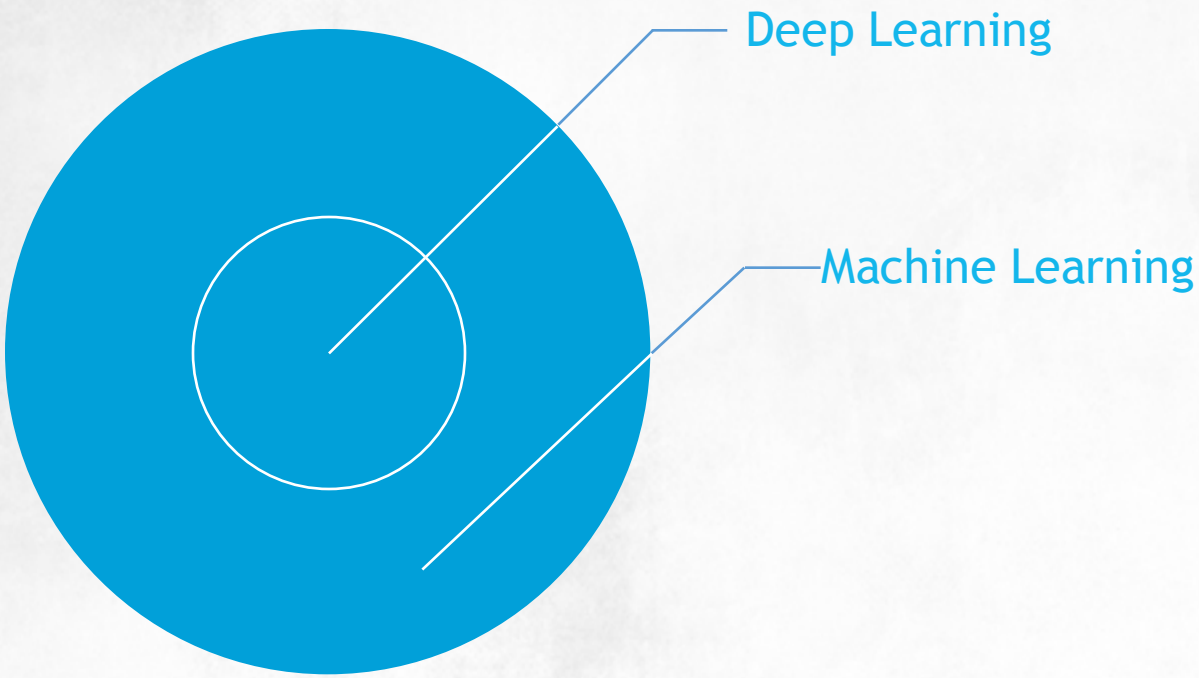
# TALK OUTLINE



# OVERVIEW

# What's Deep Learning



# What's special about Language

Deep learning has been concentrate on the analysis of natural, raw signals:

- Object detection
- Bird calls
- Detecting Cancer

Principally, any meaning is in the eyes of the beholder, based on the analysis of the signal

# What's special about Language

A human language is

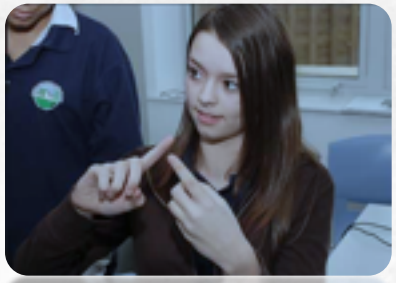
- instead a system specifically constructed to convey the speaker/writer's meaning
- a **discrete/symbolic/categorical** signaling system
  - With very minor exceptions for expressive signaling ("I loooove it." "Whoomppaaa" "啊啊啊啊")
  - Presumably because of greater information-theoretic signaling reliability
  - Symbols are not just an invention of classical AI

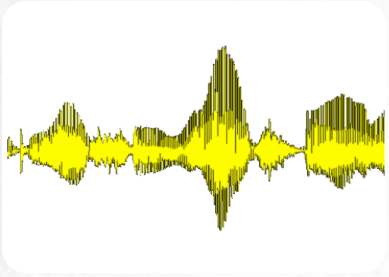
# What's special about Language

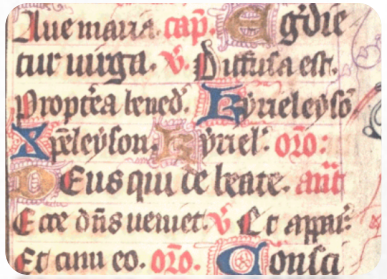
The categorical symbols of a language can be encoded as a signal for communication in several ways:

- Sound
- Gesture
- Images(writing)

**The symbol is invariant** is invariant across different encodings!





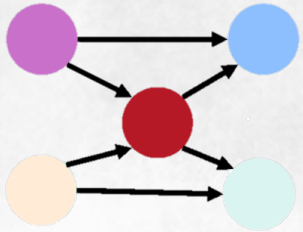


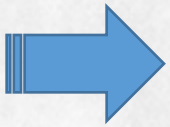
# What's special about Language

A human language is a **discrete, symbolic, categorical signaling system**

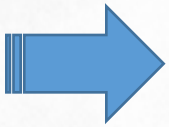
- Despite brain encoding as a continuous pattern of activation and transmission via continuous signals of sound/vision

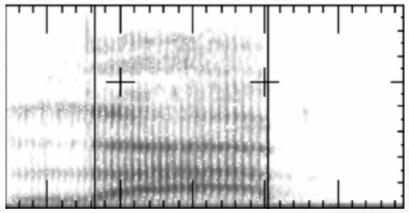
High-dimensional, symbolic encoding of words creates a problem for neural network processing – **sparsity**





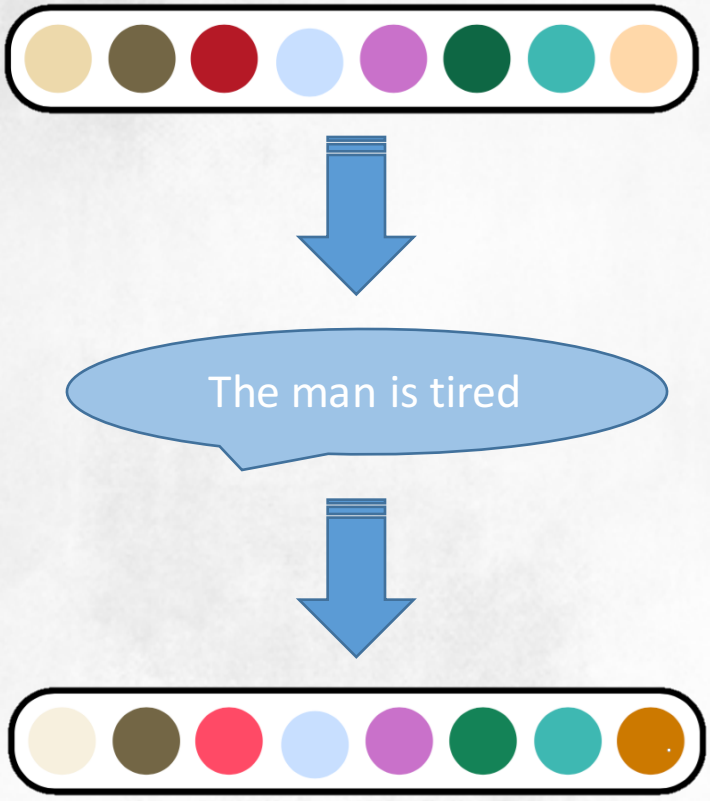
Hello



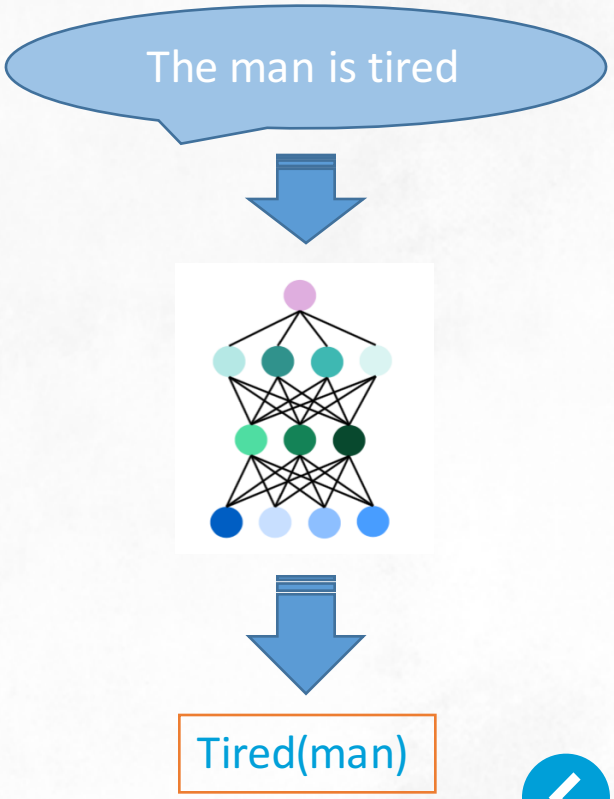


# From symbolic to distributed and back

## Working of our brains




## Working of our systems



# DISTRIBUTIONAL WORK REPRESENTATION

# From symbolic to distributed

The vast majority of rule-based **and** statistical NLP work regarded words as atomic symbols:

