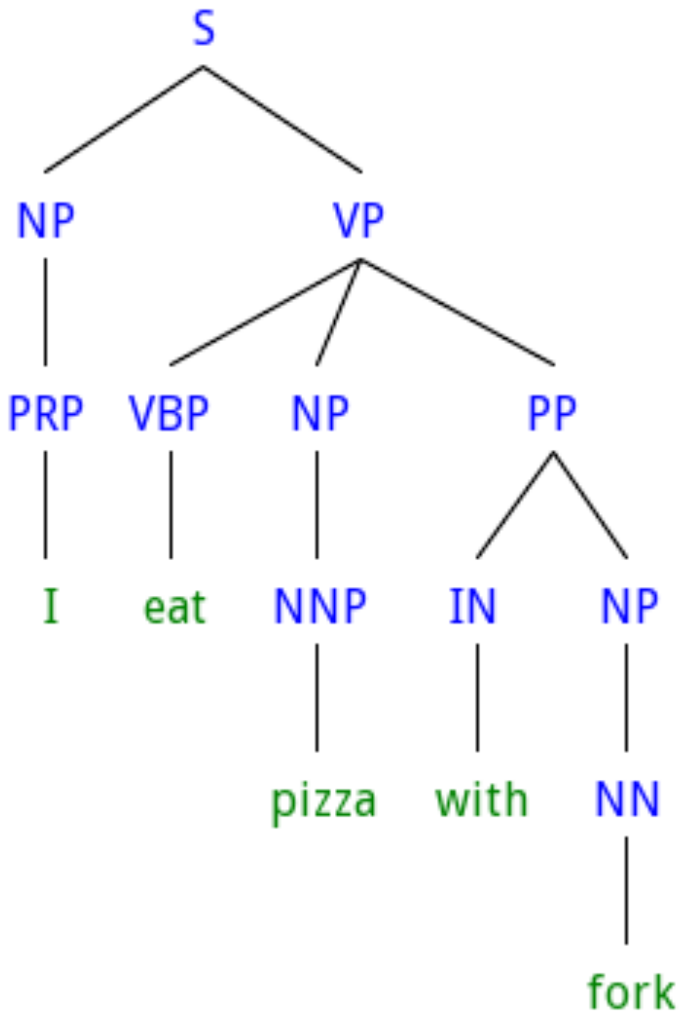
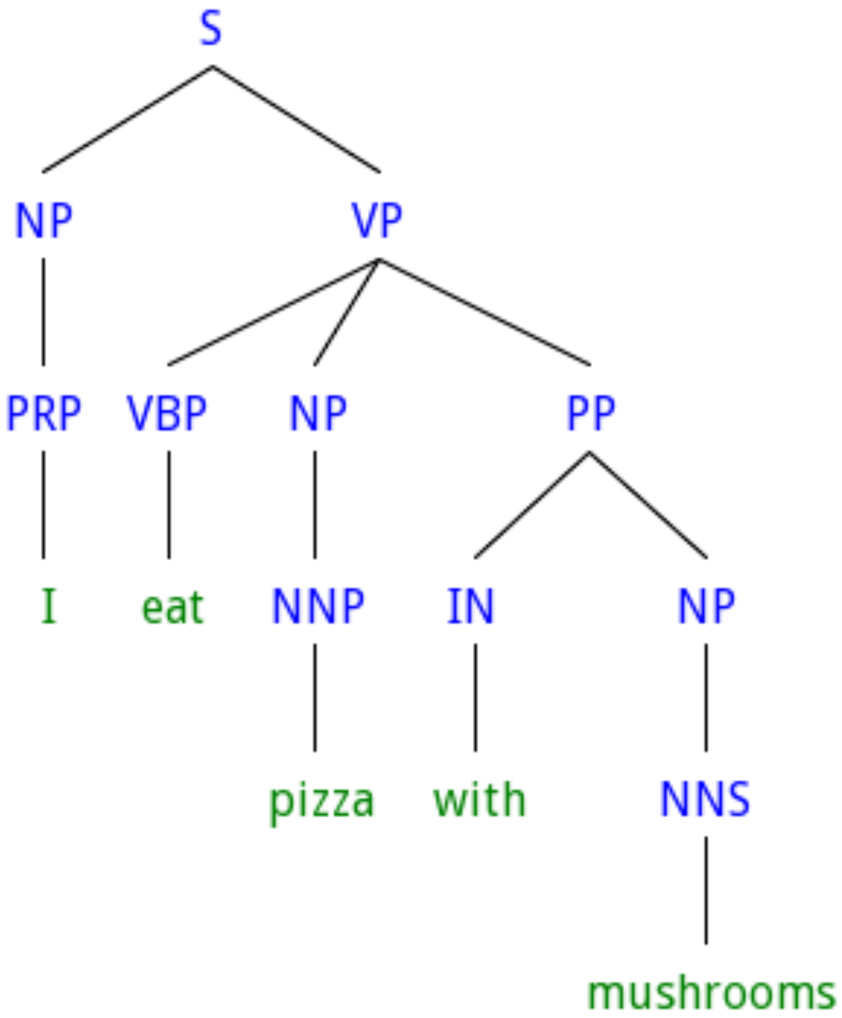
Syntactic parsing with Recursive Neural Networks

