Part III. Implicit Representation for Short Text Understanding

Zhongyuan Wang (Microsoft Research)

Haixun Wang (Facebook Inc.)

Tutorial Website:
http://www.wangzhongyuan.com/tutorial/ACL2016/Understanding-Short-Texts/

“Implicit” model

- Goal:
  - A distributed representation of a short text that captures its semantics.
- Why?
  - To solve the sparsity problem
  - Representation readily used as features in downstream models

Short Text vs. Phrase Embedding

- There's a lot of work on embedding phrases.
- A short text (e.g., a web query) is often not well formed
    - e.g., no word order, no functional words
- A short text (e.g., a web query) is often more expressive
    - e.g., "distance earth moon"

Applications

GOOGLE TECH

Google is using an AI called 'RankBrain' to answer ambiguous questions

By Colin Lecher on October 26, 2015 09:54 am @colinlecher

http://www.theverge.com/2015/10/26/9614836/google-search-ai-rankbrain

THIRD MOST IMPORTANT SEARCH SIGNAL

RankBrain

- A huge vocabulary
  - Contains every possible token
- Query, doc title, doc URL representation
  - Average word embedding
- Architecture:
  - 3 – 4 hidden layers
- Data
  - Months of search log data

The Core Problem (for the rest of us)

- What is the objective function used in training the representation?
- Does the optimal solution force the representation to capture the full semantics?

Traditional Representation of Text

- Bag-of-Words (BOW) model: Text (such as a sentence or a document) is represented as a bag (multiset) of words, disregarding grammar and word order but keeping multiplicity.

  1. John likes to watch movie, Mary likes movie too.
  2. John also likes to watch football games.

  The sentences are represented by two 10-entry vectors;

  (1) [1,2,1,1,2,0,0,0,1,1]

  (2) [1,1,1,1,0,1,1,1,0,0]

- Disadvantages: No word order. Matrix is sparse.

Assumption: Distributional Hypothesis

- **Distributional Hypothesis**: Words that are used and occur in the same contexts tend to purport similar meaning (Wikipedia).

- E.g. *Paris is the capital of France*.

- In this assumption, “Paris” will be close in semantic space with “London” , which would also be surrounded by “capital of” and country’s name.

- Based on this assumption, researchers proposed many models to learn the text representations from corpus.

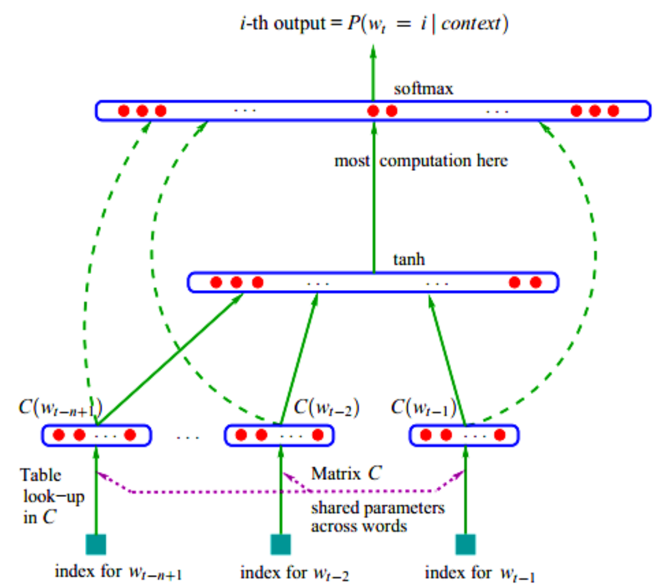
Neural Network Language Model (Bengio et al. 2003)

Statistical model

$$\hat{P}(w_1^T) = \prod_{t=1}^{T} \hat{P}(w_t | w_1^{t-1})$$

Assuming a word is determined by its **previous words**.

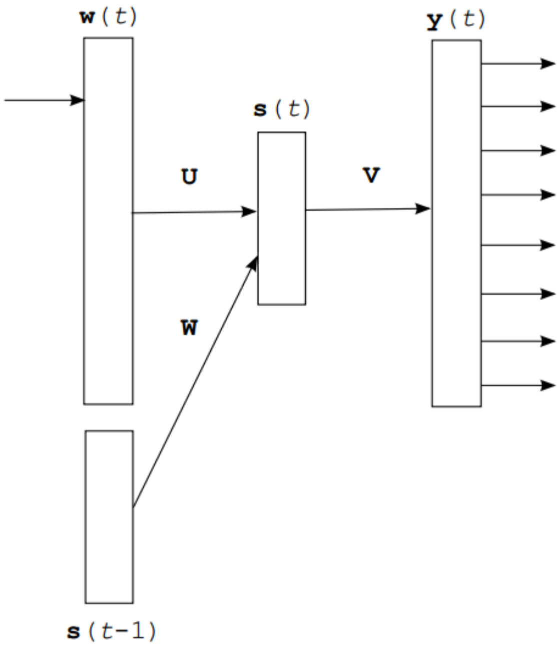
Two words with same previous words will share similar semantics.



Neural architecture: $f(i, w_{t-1}, \cdots, w_{t-n+1}) = g(i, C(w_{t-1}), \cdots, C(w_{t-n+1}))$ where g is the neural network and $C(i)$ is the i-th word feature vector.

Yoshua Bengio, Réjean Ducharme ,Pascal Vincent, Christian Jauvin “A Neural Probabilistic Language Model” Journal of Machine Learning Research 3 (2003) 1137–1155

Recurrent Neural Net Language Model (Mikolov, 2012)



Output Values:

$$s(t) = f(Uw(t) + Ws(t-1))$$

$$y(t) = g(Vs(t))$$

$w(t)$: *input word at time t*

y(t): *output probability distribution over words*

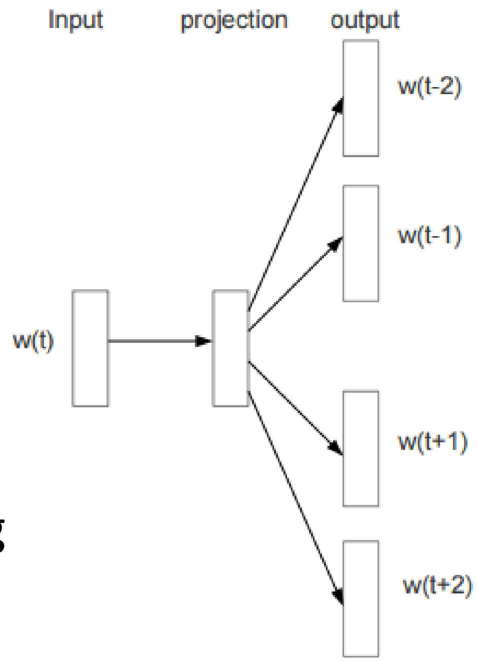
s(t): *hidden layer*

U,V,W: *transformation matrix*

- Generate much more meaningful text than n-gram models

- The sparse history h is projected into some continuous low-dimensional space, where similar histories get clustered

Word2Vector Model (Mikolov et al. 2013)



- The word2vec projects words in a **shallow** layer structure.

maximize $$\sum_{(w,c)\in D} \sum_{w_j \in c} \log P(w|w_j)$$

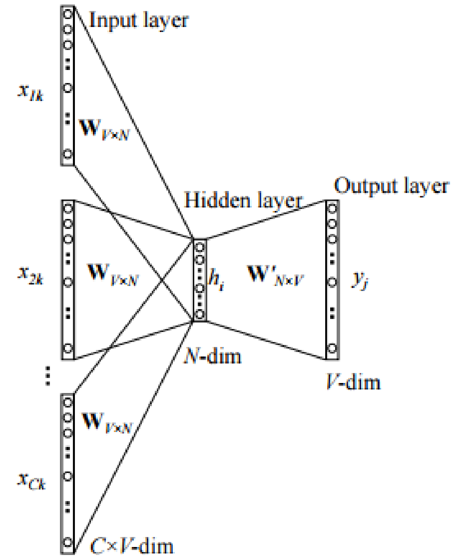
- **Directly** learn the representation of words using context words
- Optimizing the objective function in whole corpus.

Efficient estimation of word representations in vector space. Mikolov et al 2013

Word2Vector Model (Mikolov et al. 2013)

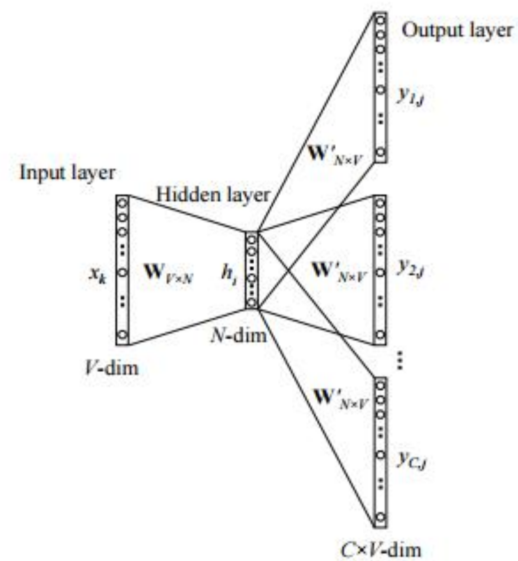
CBOW

- Given the **word**, predicting the **context**
- Faster to train than the skip-gram,
better accuracy for the **frequent words**



Skip-gram

- Given the **context**, predicting the **word**
- Works well with small training data,
represents well even **rare words** or **phrases**



GloVe: Global Vectors for Word Representation (Pennington et al. 2014)

- Constructing the word-word co-occurrence matrix of whole corpus.
- Inspired by LSA, using matrix factorization to produce word representation.