[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0]

One-hot representation

# From symbolic to distributed

Its problem:

- If user searches for [Dell notebook battery size], we would like to match with "Dell laptop battery capacity"
- If user searches for [Seattle motel], we would like to match documents containing "Seattle hotel"

But

$$\text{Motel}[0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0]^T$$

$$\text{Hotel}[0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0]=0$$

Our query and document vectors are orthogonal

There is no natural notion of similarity.

# Distributional Similarity-based **Representations**

You can get a lot of value by representing a word by means of its neighbors.

*"You shall know a word by the company it keeps"*

*(J. R. Firth 1957: 11)*

One of the most successful ideas of modern NLP

| | | |
|---|---|---|
| government debt problems turning into saying that Europe needs unified | banking banking | crises as has happened in regulation to replace the hodgepodge |

These words will represent banking

# Basic idea of learning
# Neural Word Embeddings

Definition:

$$choose\ \ argmax_w\ w \cdot (\frac{w_{j-1} + w_{j+1}}{2})$$

which has a loss function, e.g.,

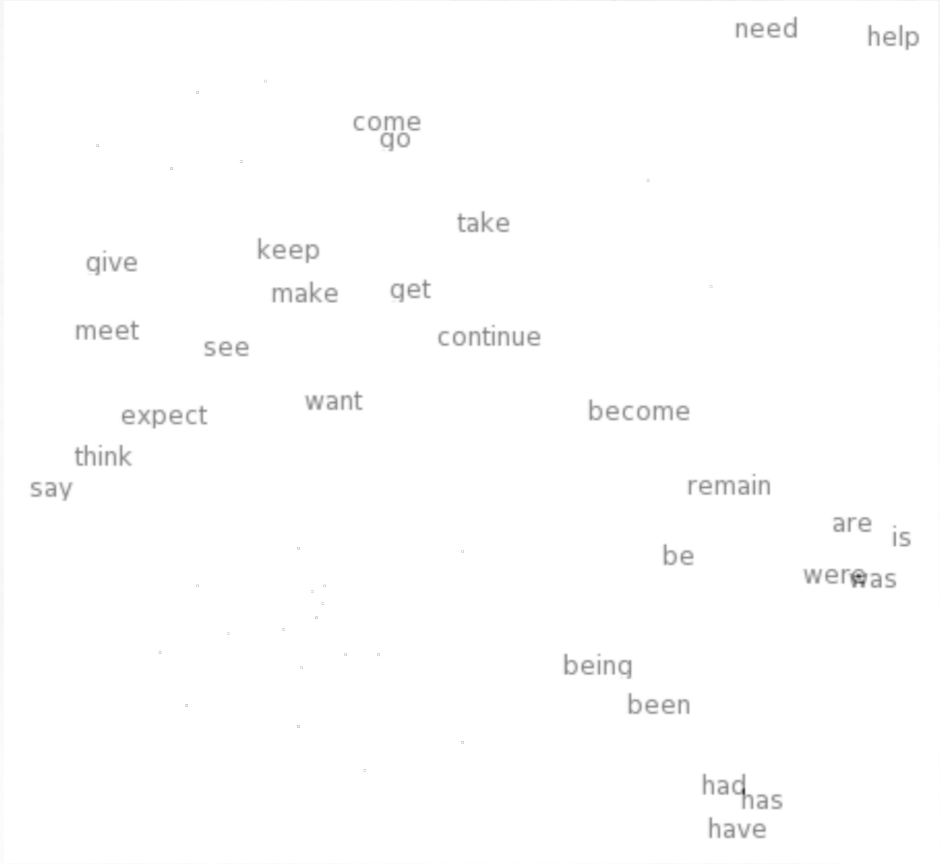
$$J = 1 - w_j \cdot (\frac{w_{j-1} + w_{j+1}}{2})$$

Unit norm vectors

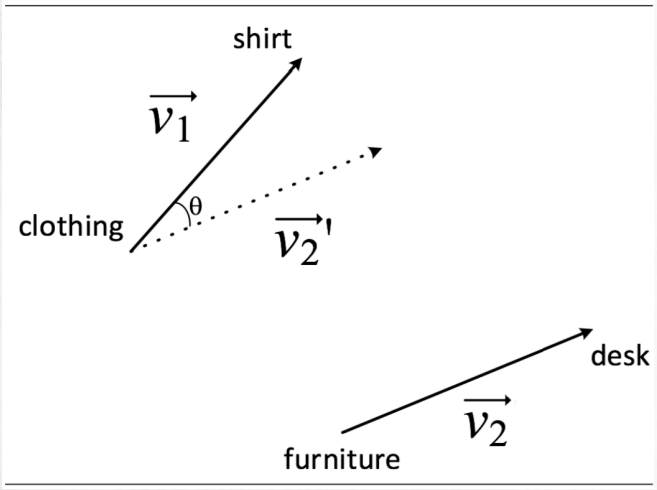
looking at many samples from a big language corpus and adjusting the vector representations of words to minimize this loss

# Basic idea of learning Neural Word Embeddings

$$take = \begin{pmatrix} 0.286 \\ 0.792 \\ -0.177 \\ -0.107 \\ 0.109 \\ -0542 \\ 0.349 \\ 0.271 \end{pmatrix}$$

