Applications

ADVANCED ARTIFICIAL INTELLIGENCE

JUCHEOL MOON

1

Neural Language Models

- NLMs are a class of language model designed to overcome the curse of dimensionality problem for modeling natural language sequences by using a distributed representation of words

one-hot-vector representation of 1B words of bag

$$\text{dog} \Rightarrow [\underbrace{0\ 0 \cdots 0\ 1\ 0\ 0 \cdots 0}_{1,000,000,000}]^T$$

$$\text{cat} \Rightarrow [0\ 0 \cdots 0\ 0\ 0\ 1 \cdots 0]^T$$

- Neural language models share statistical strength between one word (and its context) and other similar words and contexts

2

2

The distributed representation

- The language model learns treat words that have features in common similarly
  - GRE Analogies
    - ADULT : CHILD::
    - (A) horse : mare
    - (B) cat : kitten
    - (C) swine : sow
    - (D) human : animal
    - (E) cow : herd

- ENVELOPE : LETTER ::
  - (A) scarf : hat
  - (B) box : bag
  - (C) crate : produce
  - (D) neck : head
  - (E) blood : heart
- OVERDOSE : PRESCRIPTION ::
  - (A) deprivation : materialism
  - (B) indiscretion : convention
  - (C) affliction : sympathy
  - (D) adventure : expedition
  - (E) drug : medicine

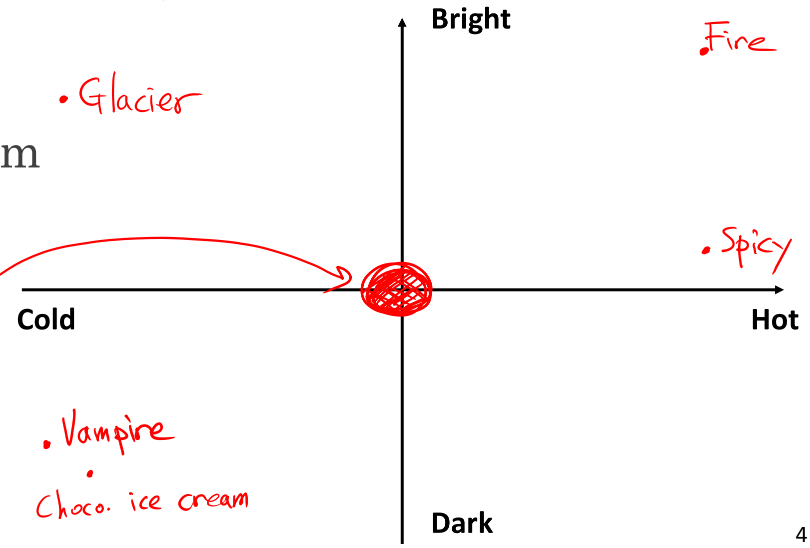
d

3

3

Word Embeddings

- Can we locate the words by their characteristics?
  - Fire
  - Spicy
  - Chocolate ice cream
  - Vampire
  - Glacier
  - Jucheol Moon
  - Gray
  - Final exam