Agenda

1. Motivation
2. Review of word vectors
3. Recursive NN for syntactic parsing
4. Resources & summary


S
NP
VP
PRP
VBP
NP
PP
I
eat
NNP
IN
NP
pizza
with
NN
fork


S
NP
VP
PRP
VBP
NP
PP
I
eat
NNP
IN
NP
pizza
with
NNS
mushrooms

Motivation for research

Lexical information is needed for correct parsing

Lexicalization approaches

- Naive
  - add lexical information into syntactic category
- Discriminative parsing
- Advanced feature engineering
  - Refining each category: Ex. NP -> {NP-1, NP-2,NP-3,...}

Problem

- No clear way of representing category/phrase
- All approaches require complicated feature engineering process

Is it possible to learn features automatically?

Solution

1. Use continuous word vectors as input
2. Train recursive neural network for structure prediction

What we achieve

+ Only 1,9% behind Stanford Parser(2003)
+ Good results on short sentences
+ No manual engineering, No POS tagging

Word representation

Symbolic representation of the word(one-hot)

0	0	0	1	…	0	0	0

chopsticks

0	0	0	0	…	1	0	0

fork

Limitations of one-hot model

- Rare words in training data -> poorly estimated
- Not seen in training data -> model cannot handle it

Word vectors

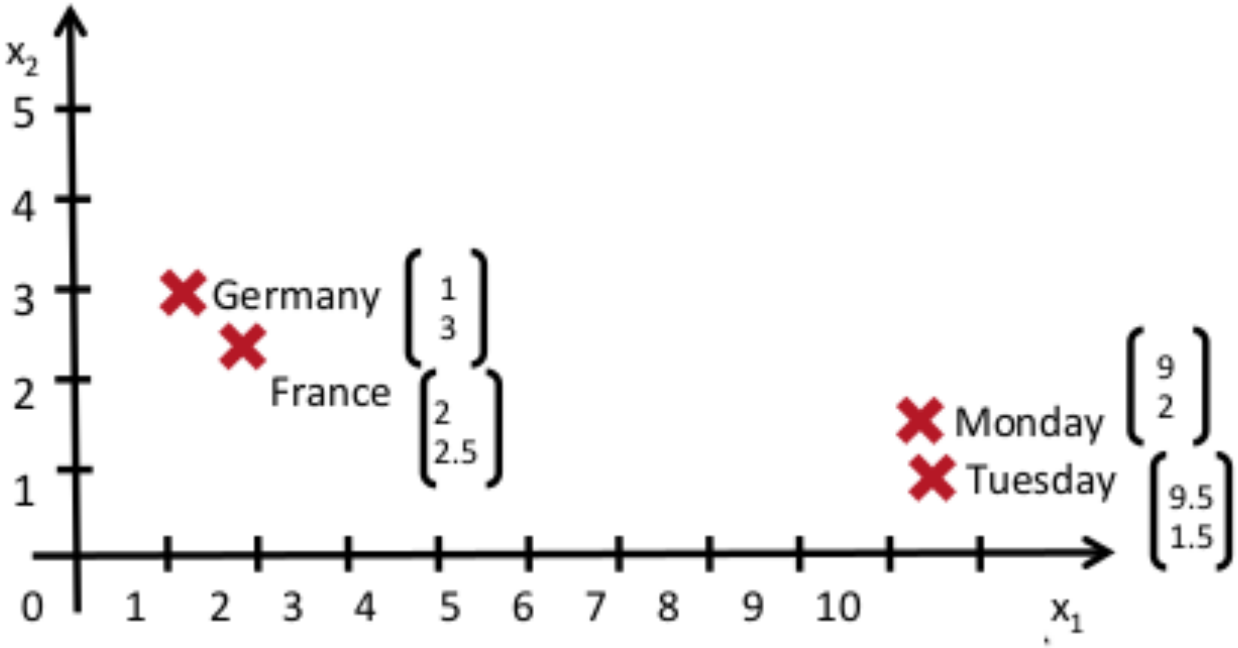


Figure from Y. Bengio tutorial 2012 “Recursive neural networks”

Problem

How to get representations for phrases if we have only word vectors?

Compositionality principle

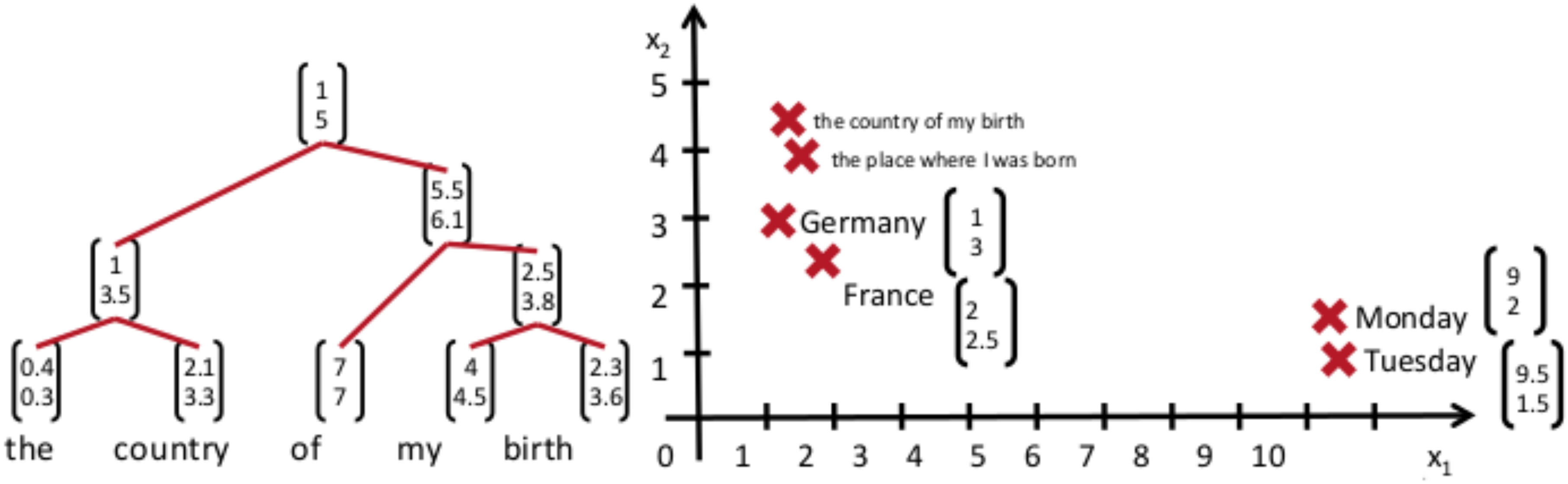


Figure from Y. Bengio tutorial 2012 “Recursive neural networks”