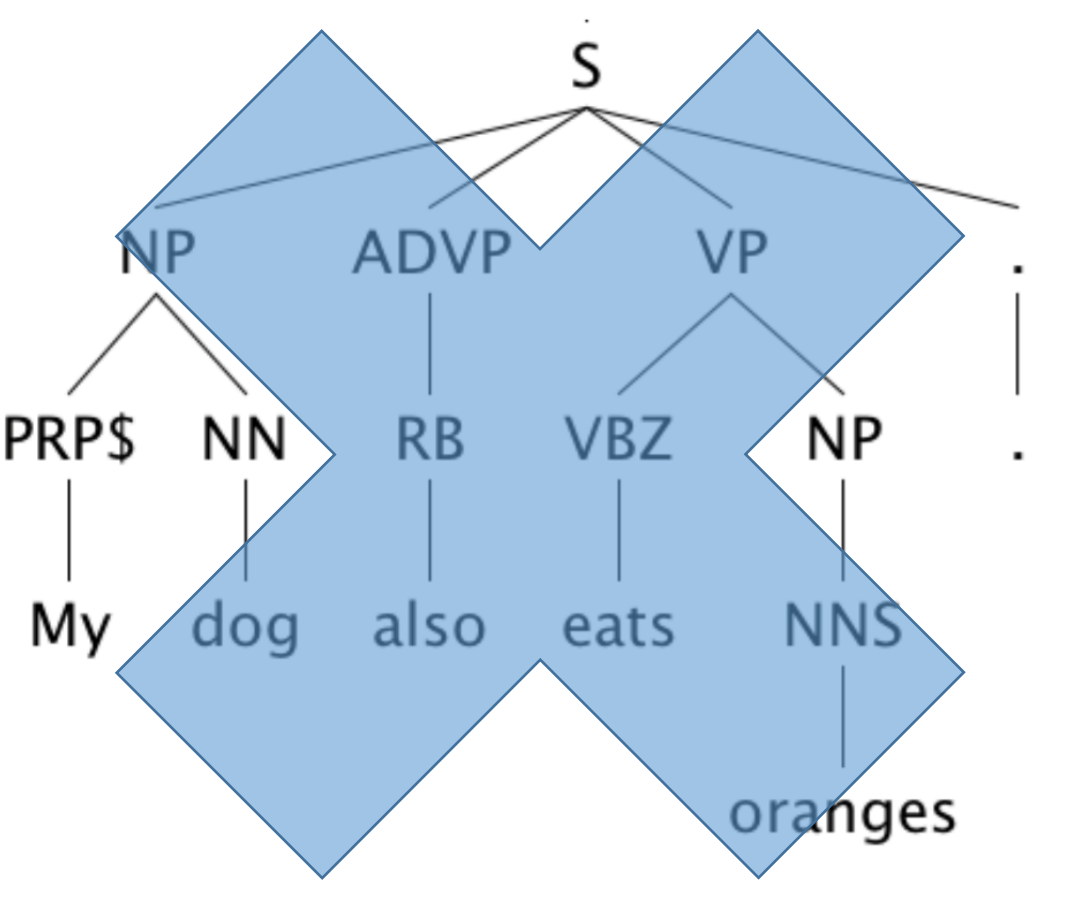


# Basic idea of learning Neural Word Embeddings




| Method | Syntax % correct |
|---|---|
| LSA 320 dim | 16.5 [previous best] |
| RNN 80 dim | 16.2 |
| RNN 320 dim | 28.5 |
| RNN 1600 dim | 39.6 |
| **Method** | **Semantics Spearm ρ** |
| UTD-NB (Rink & H. 2012) | 0.230 |
| LSA 640 | 0.149 |
| RNN 80 | 0.211 |
| RNN 1600 | 0.275 |

# Fragility of NLP tools



# Neural Embeddings

- Using word vectors is not deep learning
- Why mention it as intro:
  - The **first stage** of **most** work in Deep Learning NLP is mapping word symbols to distributed vector representations
  - This is often either so useful or so easy that people stop there (Everywhere at ACL 2015)
  - Neural embeddings aren't that different to other distributed representations.

# Application of Neural Word Embedding: Word2Vec

Two Algorithms:

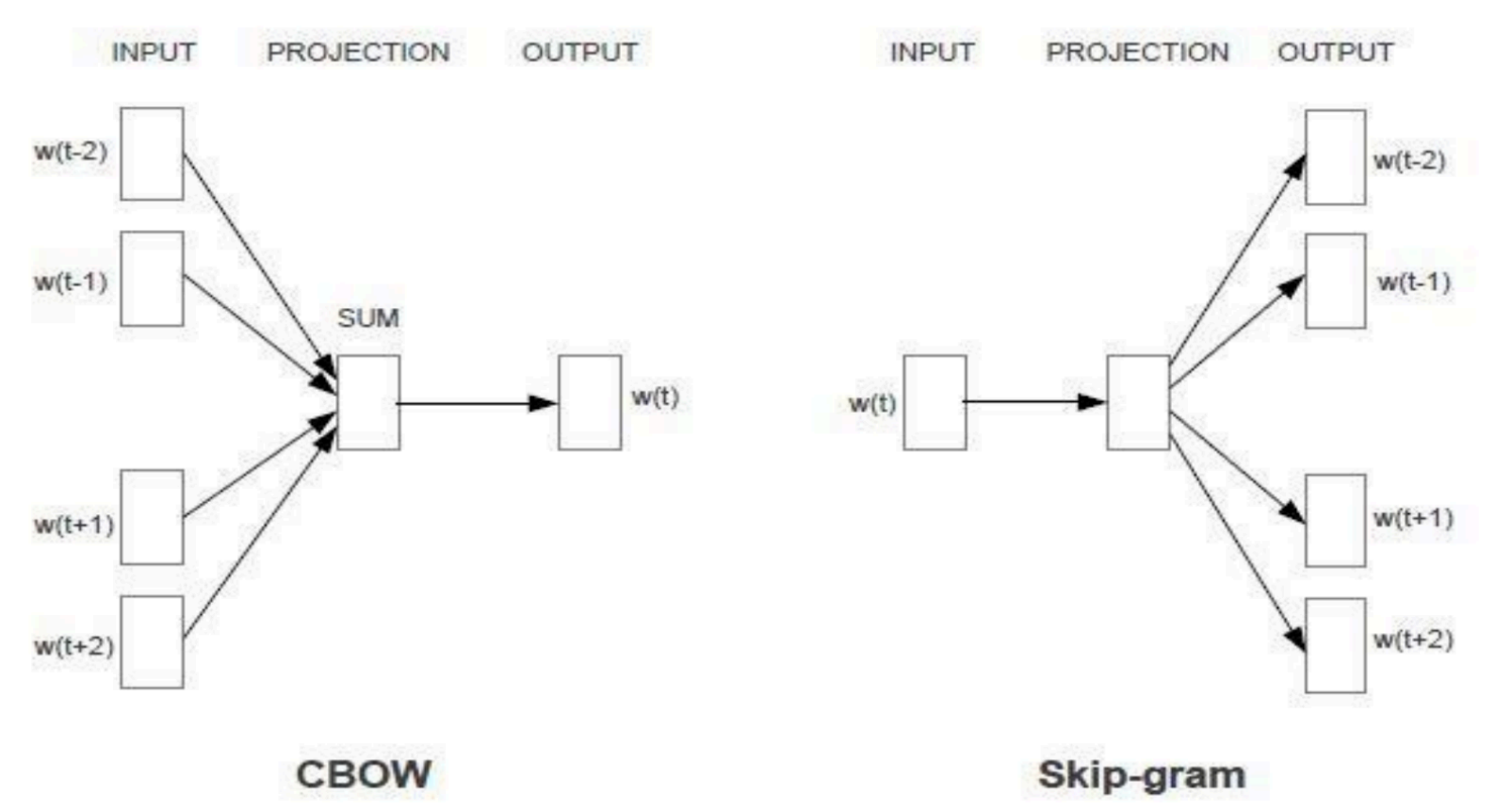
- Skip-grams
  - Predict context words given target (position independent)
- Continuous Bag of Words
  - Predict target word from bag-of-words context

Two training methods:

- Hierarchical soft max
- Negative sampling

Insights: matrix factorization

# Application of Neural Word Embedding: Word2Vec



# Application of Neural Word Embedding: Word2Vec

| FRANCE | JESUS | XBOX | REDDISH | SCRATCHED | MEGABITS |
|---|---|---|---|---|---|
| AUSTRIA | GOD | AMIGA | GREENISH | NAILED | OCTETS |
| BELGIUM | SATI | PLAYSTATION | BLUISH | SMASHED | MB/S |
| GERMANY | CHRIST | MSX | PINKISH | PUNCHED | BIT/S |
| ITALY | SATAN | IPOD | PURPLISH | POPPED | BAUD |
| GREECE | KALI | SEGA | BROWNISH | CRIMPED | CARATS |
| SWEDEN | INDRA | PSNUMBER | GREYISH | SCRAPED | KBIT/S |
| NORWAY | VISHNU | HD | GRAYISH | SCREWED | MEGAHERTZ |
| EUROPE | ANANDA | DREAMCAST | WHITISH | SECTIONED | MEGAPIXELS |
| HUNGARY | PARVATI | GEFORCE | SILVERY | SLASHED | GBIT/S |
| SWITZERLAND | GRACE | CAPCOM | YELLOWISH | RIPPED | AMPERES |

What words have embeddings closest to a given word?
From Collobert*et al.* (2011)

# Application of Neural Word Embedding: Word2Vec

| Relationship | Example 1 | Example 2 | Example 3 |
|---|---|---|---|
| France - Paris | Italy: Rome | Japan: Tokyo | Florida: Tallahassee |
| big - bigger | small: larger | cold: colder | quick: quicker |
| Miami - Florida | Baltimore: Maryland | Dallas: Texas | Kona: Hawaii |
| Einstein - scientist | Messi: midfielder | Mozart: violinist | Picasso: painter |
| Sarkozy - France | Berlusconi: Italy | Merkel: Germany | Koizumi: Japan |
| copper - Cu | zinc: Zn | gold: Au | uranium: plutonium |
| Berlusconi - Silvio | Sarkozy: Nicolas | Putin: Medvedev | Obama: Barack |
| Microsoft - Windows | Google: Android | IBM: Linux | Apple: iPhone |
| Microsoft - Ballmer | Google: Yahoo | IBM: McNealy | Apple: Jobs |
| Japan - sushi | Germany: bratwurst | France: tapas | USA: pizza |

Relationship pairs in a word embedding.
From Mikolov *et al.* (2013b).