Probability and Ratio	$k = solid$	$k = gas$	$k = water$	$k = fashion$
$P(k\|ice)$	1.9×10^{-4}	6.6×10^{-5}	3.0×10^{-3}	1.7×10^{-5}
$P(k\|steam)$	2.2×10^{-5}	7.8×10^{-4}	2.2×10^{-3}	1.8×10^{-5}
$P(k\|ice)/P(k\|steam)$	8.9	8.5×10^{-2}	1.36	0.96

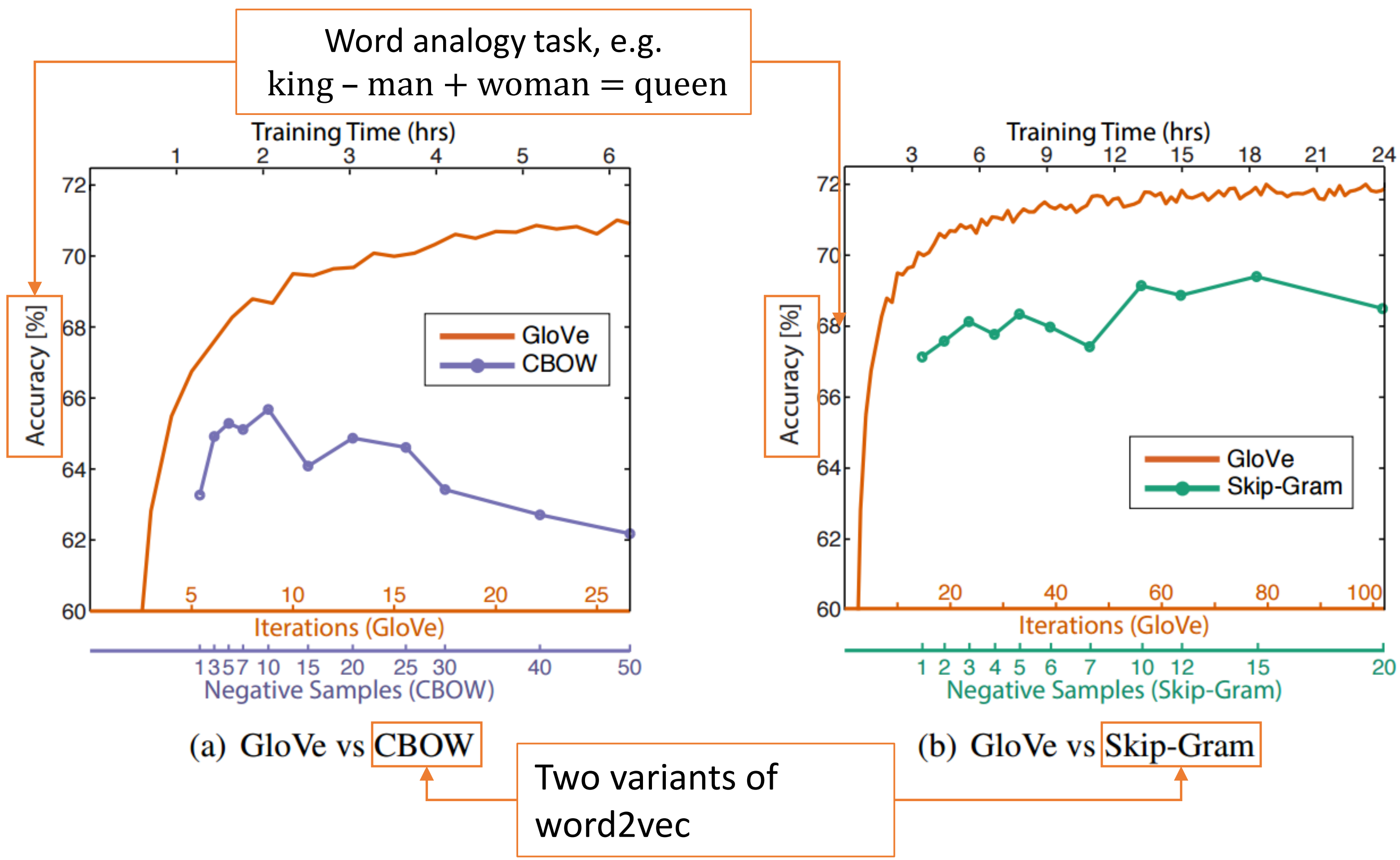
Loss function: $\hat{J} = \sum_{i,j} f(X_{ij}) \left(w_i^T \widetilde{w}_j - \log X_{ij}\right)^2$

X-ij is the count of if j-th word occurs, the occurrence of i-th word. ***w*** are word vectors. Minimize loss function.

Global Vectors for Word Representation, Pennington et al, 2014

GloVe: Global Vectors for Word Representation (Pennington et al. 2014)

- GloVe vs Word2Vec



Beyond words

Word embedding is a great success.

Phrase and sentence embedding is much harder:

- Sparsity: from atomic symbols to compositional structures
- Ground truth: from syntactic context to semantic similarity

Composition methods

- Algebraic composition

- Composition tied with syntax (dependency tree of phrase / sentences)

Averaging

- Expand vocabulary to include ngrams
- Otherwise go with bag of unigrams.

"A cat is being chased by a dog in yard"

$$\frac{v_1 + v_2 + \cdots + v_n}{n} = v_{sentence}\ ?$$

- But a "jade elephant" is not an "elephant"

Linear transformation

- $p = f(u, v)$, where

  u, v are embedding of uni-grams u, v

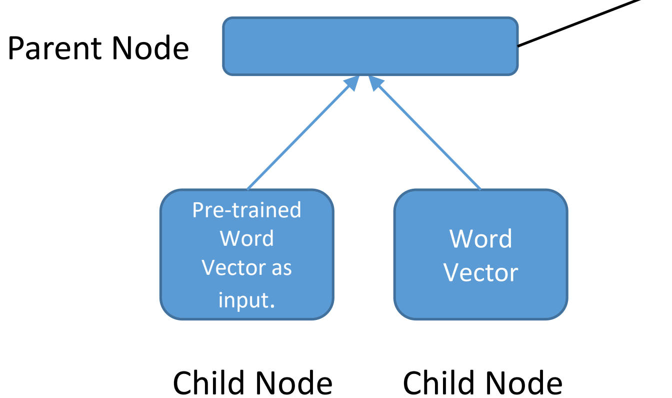
  f is a composition function

- Common composition model: **linear transformation**

- training data: unigram and bigram embeddings

Recursive Auto-encoder with Dynamic Pooling

- **Recursive Auto-encoder**
- **From bottom to top**, leaves to root.
- After parsing, **important** components in sentence will trend to get on **higher** level.



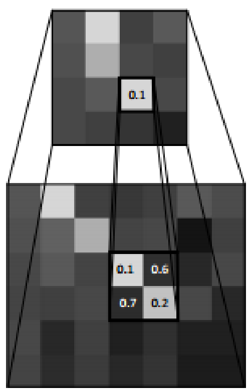
Non-linear activation function

$$p = f(W_e[c_1; c_2] + b)$$

$[c_1: c_2]$ is the concatenation of two word vectors

Recursive Auto-encoder with Dynamic Pooling

- **Dynamic Pooling**

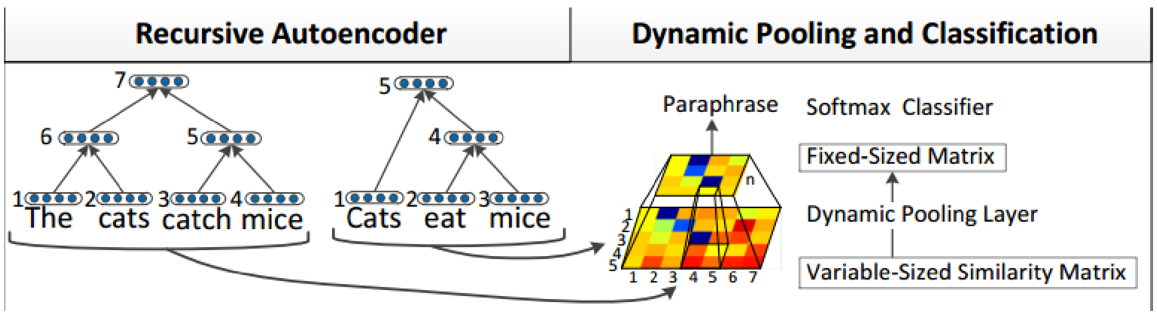


Example of the dynamic min-pooling layer finding the **smallest** number in a pooling window region of the original similarity matrix S.

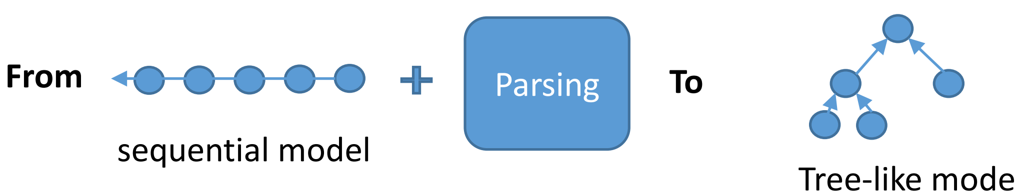
- The sentences are not fixed-size. Using pooling to map them into fix-sized vector.
- Using fixed-size matrix as input of neural network or other classifiers.

Recursive Auto-encoder with Dynamic Pooling [Socher et al. 2011]

- Using **dependency parser** to transform sequence to tree structure, which retains **syntactical info**
- Using **dynamic pooling** to map varied-size sentence to a **fixed-size form**



Most time, the para2vec model or traditional RNN/LSTM doesn’t consider the syntactical information of sentences.

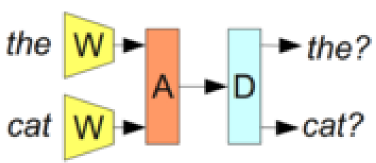


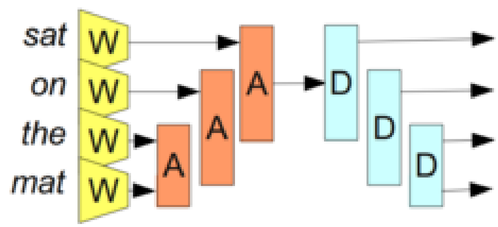
Richard Socher, Eric H. Huang, Jeffrey Pennington, Andrew Y. Ng, Christopher D. Manning: “Dynamic Pooling and Unfolding Recursive Autoencoders for Paraphrase Detection.” NIPS 2011: 801-809

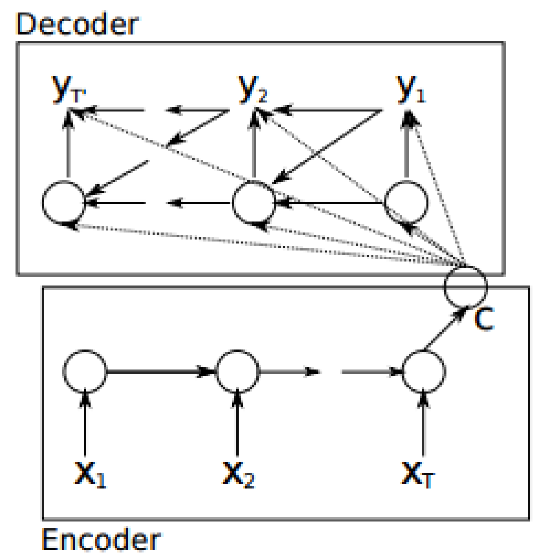
RNN encoder-decoder

(Cho et al. 2014)

- Create a **reversible** sentence representation.
- The representation can be reconstructed to an actual sentence form which is reasonable and novel.