Recursive NN

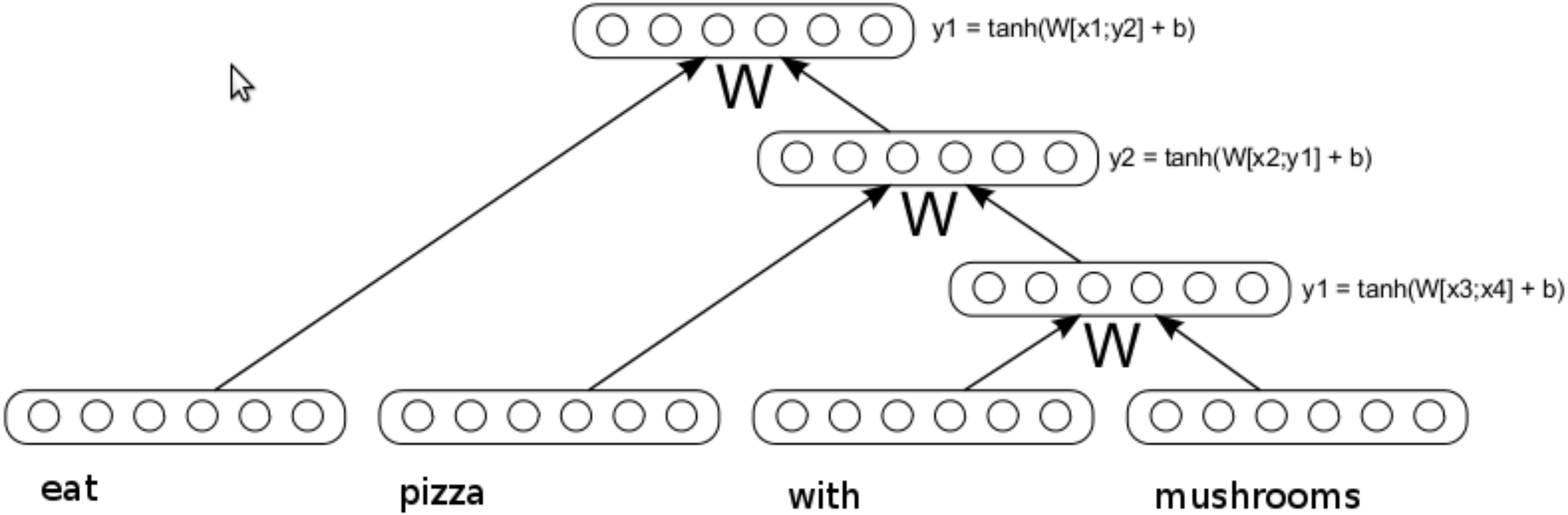
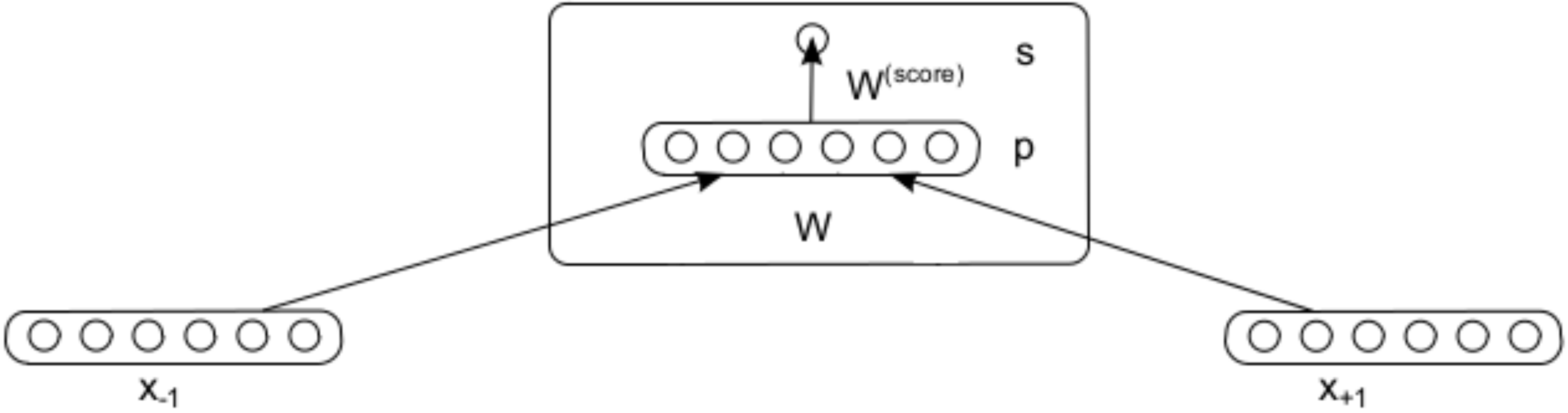