# Encoding meaning in vector differences: Glove

Insights: Ratios of co-occurrence probabilities can encode meaning components

| | $x$ = solid | $x$ = gas | $x$ = water | $x$ = random |
|---|---|---|---|---|
| $P(x\|\text{ice})$ | large | small | large | small |
| $P(x\|\text{steam})$ | small | large | large | small |
| $\frac{P(x\|\text{ice})}{P(x\|\text{steam})}$ | large | small | ~1 | ~1 |

[Pennington, Socher, and Manning, EMNLP 2014]

# Comparison

LSA,HAL(Lund & Burgess) ,

COALS(Rohde et al) ,

Hellinger-PCA(Lebret & Collobert)

- Fast training
- Efficient usage of statistics
- Primarily used to capture word similarity
- Disproportionate importance given to small counts

NNLM, HLBL, RNN, word2vec Skip-gram/CBOW, Glove (Bengio et al; Collobert & Weston; Huang et al; Mnih & Hinton; Mikolov et al; Mnih & Kavukcuoglu)

- Scales with corpus size
- Inefficient usage of statistics
- Generate improved performance on other tasks
- Can capture complex patterns beyond word similarity

# Word embedding: other work

**Can one better explain word2vec's linear structure?**

Arora, Li, Liang, Ma, & Risteski (2015) Random Walks on Context Spaces: Towards an Explanation of the Mysteries of Semantic Word Embeddings.

# Word embedding: word senses

Most words have multiple senses

- 走：离开、走路；吃土：穷、饿；呵呵：语气词、 
- Environmental effect.

Most current models just give one vector per string

- Vector is (weighted) average ("superposition") of all the senses

Just a little work has tried to model multiple senses per word

- Mikolov et al. (2013)
- Mnih et al. (2013)

# Word embedding:
# Semantic v.s. Synthetic

SEMANTIC

Carrying-carried,carry

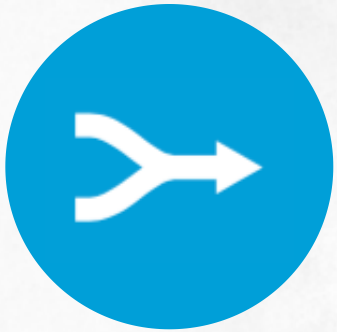
SYNTACTIC

Carrying-transporting

# What's more?

DOWN TO SUB-WORD

- Sub-word embedding
- Character embedding



UP TO SENTENCE/PARA

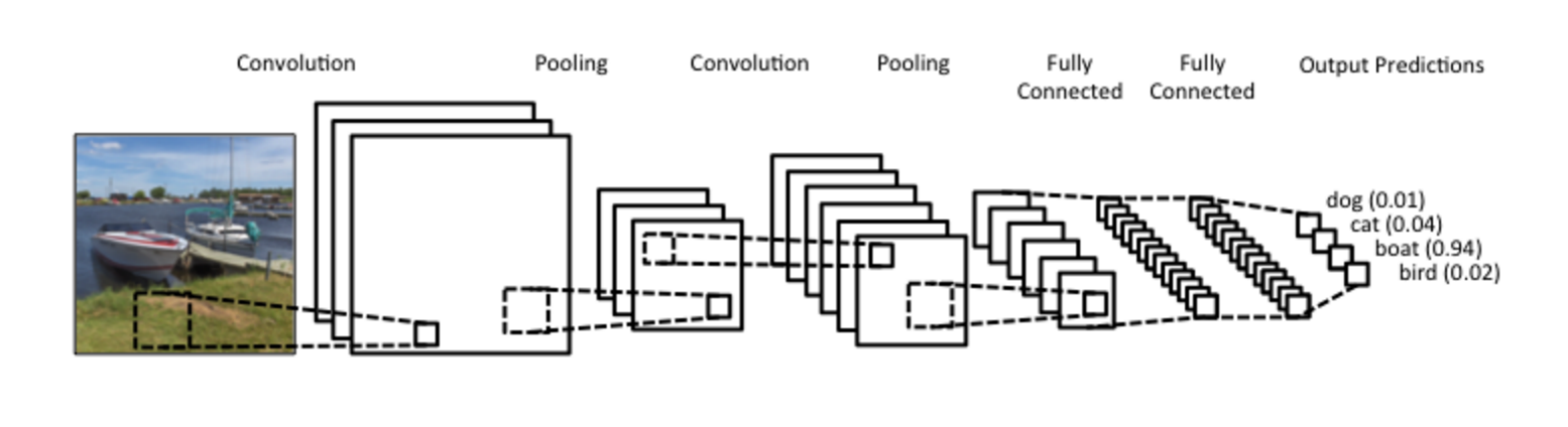
- Paragraph Vector
- Input to DL

# DEEP NEURAL STRUCUTRE

# Recap: **popular DNN**

- Feed Forward NN
- Convolutional NN
- Autoencoders
- RBM
  - DBN
- Recurrent NN
  - LSTM
  - GRU

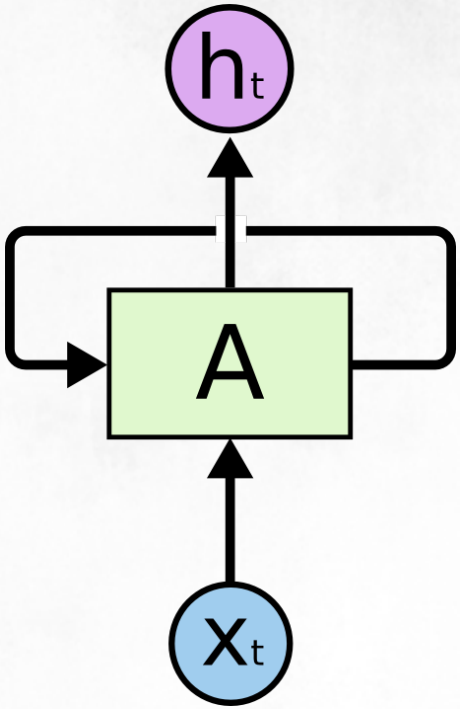
# Recap: CNN

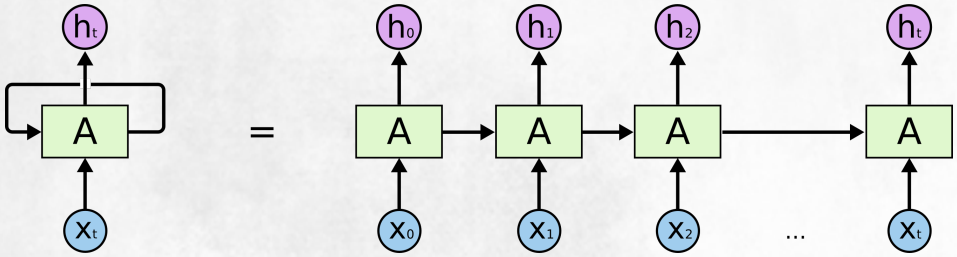



- Idea
  - Partially connected NN
  - convolution operation(Mathematically)
- Method
  - Convolution
  - Pooling
- Feature
  - **Location Invariance** and **Compositionality**

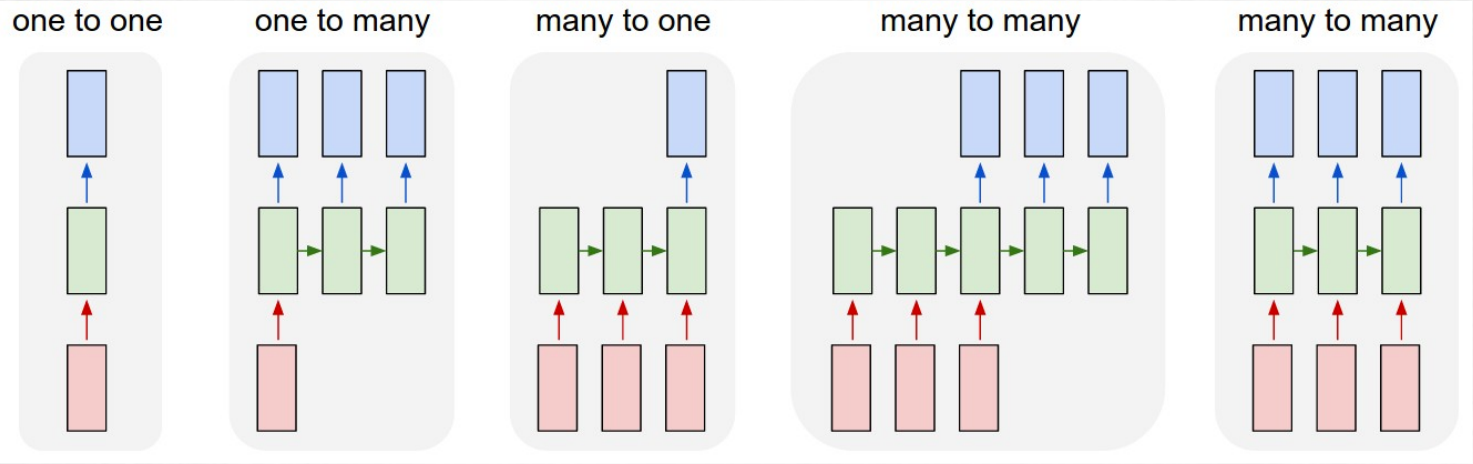
# Recap: RNN

- Idea
  - Thought has persistence.
- Method
  - Networks with loops in them
- Feature
  - Allowing information to persist.