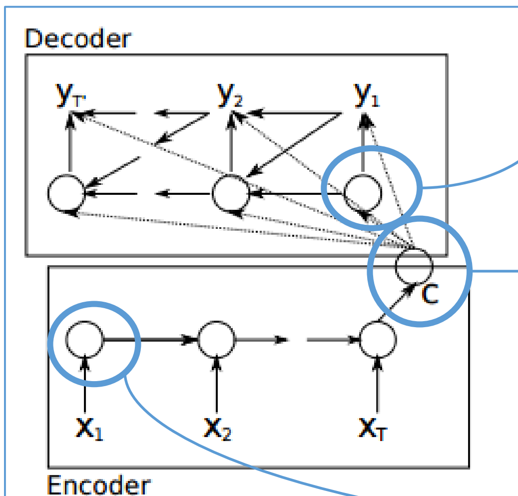




Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, Yoshua Bengio: “Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation.” EMNLP 2014: 1724-1734

RNN encoder-decoder

(Cho et al. 2014)



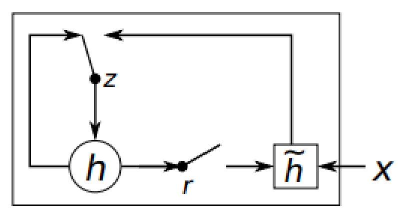
- The conditional distribution of next symbol.

$$P(y_t|y_{t-1}, y_{t-2}, \ldots, y_1, c) = g(h_{<t>}, y_{t-1}, c)$$

- Add a summary(constant) symbol, it will hold the semantics of sentence.

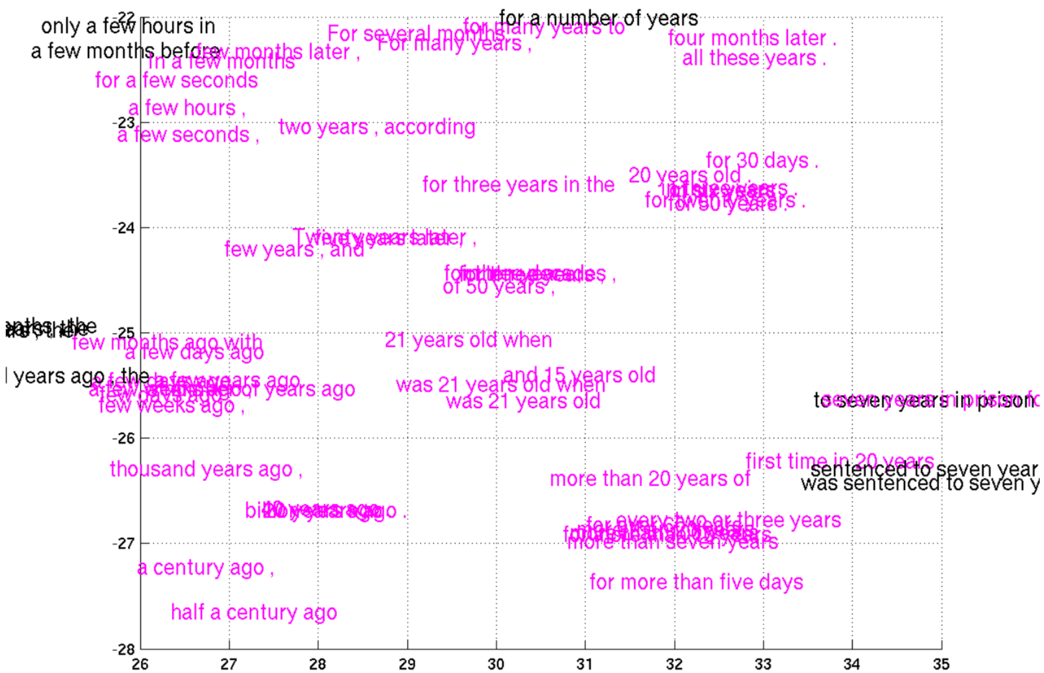
$$h_{<t>} = f(h_{<t-1>}, y_{t-1}, c)$$

- For long sentences, adding hidden unit to remember/forget memory.



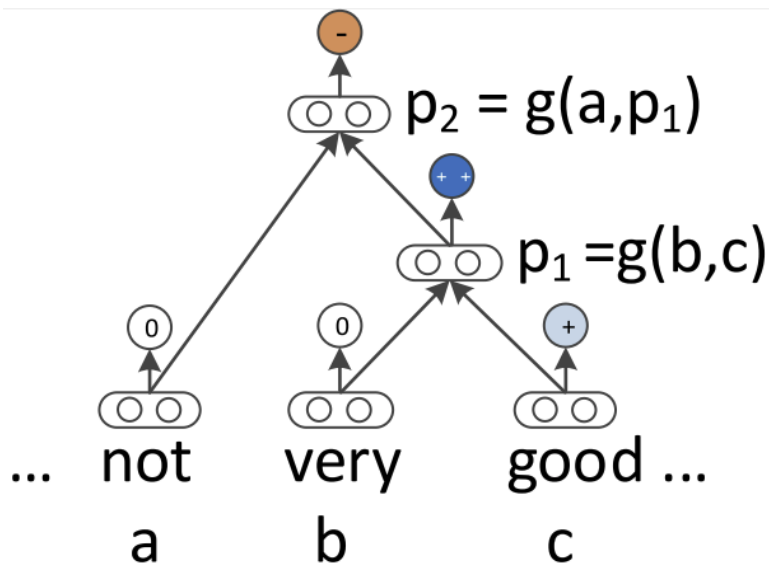
RNN encoder-decoder

(Cho et al. 2014)



Small section of the t-SNE of the phrase representation

RNN for composition [Socher et al 2011]



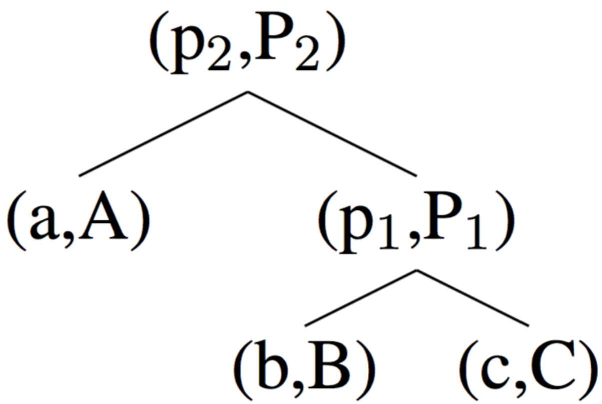
$$p_1 = f\left(W\begin{bmatrix} b \\ c \end{bmatrix}\right), p_2 = f\left(W\begin{bmatrix} a \\ p_1 \end{bmatrix}\right)$$

f = tanh is a standard element-wise nonlinearity

W is shared

MV-RNN [Socher et al. 2012]

- Each composition function depends on the actual words being combined.

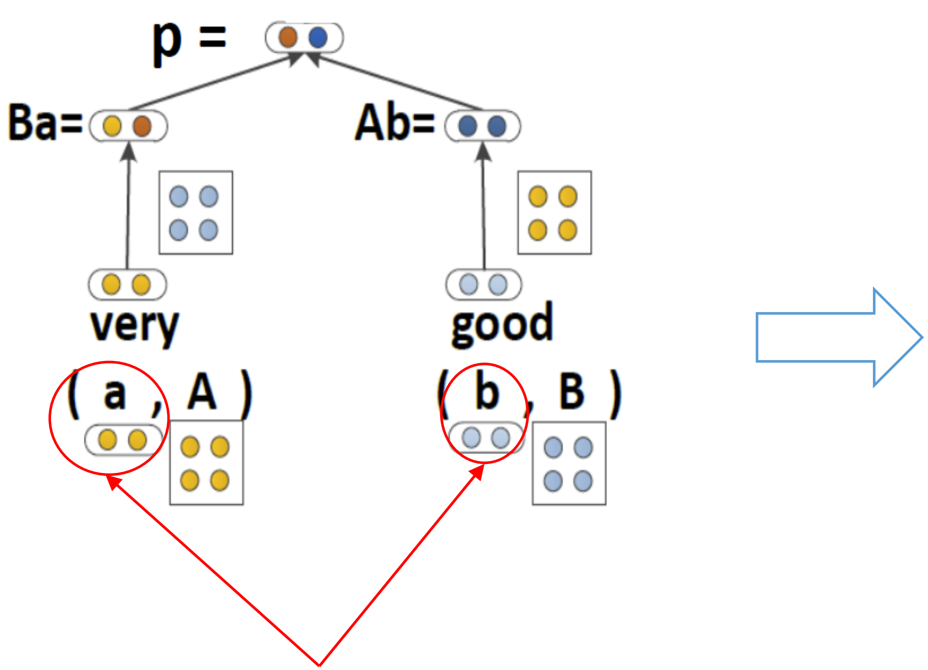


$$p_1 = f\left(W \begin{bmatrix} Cb \\ Bc \end{bmatrix}\right), P_1 = f\left(W_M \begin{bmatrix} B \\ C \end{bmatrix}\right)$$

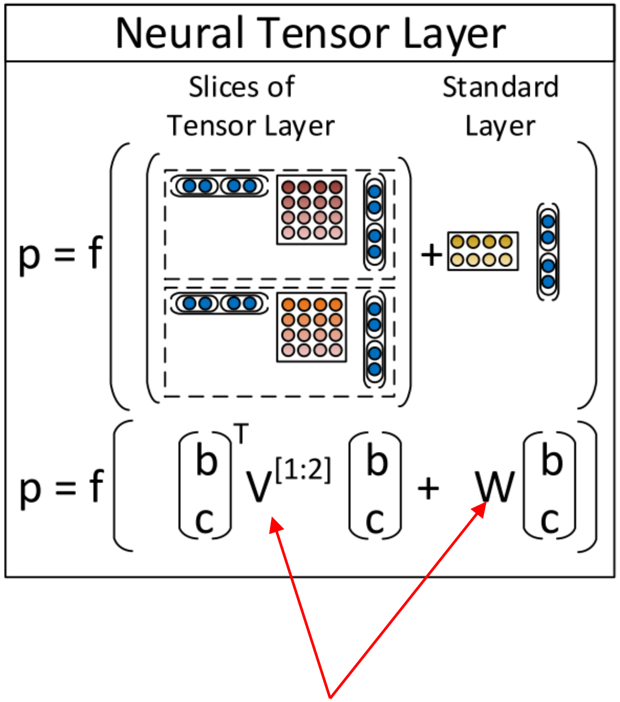
- Represent every word and phrase as both a vector and a matrix.