Figure from Socher et al. 2010

Architecture



$$p = \tanh(W[c_1; c_2] + b)$$

$$s_{1,2} = W^{score} p$$

Figure from Socher et al. 2010

Parsing a sentence

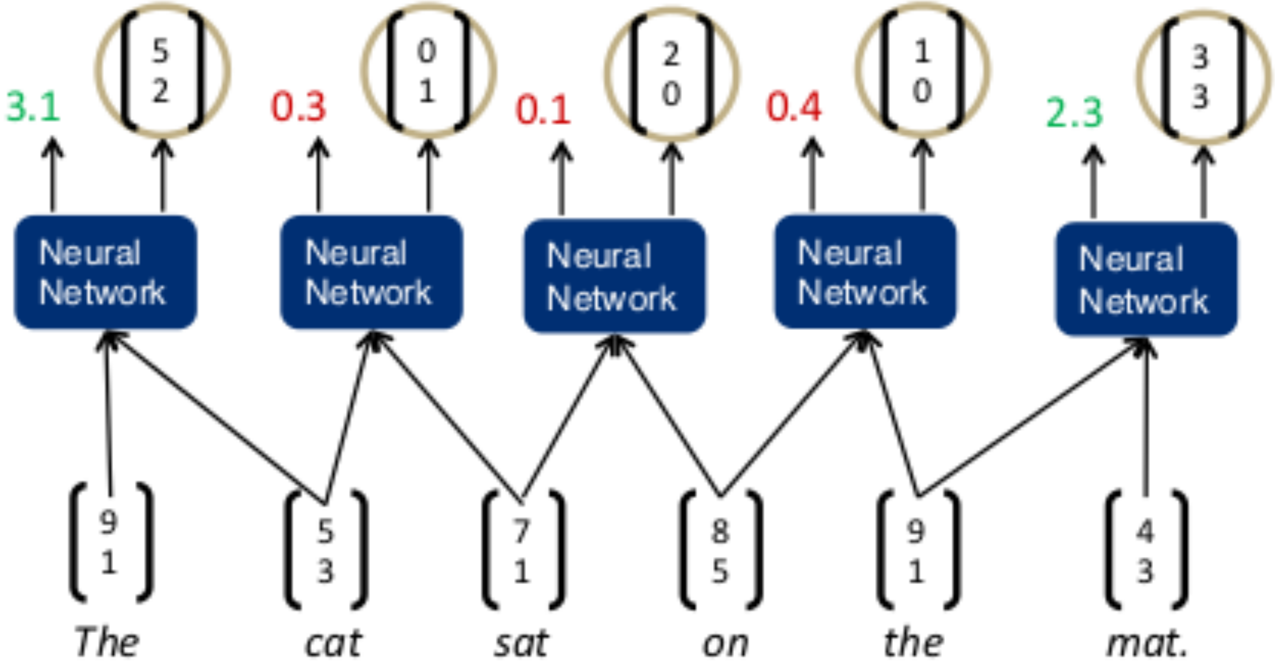


Figure from Y. Bengio tutorial 2012 “Recursive neural networks”