# Recap: RNN

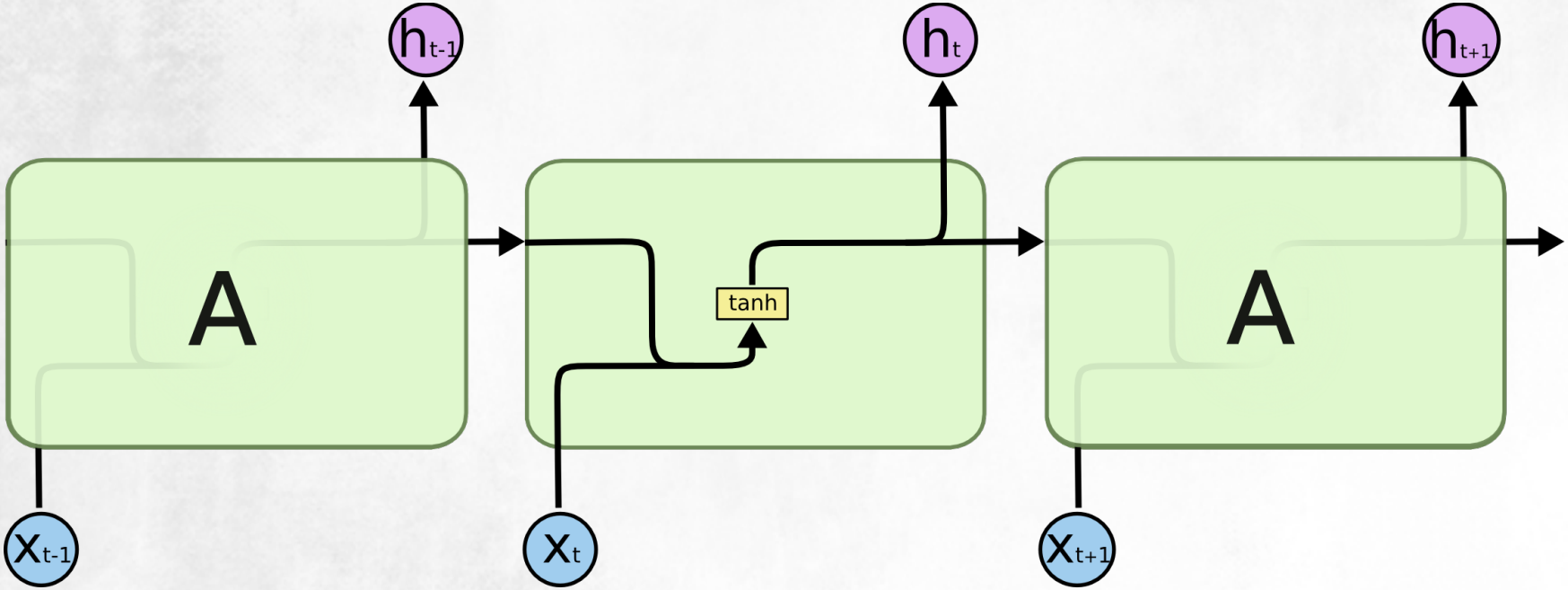



- Advantage
  - Naturally designed for lists and sequences.
  - Turing-complete
  - Variable
- Disadvantage
  - Hard to capture long-term dependency.

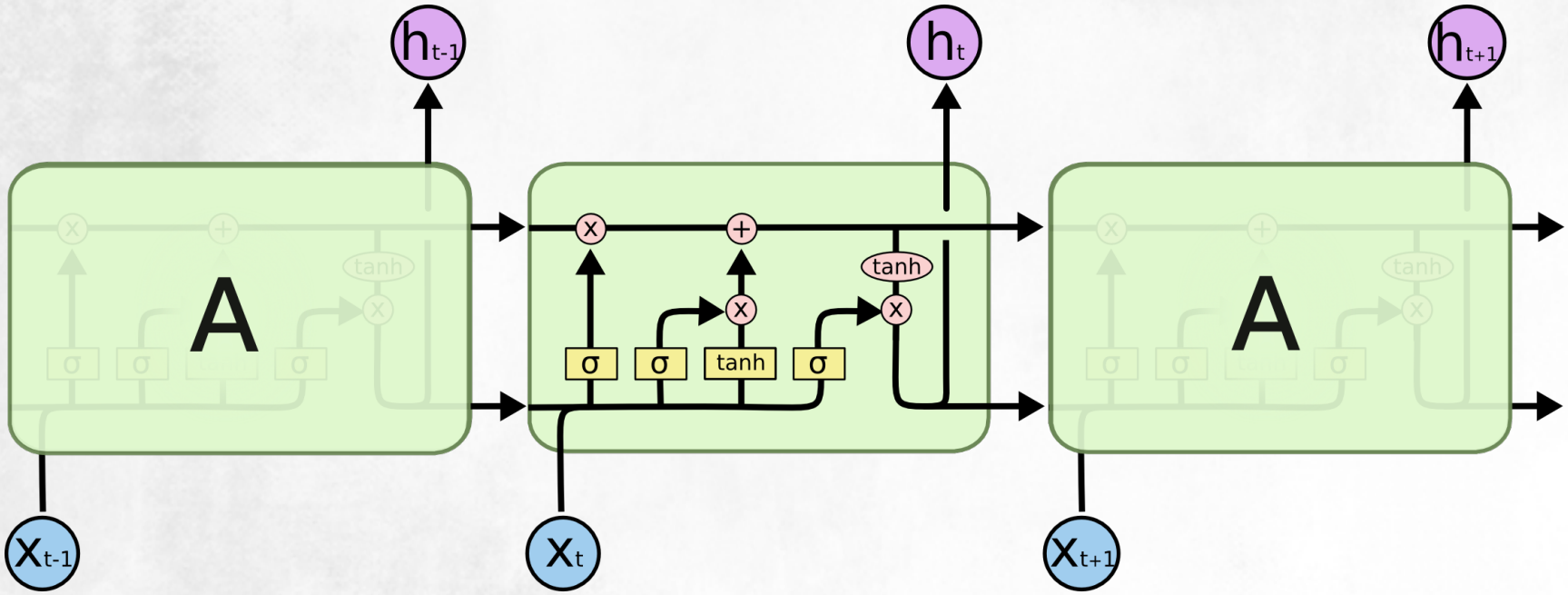
# Recap: LSTM

- Idea
  - A special kind of Recurrent-type NNs.
- Method
  - Networks with loops in them and long-term memory mechanism
- Feature
  - Cell state
- Variants
  - Gated Recurrent Unit, or GRU
  - Depth Gated RNNs
  - Clockwork RNNs