Recursive Neural Tensor Network [Socher et al. 2013]

- Number of parameters is very large for MV-RNN



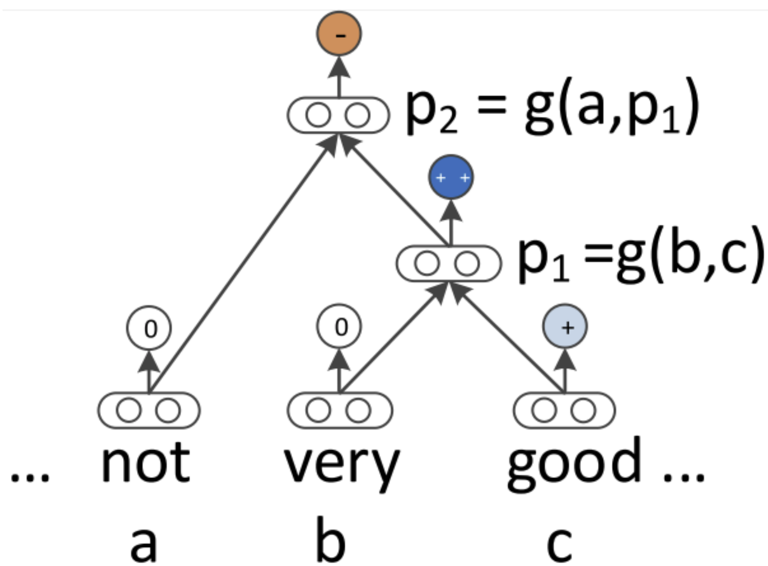
MV-RNN: need to train a new parameter for each leaf node



Use tensor: unified parameter for all nodes

Recursive Neural Tensor Network [Socher et al. 2013]

- Interpret each slice of the tensor as capturing a specific type of composition



$$p_1 = f\left(\begin{bmatrix} b \\ c \end{bmatrix}^T V^{[1:d]} \begin{bmatrix} b \\ c \end{bmatrix} + W \begin{bmatrix} b \\ c \end{bmatrix}\right)$$

$$p_2 = f\left(\begin{bmatrix} a \\ p_1 \end{bmatrix}^T V^{[1:d]} \begin{bmatrix} a \\ p_1 \end{bmatrix} + W \begin{bmatrix} a \\ p_1 \end{bmatrix}\right)$$

Assign label to each node via:

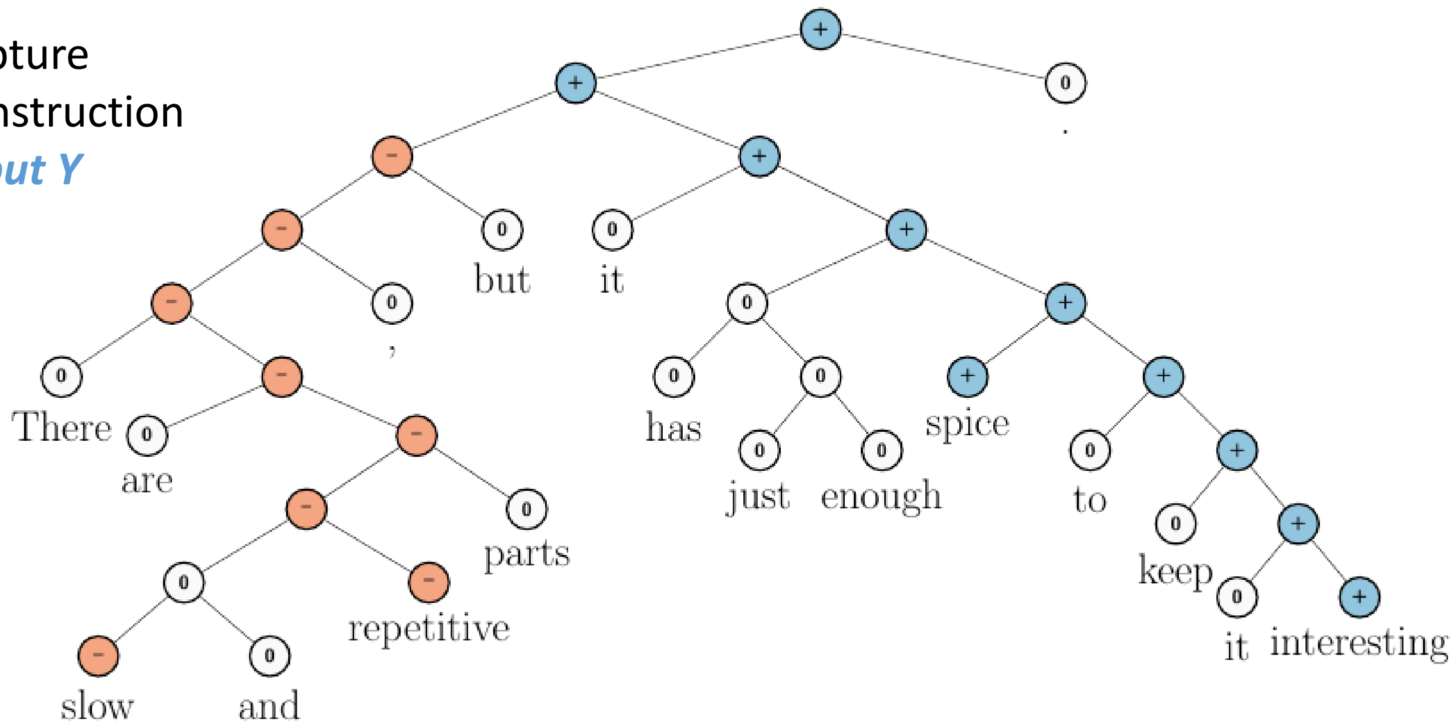
$$y^a = \mathrm{softmax}(W_s a)$$

Recursive Neural Tensor Network

- Target : sentiment analysis

Sentence: There are slow and repetitive parts,
but it has just enough spice to keep it interesting

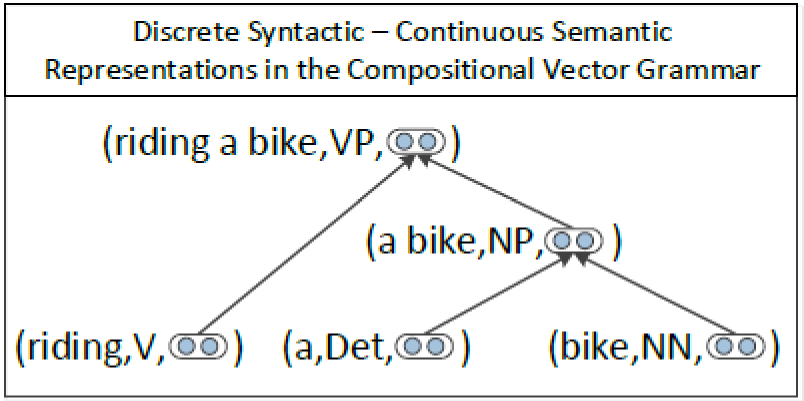
capture
construction
X but Y



Demo: http://nlp.stanford.edu:8080/sentiment/rntnDemo.html

CVG (Compositional Vector Grammars)
[Socher et al. 2013]

- Task: Represent phrase and categories
  - PCFG: capture discrete categorization of phrases
  - RNN: capture fine-grained syntactic and compositional-semantic information
- Parse and represent phrases as vector



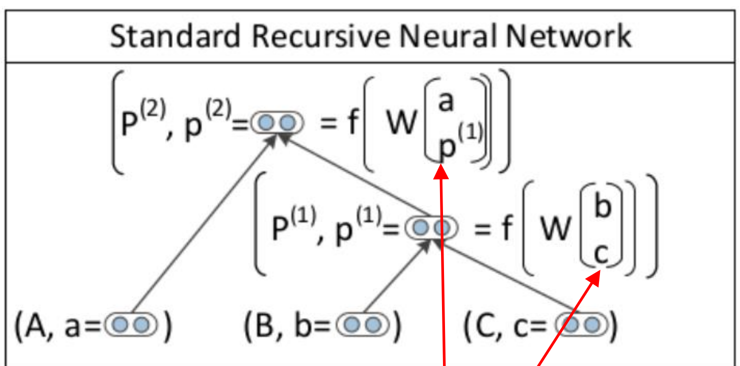
An example of CVG Tree

Parsing with Compositional Vector Grammars, Socher et al 2013

CVG

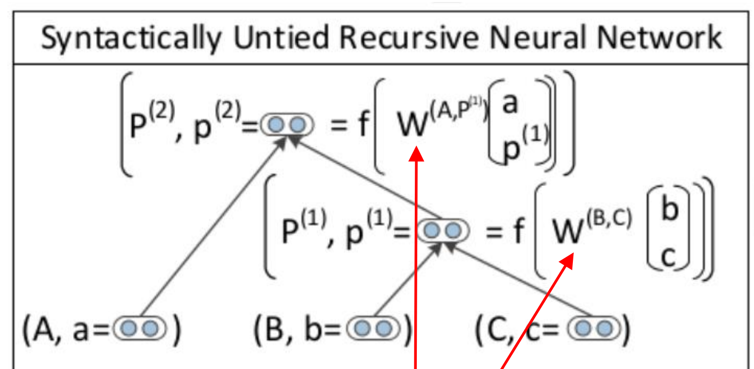
- Weights at each node are conditionally dependent on categories of the child constituents
- Combined with Syntactically Untied RNN

Normal RNN



Replicated weight matrix

SU-RNN

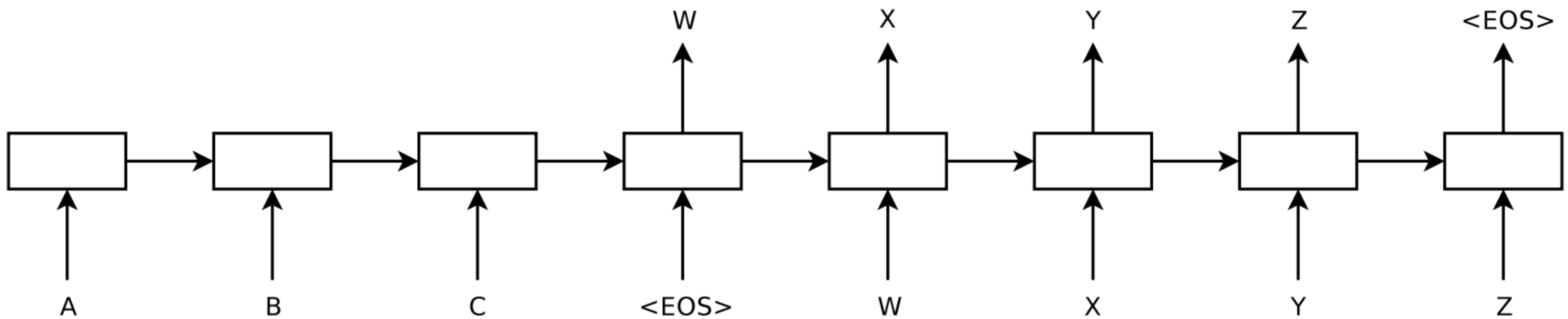


depends on syntactic categories of its children

Phrases & Sentences

- Composition based approaches
  - Algebraic composition not powerful enough
  - Syntactic composition requires parsing
- Non-composition based approaches
  - translation based approaches
  - extend word2vec to sentences, phrases
  - ground truth: search log, dictionary, image