Parsing a sentence

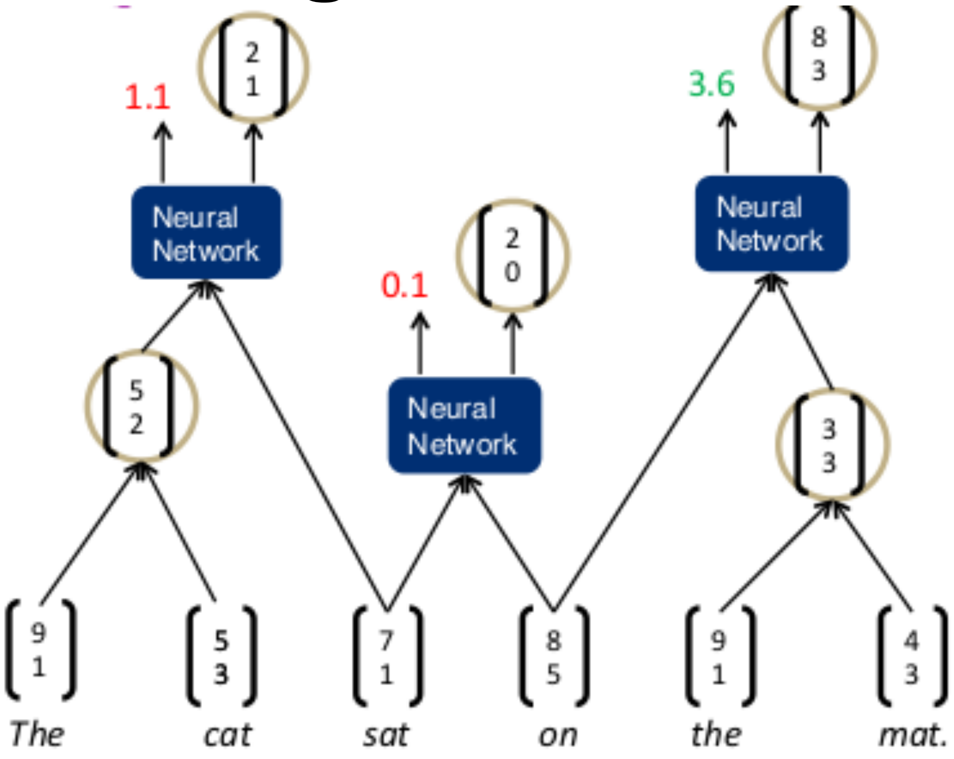


Figure from Y. Bengio tutorial 2012 “Recursive neural networks”

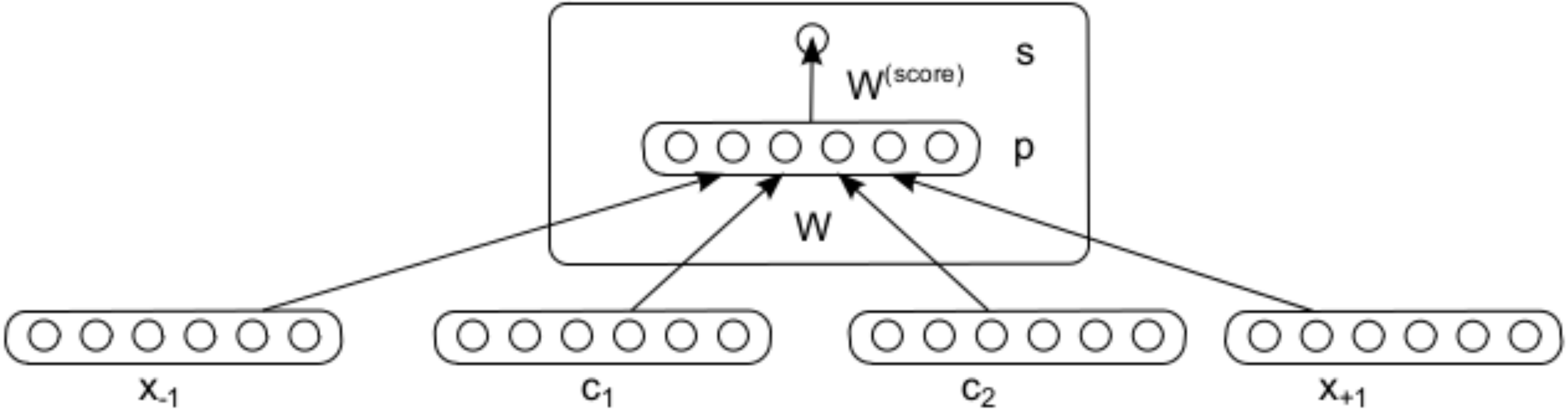
Training RNN

- Collect correct parse trees from the corpora
- For each collapses of 2 words into phrase we assign score:
    - Correct collapse -> higher score
    - Incorrect collapse -> lower score
- Thats how we learn W and Wscore

Some types of RNN

1. Greedy RNN
2. Context aware RNN
3. Context aware RNN + category classifier
4. Max-Margin Framework with Beam-Search

Context aware RNN



$$s = W^{score} p$$

$$p = \tanh(W[x_{-1}; c_1; c_2; x_{+1};] + b^{(1)})$$

Figure from Socher et al. 2010

Materials

1. CW08 “A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning”
2. TRB10, “Word representations: A simple and general method for semi-supervised learning”

Summary

- Learned how to represent phrases having only word vectors
- Build framework for syntactic parsing
  - Can be used also for paraphrase detection