# Recap: LSTM



# Recap: LSTM




NETWORK
MODEL
IN NLP

# More than word: Sentence

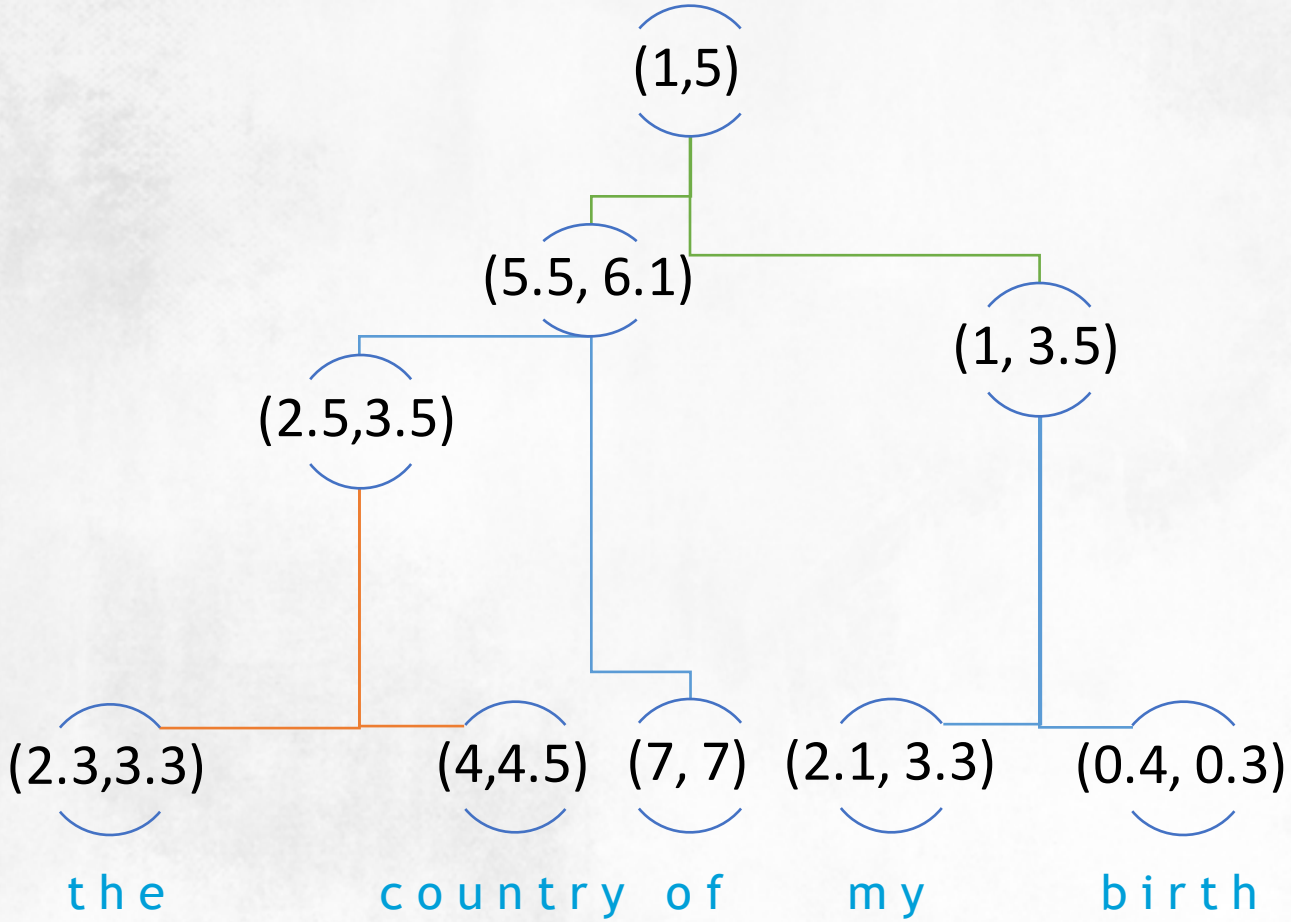
**How can we know when larger units are similar in meaning?**

People interpret the meaning of larger text units:

"entities, descriptive terms, facts, arguments, stories "

by semantic composition of smaller elements.
So does paragraph, document, etc.

# Principle of **Compositionality**



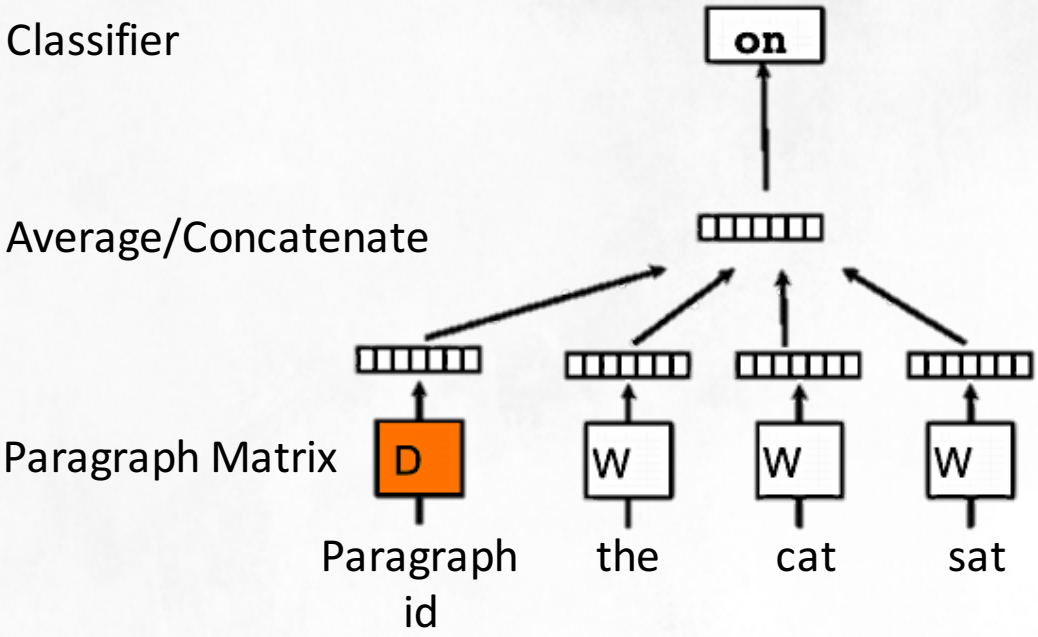
# Conjecture of **Structure**

**How to know meaning of high level structure?**

- Learn by pure embedding
  - Paragraph Vector(Google, Mikolov), also known as Doc2Vec, don't be confused☺
- Learn by input-dependent tree structure-based NN and memory unit/recursive cycle
  - TreeRNN(Stanford)
- Learn by deep continuous semantic space
  - DSSM(Microsoft)

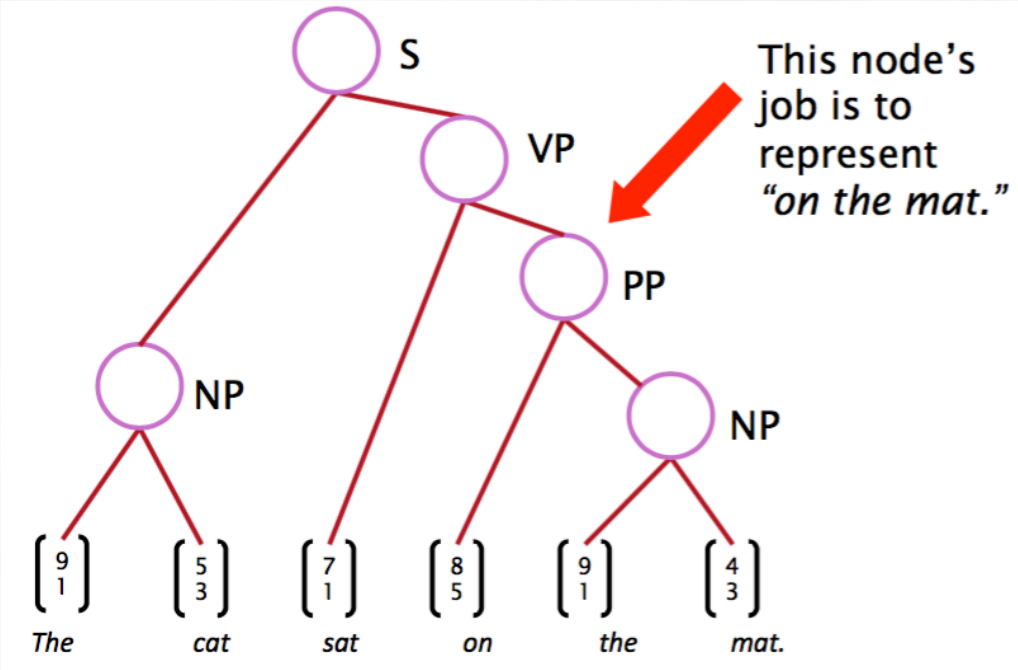
# Paragraph Vector

- Add up a paragraph vector to conclude paragraph independent information.
- Prediction task is typically done via a multiclass classifier, such as softmax or logistic regression.
- Learned from unlabeled data and thus can work well for tasks that do not have enough labeled data.
- Loss Function: $\frac{1}{T}\sum_{t=k}^{T-k}\log p(w_t|w_{t-k},...,w_{t+k})$