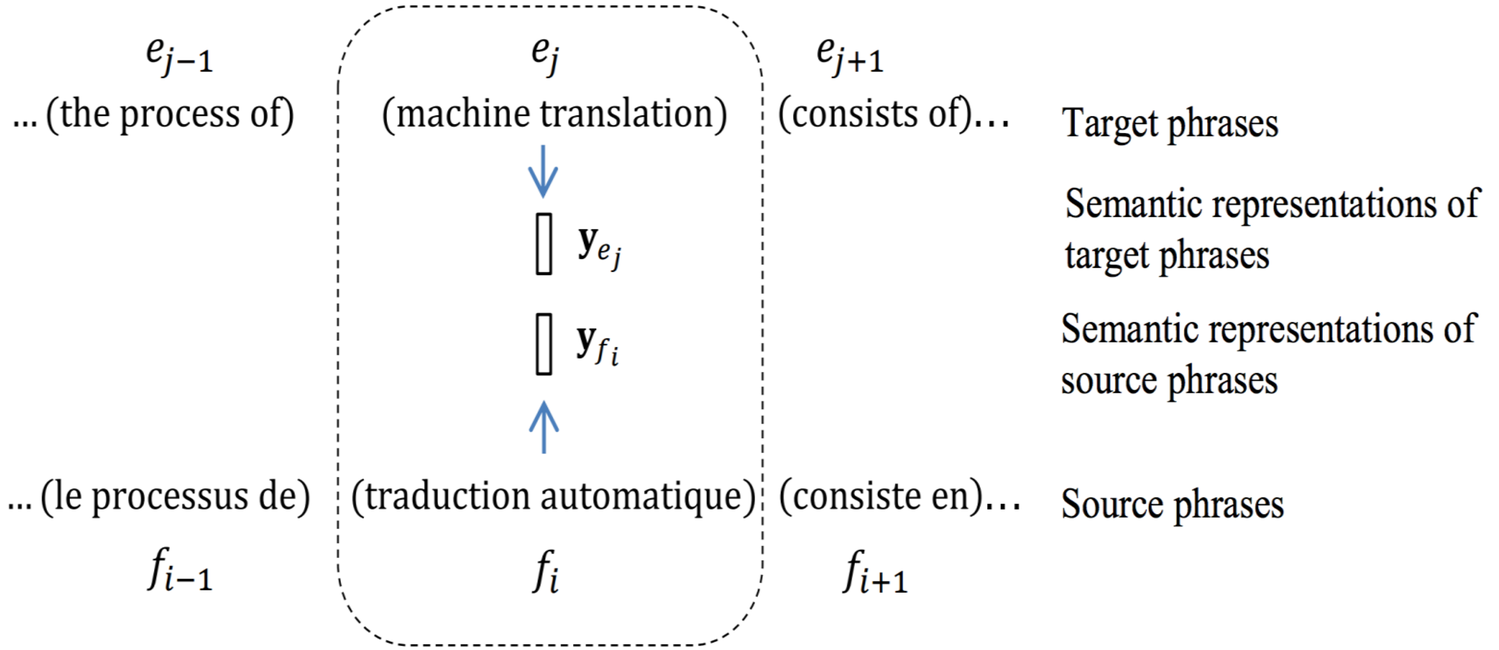
Sequence to sequence translation



- Last node “remembers” the semantics of the input sentence

- Not feasible for embedding web queries

Phrase Translation Model [Gao et al 2013]



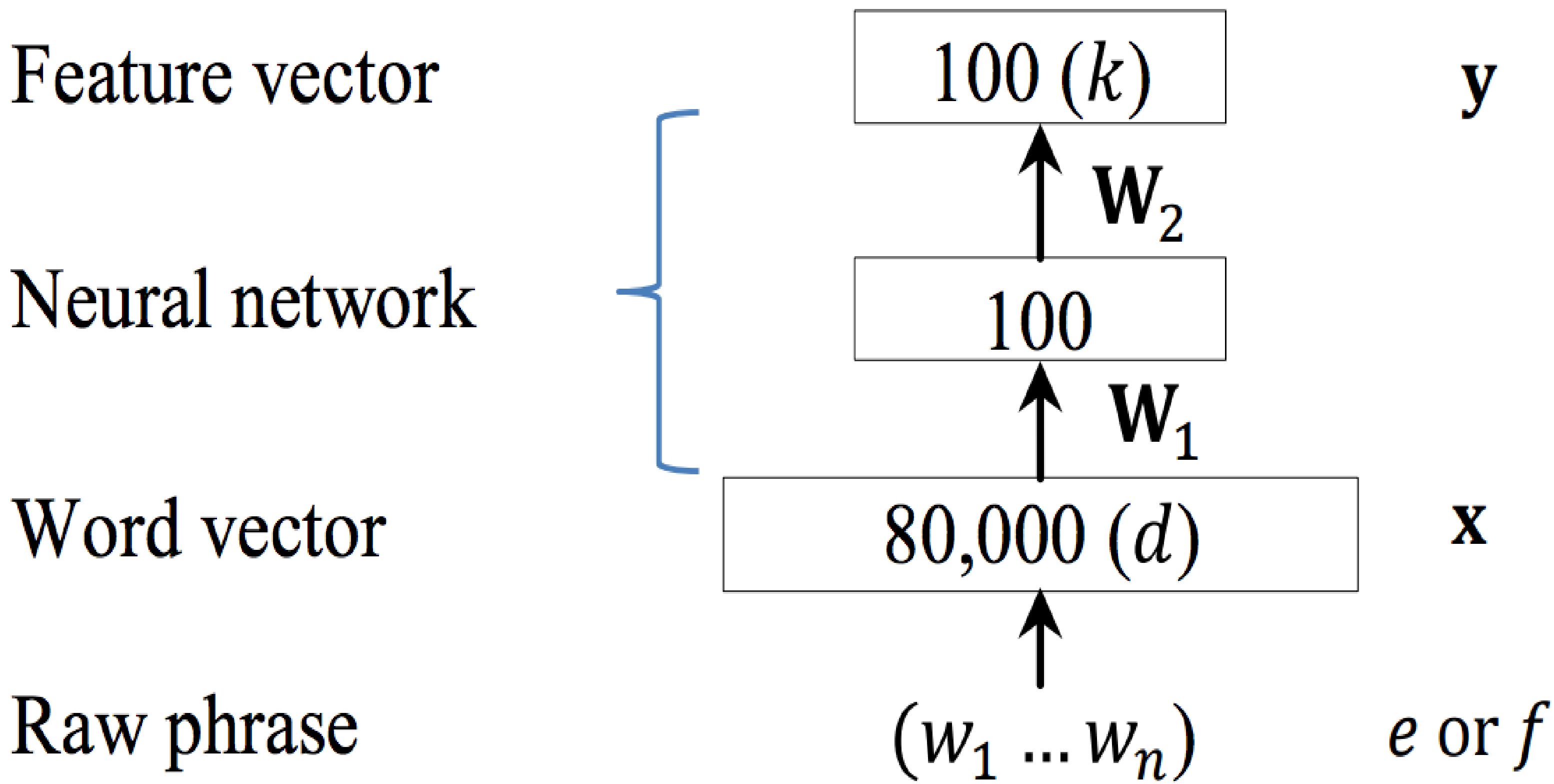
$$\text{score}(f_i, e_j) = \mathbf{y}_{f_i}^{\mathrm{T}} \mathbf{y}_{e_j}$$

Translation score as dot product of feature vectors in the semantic space

The quality of a phrase translation is judged implicitly through the translation quality (BLEU) of the sentences that contain the phrase pair.

Learning Semantic Representations for the Phrase Translation Model, Gao et al 2013