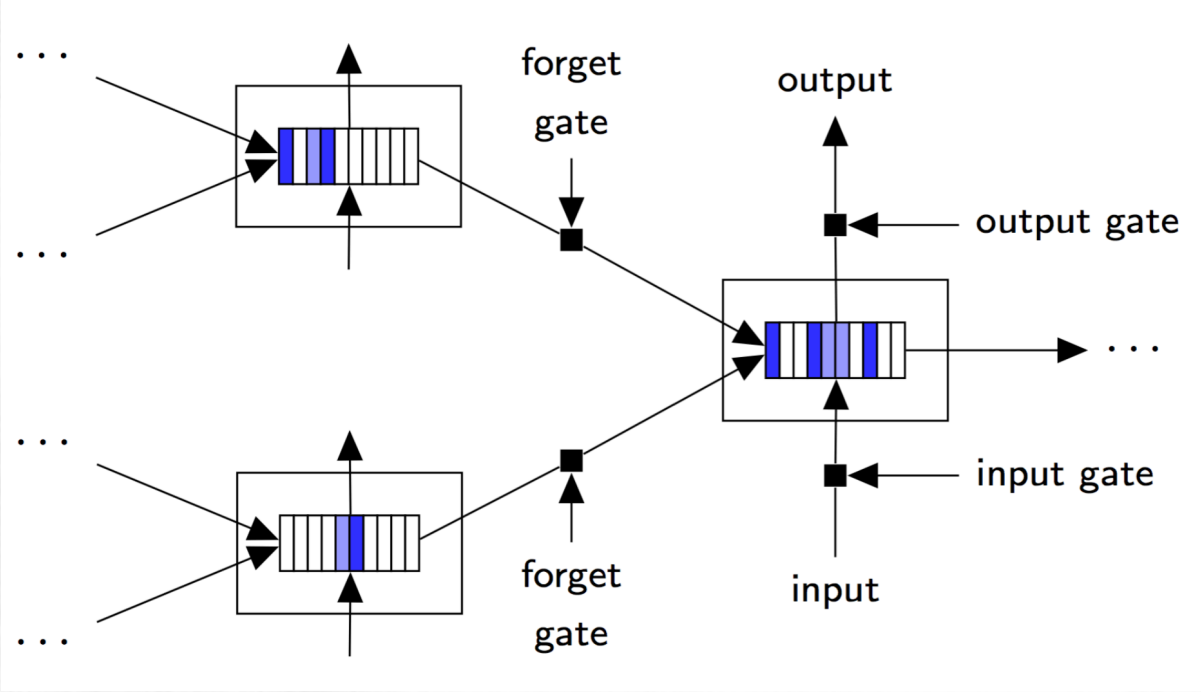
# Tree-based **RNN**

- Structured POS tree from input-dependent generation
- It can come from
  - A conventional statistical NLP parser, such as the Stanford Parser's PCFG
  - A neural network component, such as a neural network dependency parser
  - Learned and built as part of the training/operation of the TreeRNN system, by adding score matrix of the goodness of constituents
- Refer: Deep Learning Dependency Parser [Chen & Manning, EMNLP 2014] (支持中文)



# Tree-based **RNN**

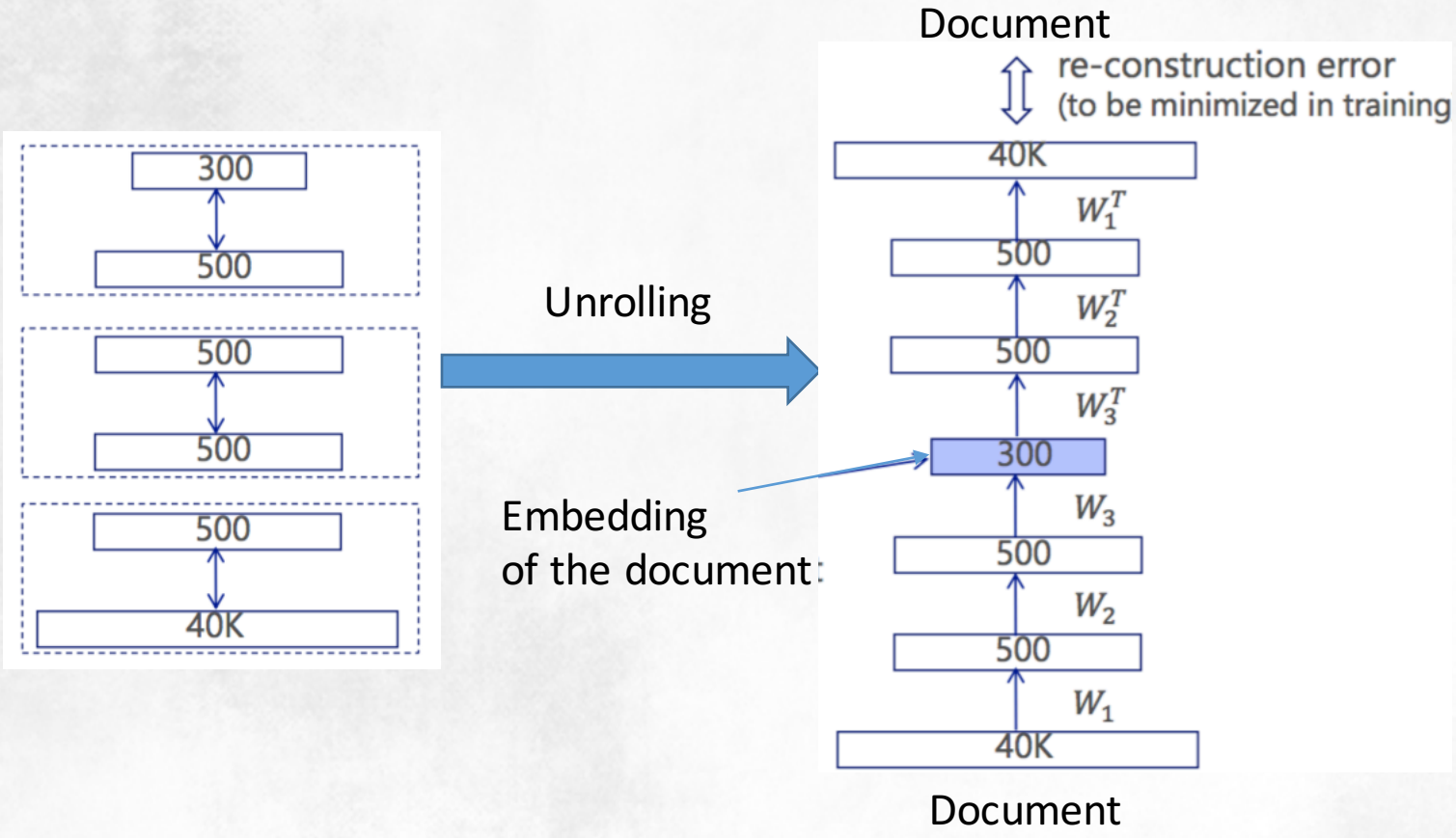
- Use Long Short-Term Memories
- Sentences have structure beyond word order - Use this syntactic structure
- Generalizes sequenGal LSTM to trees with any branching factor



# DSSM: Semantic Hashing

1) Single layer learning: Restricted Boltzmann Machine (RBM)
2) Multi-layer training: deep auto-encoder, learn internal representations