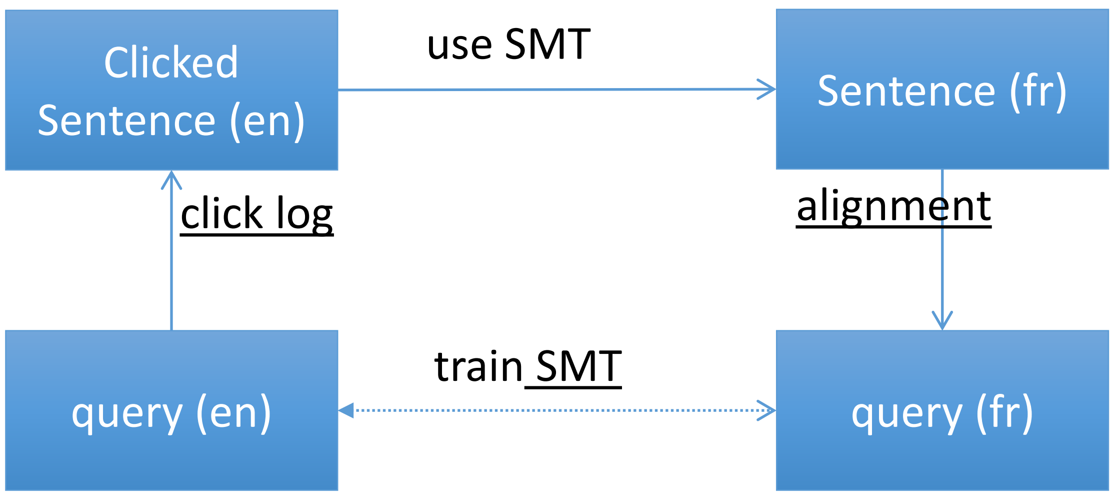
Phrase Translation Model



The core is the bag-of-words approach

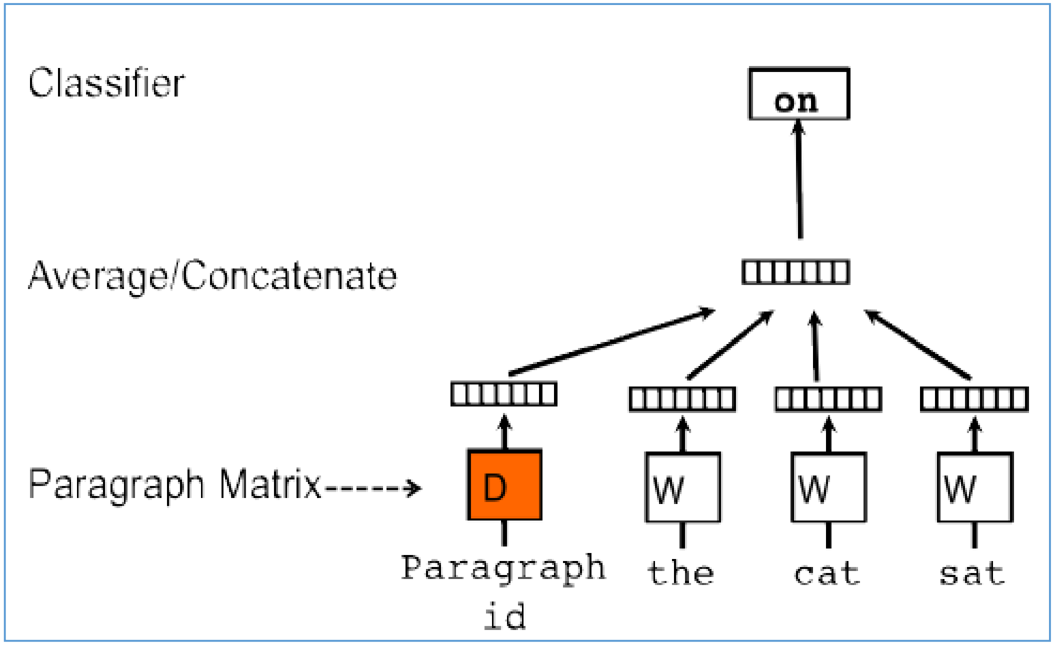
Web query translation model

- Training data



- Train an NN translation model on query(en) and query(fr) pair

Doc2Vec (Quoc Le et al 2014)



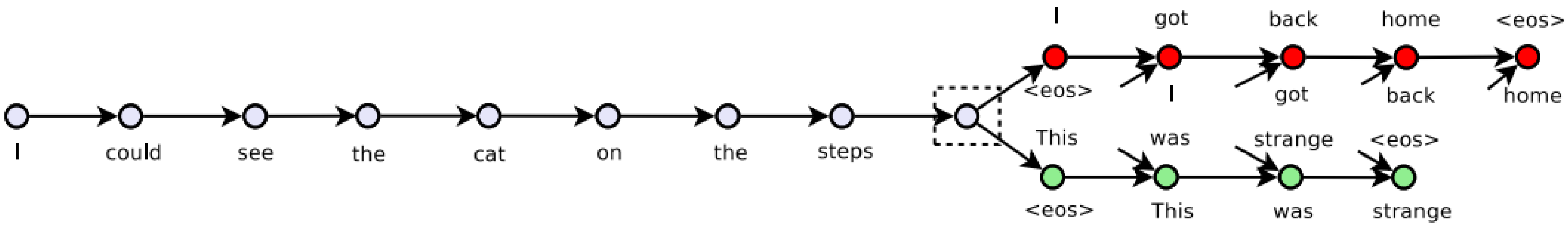
Distributed Representations of Sentences and Documents, Quoc Le et al. 2014

LDA vs. Doc2Vec

LDA	**Paragraph Vectors**
Artificial neural network	Artificial neural network
Predictive analytics	Types of artificial neural networks
Structured prediction	Unsupervised learning
Mathematical geophysics	Feature learning
Supervised learning	Predictive analytics
Constrained conditional model	Pattern recognition
Sensitivity analysis	Statistical classification
SXML	Structured prediction
Feature scaling	Training set
Boosting (machine learning)	Meta learning (computer science)
Prior probability	Kernel method
Curse of dimensionality	Supervised learning
Scientific evidence	Generalization error
Online machine learning	Overfitting
N-gram	Multi-task learning
Cluster analysis	Generative model
Dimensionality reduction	Computational learning theory
Functional decomposition	Inductive bias
Bayesian network	Semi-supervised learning

Similar topics to “Machine Learning” returned by LDA and Doc2Vec

Skip Thought Vectors (Kiros et al 2015)



Given a tuple (s_{i-1}, s_i, s_{i+1}) of contiguous sentences, with s_i the i-th sentence of a book, the sentence s_i is encoded and tries to reconstruct the previous sentence $s_{i-1}$and next sentence s_{i+1}.

In this example, the input is the sentence triplet *I got back home*. *I could see the cat on the steps. This was strange.* Unattached arrows are connected to the encoder output. Colors indicate which components share parameters. <eos> is the end of sentence token.

Skip Thought Vector

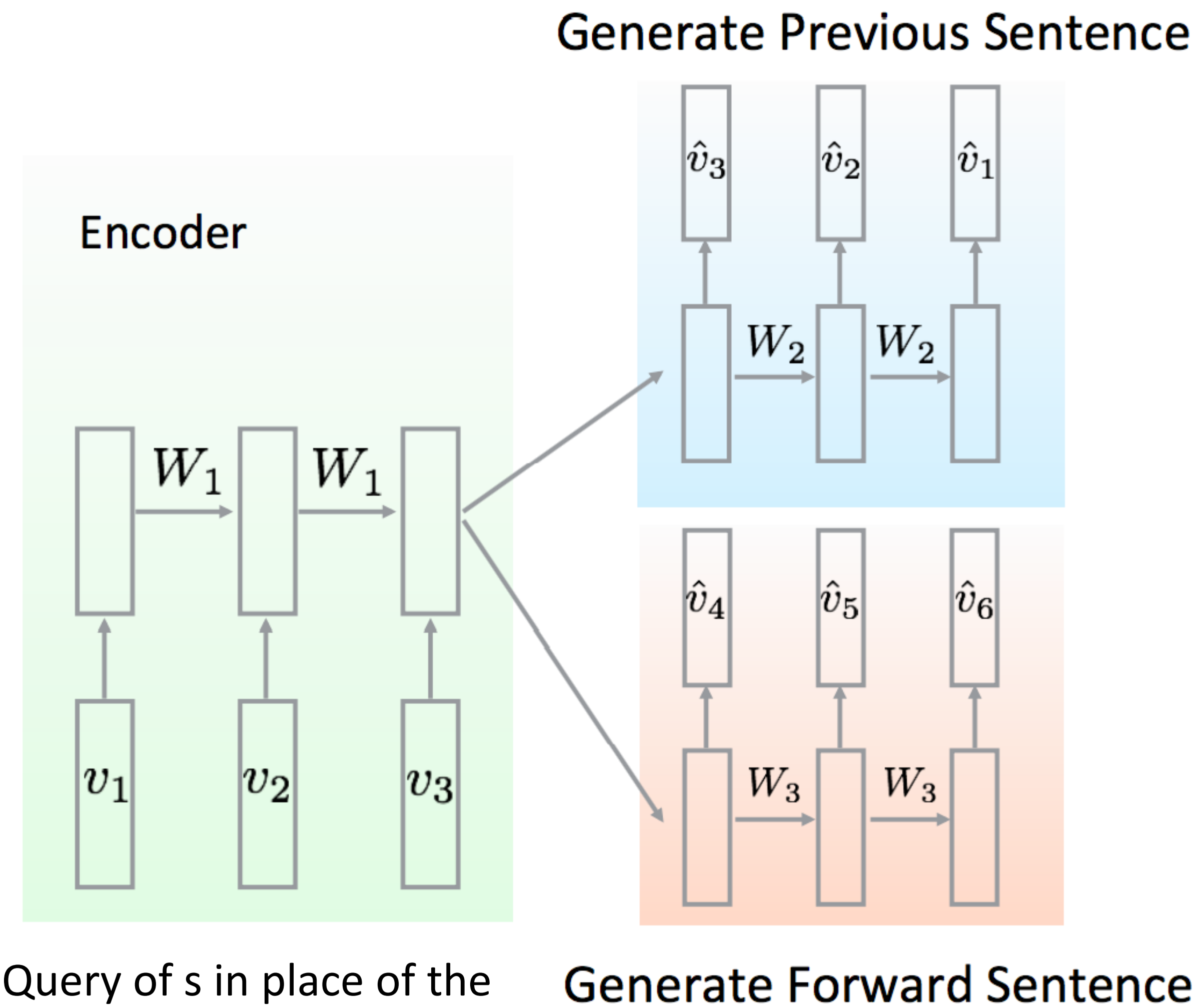
(Ryan Kiros et al. 2015)

- Semantic relatedness:

Sentence 1	**Sentence 2**	**GT**	**pred**
A little girl is looking at a woman in costume	A young girl is looking at a woman in costume	4.7	4.5
A little girl is looking at a woman in costume	The little girl is looking at a man in costume	3.8	4.0
A little girl is looking at a woman in costume	A little girl in costume looks like a woman	2.9	3.5
A sea turtle is hunting for fish	A sea turtle is hunting for food	4.5	4.5
A sea turtle is not hunting for fish	A sea turtle is hunting for fish	3.4	3.8
A man is driving a car	The car is being driven by a man	5	4.9
There is no man driving the car	A man is driving a car	3.6	3.5

GT is ground truth relatedness, Pred is prediction by trained model.

Skip thought vector for phrases

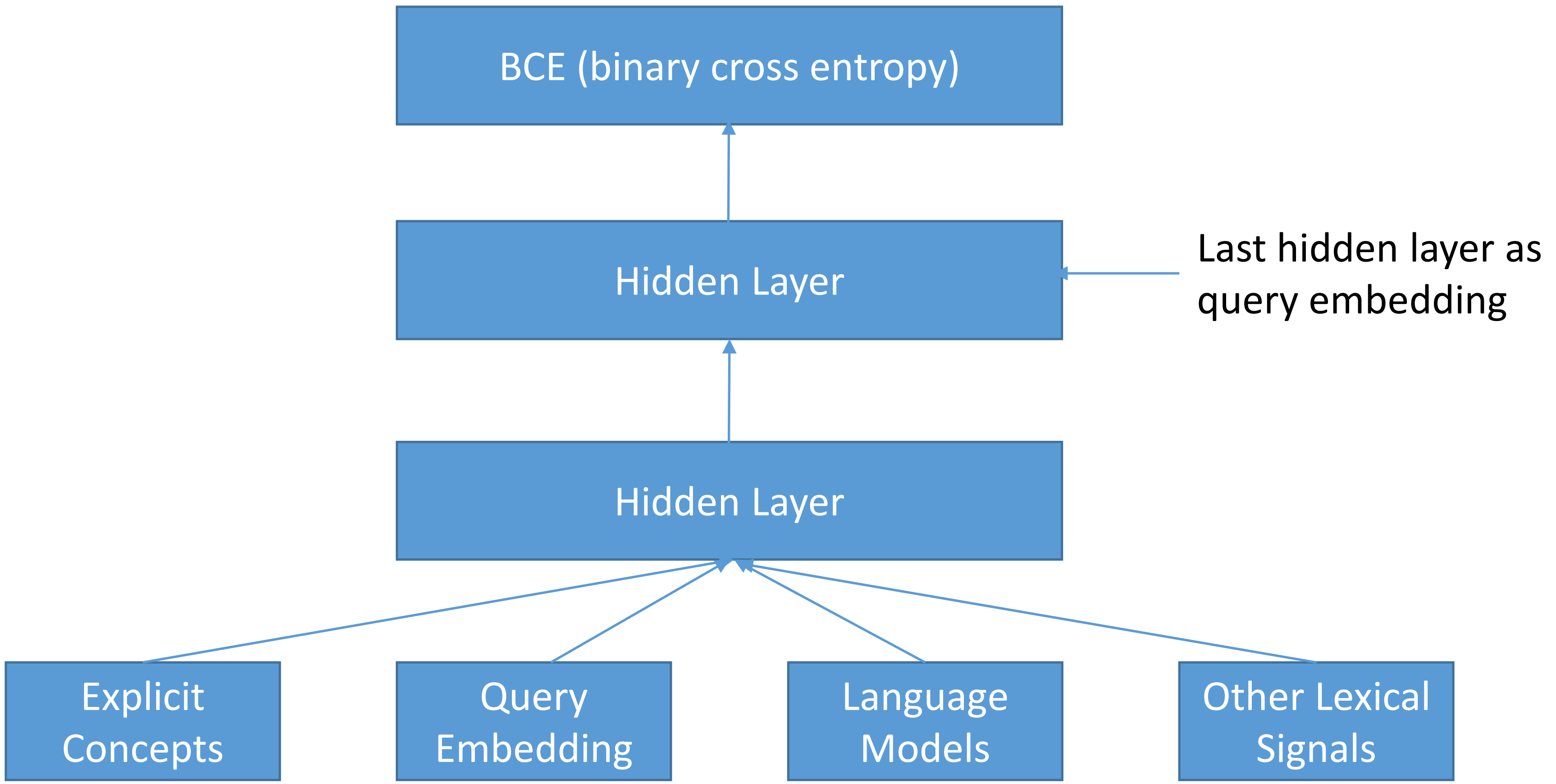


Query of s in place of the original, clicked sentence s

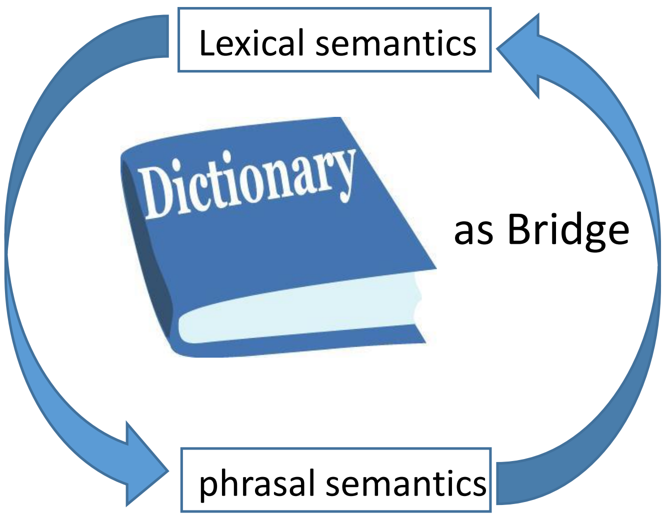
Translation vs. Syntactic context

- Different property of representation
- Different perplexity
- Different applications

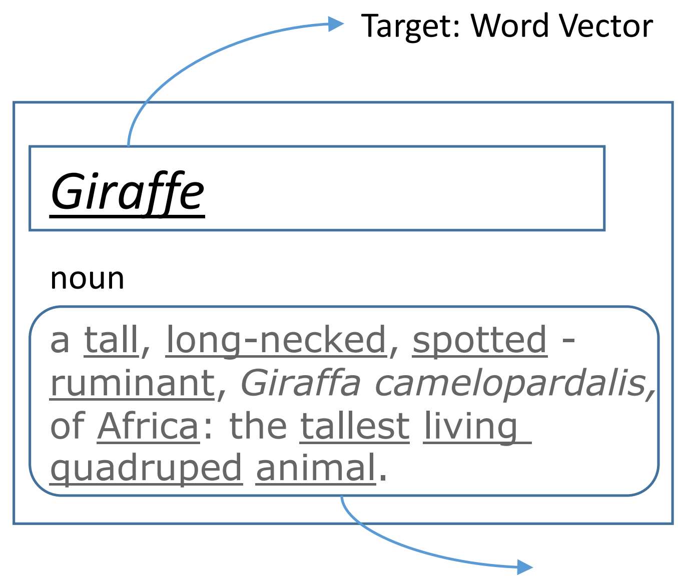
Phrase Embedding: Using a Multi-label Classifier



Phrase Embedding: Using a Dictionary



Goal: From word representation to phrase and sentence representation



The representation of definition should be closed with defining word vector.

Felix Hill, Kyunghyun Cho, Anna Korhonen, Yoshua Bengio:
Learning to Understand Phrases by Embedding the Dictionary. TACL 4: 17-30 (2016)

Phrase Embedding: Using a Dictionary

Model:

Pre-trained Input Representation	$\xrightarrow{\text{input}}$	Neural Language Model	$\xrightarrow{\text{optimize}}$	Objective Function

Pre-trained Input Representation

Definitions

"control consisting of a mechanical device for controlling fluid flow"

"when you like one thing more than another thing"

.
.
.

Word2Vec as each word's representation

Neural Language Model

$$A_t = \phi(Uv_{t-1} + Wv_t + b)$$

Recurrent Neural Networks

or

Bag-of-Words

$$A_t = A_{t-1} + Wv_t$$

Objective Function

Words

"Valve"

"Prefer"

$$\max(0, m - \cos(M(s_c), v_c) - \cos(M(s_c), v_r))$$

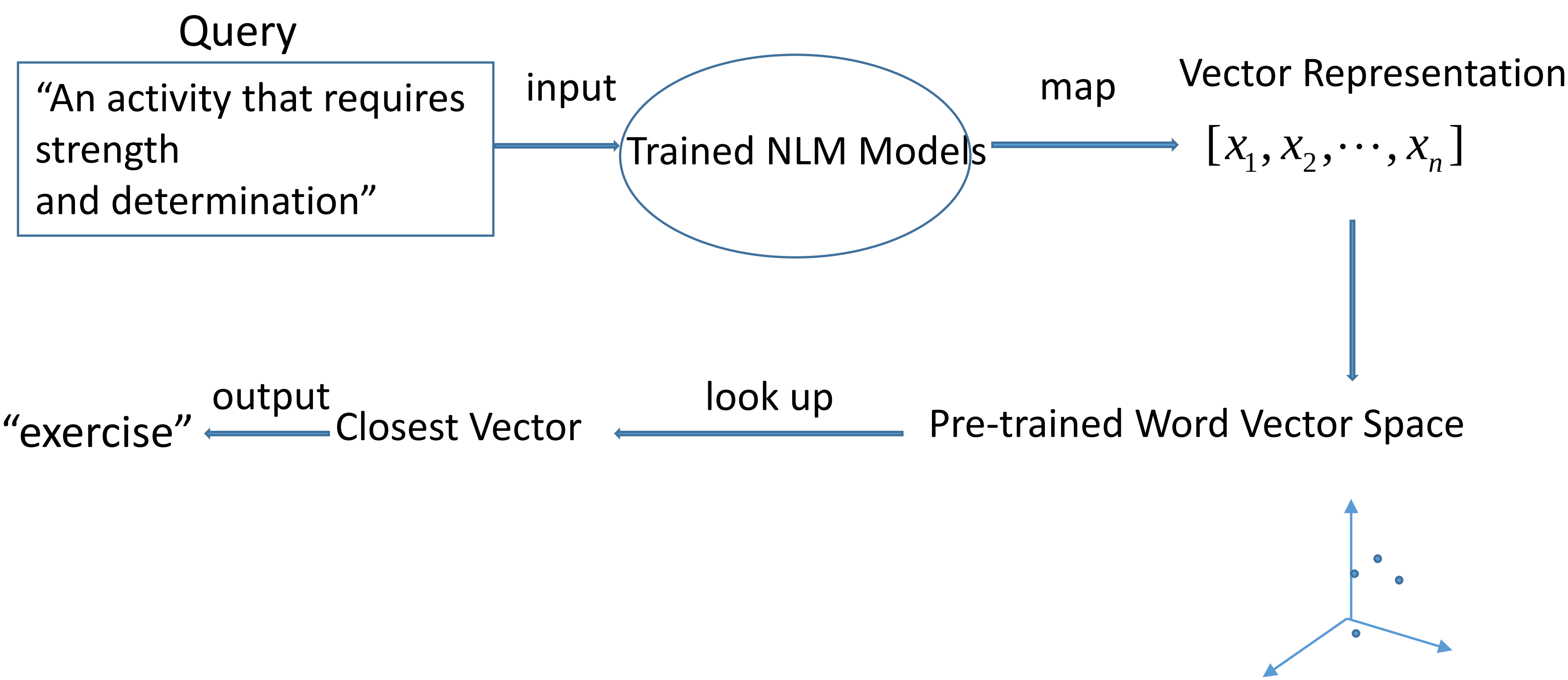
s_c : input phrase embedding

v_c : pre-trained embedding of defining word

v_r : randomly selected word from vocabulary

Phrase Embedding: Using a Dictionary

- Application: Reverse Dictionaries
  - Given a test description, definition, or question, all models produce a ranking of possible word answers based on the proximity of their representations of the input phrase and all possible output words.