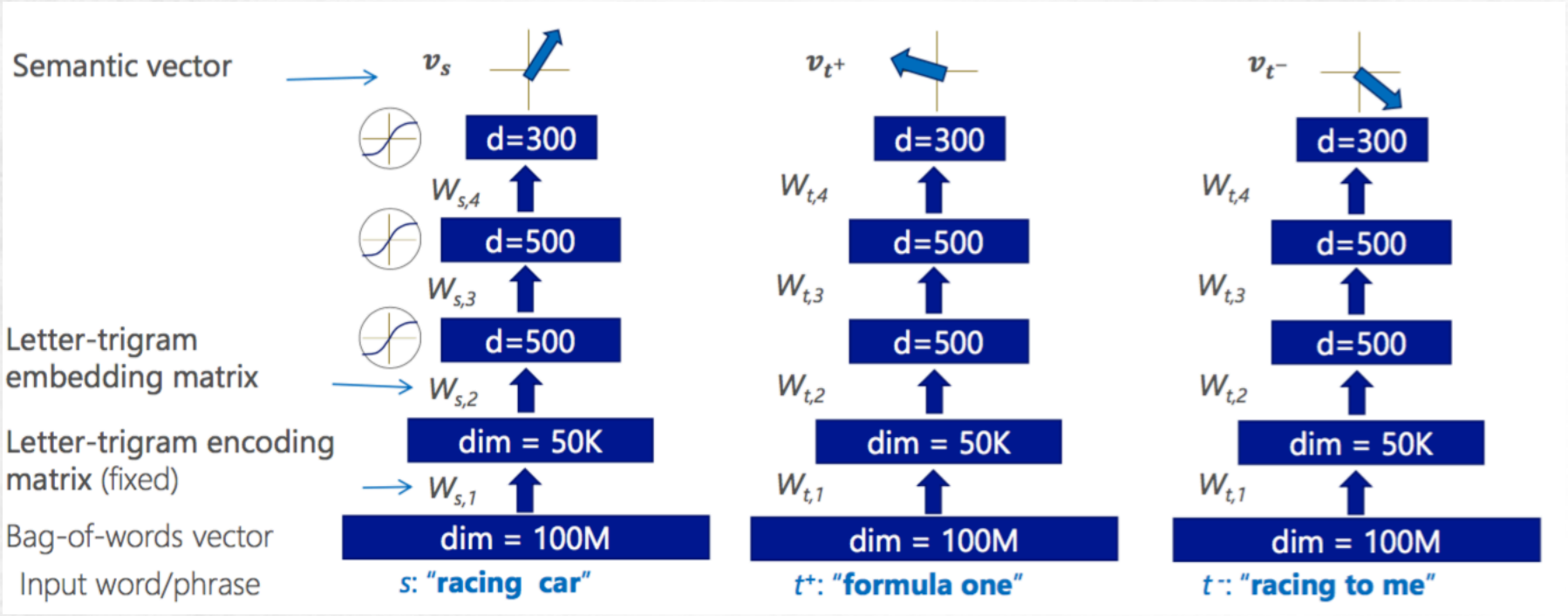
Model is trained to minimize the reconstruction error



# DSSM: rethinking

- The objective of training  the auto-encoder
  - The relation between minimizing reconstruction error and good embedding
- What is a good embedding
  - Good embedding helps end-to-end task:
    - Optimizing embedding directly instead of minimizing the doc reconstruction error
    - Learning the model with end-to-end user behavior log data(weak supervision) beside documents

# DSSM: Semantic Embedding




## Initialization:

Neural networks are initialized with random weights

## Training:

Compute Cosine similarity between semantic vectors

Compute gradients.

# DSSM: Summary

- Learn phrase/sentence level semantic vector representation
- Build upon sub-word units for scalability and generalizability
- Trained by an similarity-driven objective
  - Semi-Supervised
- Trained using various signals, with or without human labeling effort

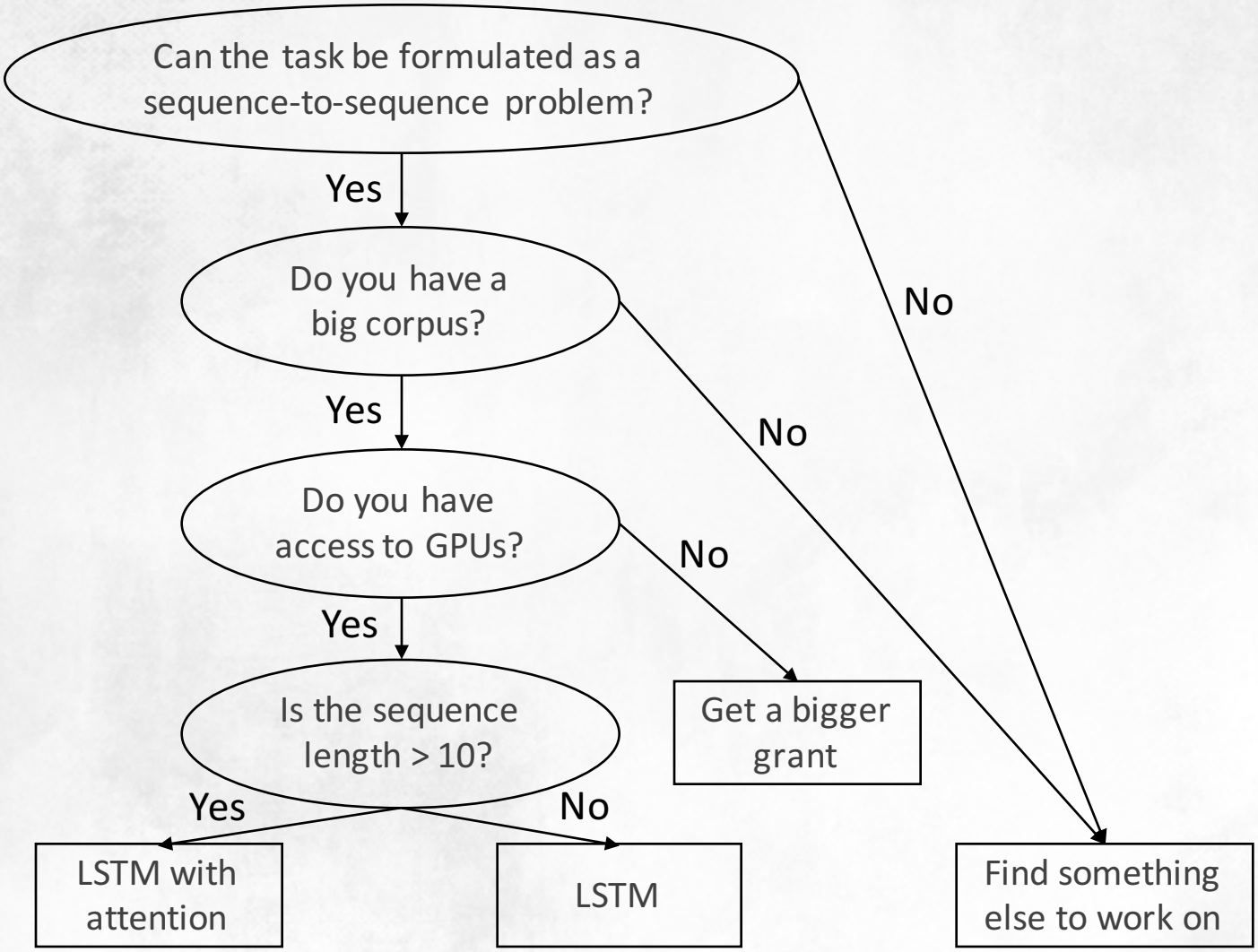
# DSSM: Application

- Compute semantic similarity between X and Y
  - Map X and Y to feature vectors in a latent semantic space via deep neural net
  - Computer the cosine similarity between the feature vectors
- DSSM for text processing tasks

| Tasks | X | Y |
|---|---|---|
| Web Search | Search query | Web document |
| Automatic highlighting | Doc in reading | Key phases to be highlighted |
| Contextual entity search | Key phrase and context | Entity and its corresponding page |
| Machine translation | Sentence in language A | Sentence in language B |
| Automatic Imaging captioning | Image | Caption |

PROSPECT

# FIRST OF ALL: A JOKE



# Summary

- LSTM-RNN
  - Capable to capture long-span dependency in natural language
  - LSTM-DSSM for IR (Palangi, et al.), LSTM for MT(Suskever, et al.)
- Recursive NN (ReNN)
  - Model the hierarchical structure of nature language
  - ReNN for parsing (Socher et al., 2011)
- Tensor product Representation (TPR)
  - Efficient representation of the structure of natural language
  - Smolensky & Legendre: The Harmonic Mind, From Neural Computation to Optimality-Theoretic Grammer, MIT Press, 2006

# Summary

- NLP has been in a boosting increasing with the help of DL.
- Word Embedding has been a essential step before further language processing.
- Supervised learning still takes an important role in industrial Application.
- Future Model would based on Unsupervised Learning and Reinforcement learning.
  - DQN
- Also does for knowledge transfer (Transfer learning).

# ThankYou

FOR LISTENING

# QUESTIONS AND ANSWERS

Go ahead. Ask away.