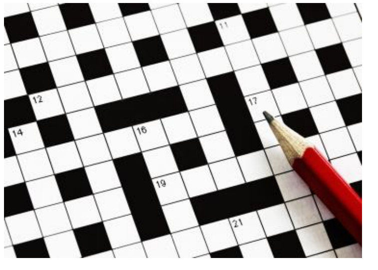


Phrase Embedding: Using a Dictionary

- Application: Crossword Question Answering
  - Given the absence of a knowledge base or web-scale information in our architecture, they narrow the scope of the task by focusing on general knowledge crossword questions

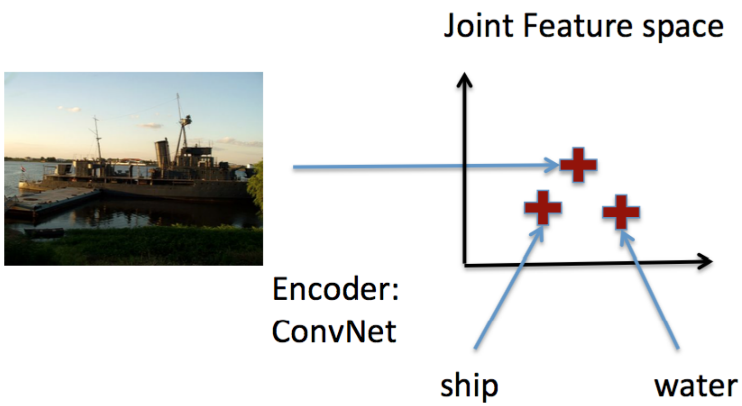
Test set	Description	Word
Long (150 Char)	“French poet and key figure in the development of Symbolism”	Baudelaire
Short (120 Char)	”devil devotee”	satanist
Single-Word (30 Char)	”culpability”	guilt

+ several constrains to reduce the target space



Learning to Understand Phrases by Embedding the Dictionary (Felix Hill et al. 2016)

Phrase Embedding: Using Images



A Deep Visual-Semantic Embedding Model, NIPS 2013

Zero-Shot Learning Through Cross-Modal Transfer, NIPS 2013

Caption: a girl in a blue shirt is on a swing

Keywords: girl, blue shirt, swing

Phrase Embedding: Using Images

- (image, query)

- But image maps to multiple queries

- (*image*, girl)
- (*image*, blue shirt)
- (*image*, swing)

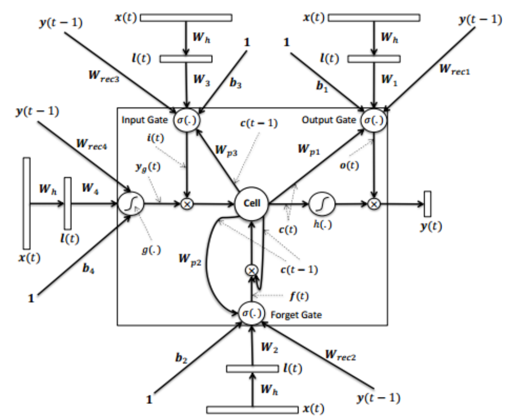
- The image places unnecessary constraint on the 3 queries.

Query Embedding: Using clicked data

Query Side: Shanghai Hotel

(CTR data indicates the semantic relation between Query Side and Document Side)

Document Side: “shanghai hotels accommodation hotel in shanghai discount and reservation”



Basic LSTM architecture for sentence embedding

Deep Sentence Embedding Using Long Short-Term Memory Networks, Palangi et al 2016

Latent Semantic Model with Convolutional-Pooling Structure
(Yelong Shen, et al. 2014)

Search Engine	Microsoft *office* excel
Search Engine	welcome to the apartment *office*

Query examples on internet

what's the meaning of *office* ?

Traditional method: Bag-of-Words — Contextual Information — *office 1 = office 2*

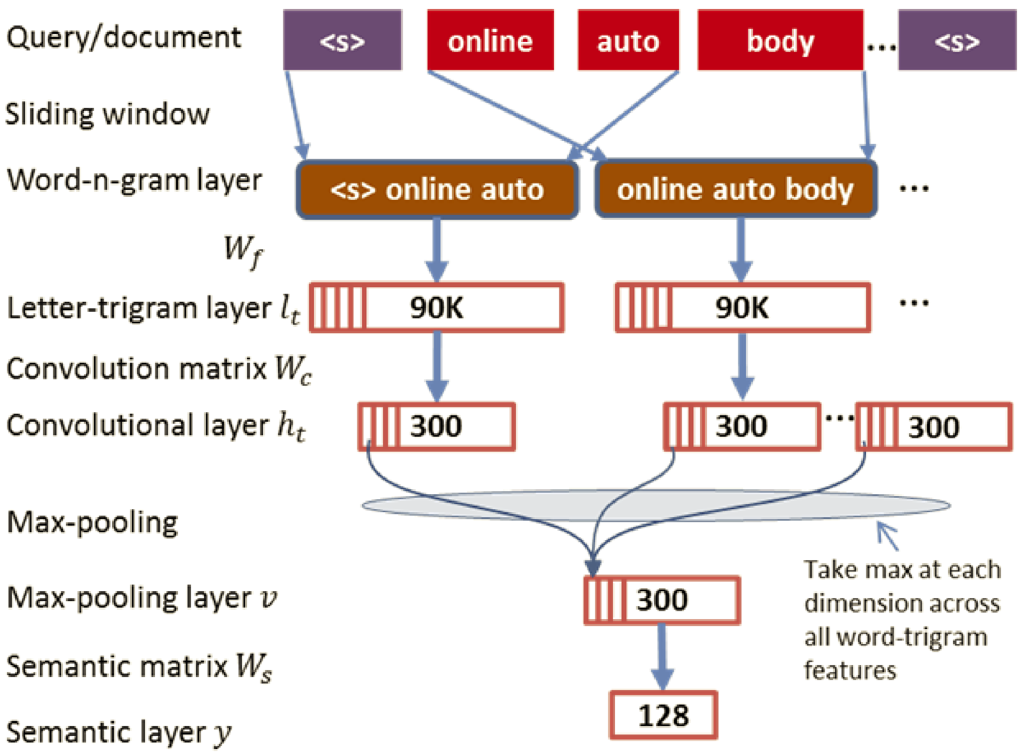
Word Sequence + convolutional-pooling structure — office 1 $\neq$ office 2

low-dimentional, semantic vector representations
for search queries and web document

Shen, Yelong, et al. "A latent semantic model with convolutional-pooling structure for information retrieval." Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management. ACM, 2014.

Latent Semantic Model with Convolutional-Pooling Structure
(Yelong Shen, et al. 2014)

- Models:



The CLSM maps a **variable-length** word sequence to **a low-dimensional vector** in a latent semantic space.

Latent Semantic Model with Convolutional-Pooling Structure

(Yelong Shen, et al. 2014)

- Models:

$$f(boy) = \begin{bmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 1 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix}$$

Indices of *#-b-o*, *b-o-y*, *o-y-#* in the letter-tri-gram list, respectively.

\# is word boundary symbol

Letter-trigram based Word-n-gram Representation

Concatenating

$$l_t = [f_{t-d}^T, \dots, f_t^T, \dots, f_{t+d}^T]^T, \qquad t = 1, \dots, T$$

Word trigram vector

Convolution operation

$$h_t = tanh(W_c \cdot l_t), \qquad t = 1, \dots, T$$

Variable length sequence of feature vectors

max pooling

$$v(i) = \max_{t=1,\dots,T}\{h_t(i)\}, \qquad i = 1, \dots, K$$

microsoft **office excel** could allow remote **code execution**
welcome to the **apartment office**
online **body fat** percentage **calculator**
online **auto body** repair **estimates**
vitamin a the **health** benefits given by **carrots**

Bold words win max operation

Latent Semantic Model with Convolutional-Pooling Structure

(Yelong Shen, et al. 2014)

- Models:

  - Latent Semantic Vector Representations

$$y = \tanh(W_s \cdot v)$$

v is the global feature vector after max pooling, W_s is the semantic projection matrix, and *y* is the vector representation of the input query.

Using cosine similarity to measure relatedness between queries and documents

$$R(Q, D) = \text{cosine}(y_Q, y_D) = \frac{{y_Q}^T y_D}{\| y_Q \| \| y_D \|}$$

Summary

- Bag of words not powerful enough unless we have huge amount of high quality pairs.

- Web queries are not phrases. Simple composition or phrase translation does not work for web queries.

- Sentiment, classification as targets are not powerful enough to capture full semantics.

- Translation is a better target, as it forces the representation to contain the full semantics.

Conclusion

For short text understanding:

- Short text’s understanding is still hard because the complexity meaning of combining the word in short text, the absence of certain context and syntactical structure.

- There are not very suitable embedding approaches. But just like Hamid’s work, we can incorporate some external data to help do the similarity measurement.

- Word Embedding can be a good feature but not the only feature, we can utilize more NLP tools such as POS or Entity Recognition to do disambiguation.

Reference

[Bengio et al. 2003] Yoshua Bengio, Réjean Ducharme ,Pascal Vincent, Christian Jauvin. A Neural Probabilistic Language Model. In Journal of Machine Learning Research 3 (2003) 1137–1155.

[Mikolov et al. 2013a] Mikolov, Tomas, Chen, Kai, Corrado, Greg, and Dean, Jeffrey. Efficient estimation of word representations in vector space. CoRR abs/1301.3781 (2013)

[Mikolov et al. 2013b] Mikolov, Tomas, et al. Distributed representations of words and phrases and their compositionality.In NIPS, 2013.

[Pennington et al. 2014]   J Pennington, R Socher, CD Manning. Glove: Global Vectors for Word Representation.In EMNLP 2014, 1532-1543.

[Socher et al. 2011]   Richard Socher, Eric H. Huang, Jeffrey Pennington, Andrew Y. Ng, Christopher D. Manning. Dynamic Pooling and Unfolding Recursive Autoencoders for Paraphrase Detection. In NIPS 2011: 801-809

[Cho et al. 2014] Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, Yoshua Bengio. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation.In EMNLP 2014: 1724-1734

[Gao et al. 2013] Jianfeng Gao, Xiaodong He, Wen-tau Yih, Li Deng:Learning Semantic Representations for the Phrase Translation Model. CoRR abs/1312.0482 (2013)

Reference

[Quoc et al. 2014] Quoc V. Le, Tomas Mikolov: Distributed Representations of Sentences and Documents.In ICML 2014: 1188-1196.

[Ryan Kiros et al. 2015] Kiros, Ryan, et al. Skip-thought vectors. In NIPS 2015.

[Felix Hill et al. 2016] Felix Hill, Kyunghyun Cho, Anna Korhonen, Yoshua Bengio. Learning to Understand Phrases by Embedding the Dictionary.In TACL 4: 17-30 (2016)

[Hamid et al. 2016] Hamid Palangi, Li Deng, Yelong Shen, Jianfeng Gao, Xiaodong He, Jianshu Chen, Xinying Song, Rabab K. Ward. Deep Sentence Embedding Using Long Short-Term Memory Networks: Analysis and Application to Information Retrieval. IEEE/ACM Trans. Audio, Speech & Language Processing 24(4): 694-707 (2016)

[Shen, et al. 2014]   Shen et al. A latent semantic model with convolutional-pooling structure for information retrieval. In CIKM 2014.

[Socher et al. 2012] R. Socher, B. Huval, C. Manning and A. Ng. Semantic Compositionality through Recursive Matrix-Vector Spaces. In EMNLP 2012.

[Socher et al. 2013a] R. Socher, J. Bauer, C. Manning and A. Ng. Parsing with Compositional Vector Grammars. In ACL 2013.

[Socher et al. 2013b] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. Manning, A. Ng and C. Potts. Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank. In EMNLP 2013.