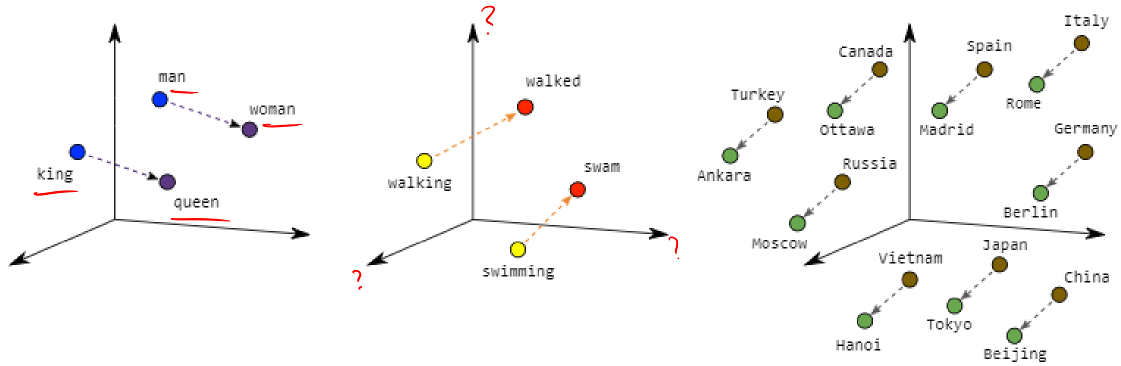


4

4

Word Embeddings

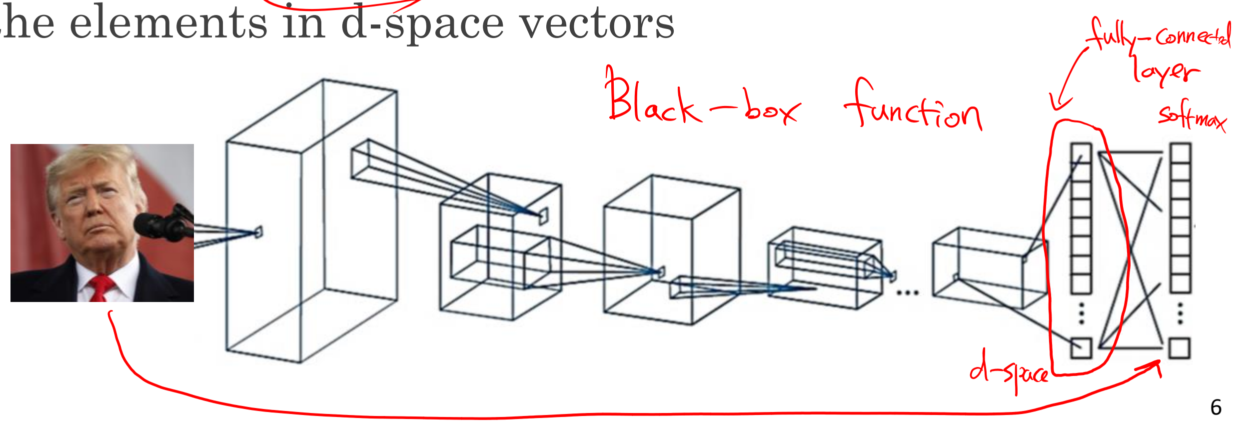
- In the embedding space, words that frequently appear in similar contexts (or any pair of words sharing some "features" learned by the model) are close to each other.



Male-Female Verb Tense Country-Capital

Embedding in Vision

- A hidden layer of a convolutional network provides an "image embedding" in d-space
- We (could / could not) interpret the characteristics of the elements in d-space vectors



Word Embedding

- Assume we are using word embedding vectors in d-space
  - how → $e_{how} = [0.1 \quad -0.3 \quad 0.9 \quad \cdots]^T$
  - are → $e_{are} = [0.4 \quad 0.1 \quad -0.7 \quad \cdots]^T$
  - you → $e_{you} = [0.01 \quad 0.2 \quad -0.3 \quad \cdots]^T$

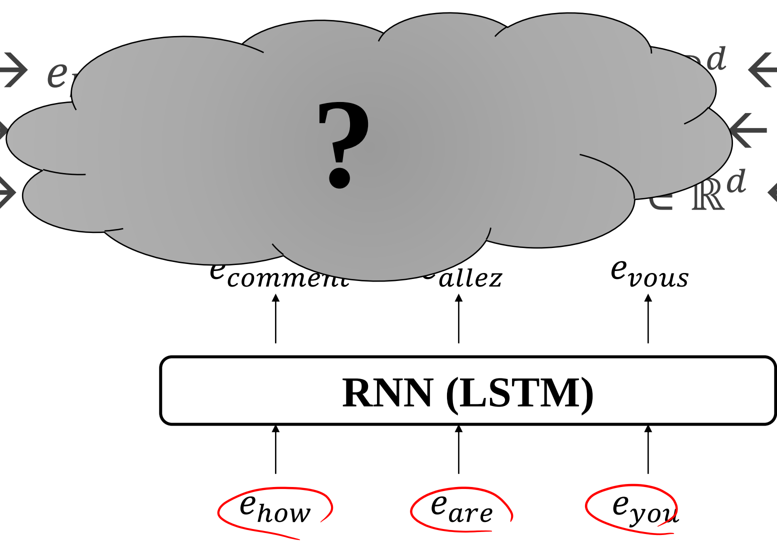
$e_{comment}$ e_{allez} e_{vous}

e_{how} e_{are} e_{you}

7

Word Embedding

- Assume we are using word embedding vectors in d-space
  - how → e ... ^{d} ← comment
  - are → ... ← allez
  - you → ... $\in \mathbb{R}^d$ ← vous



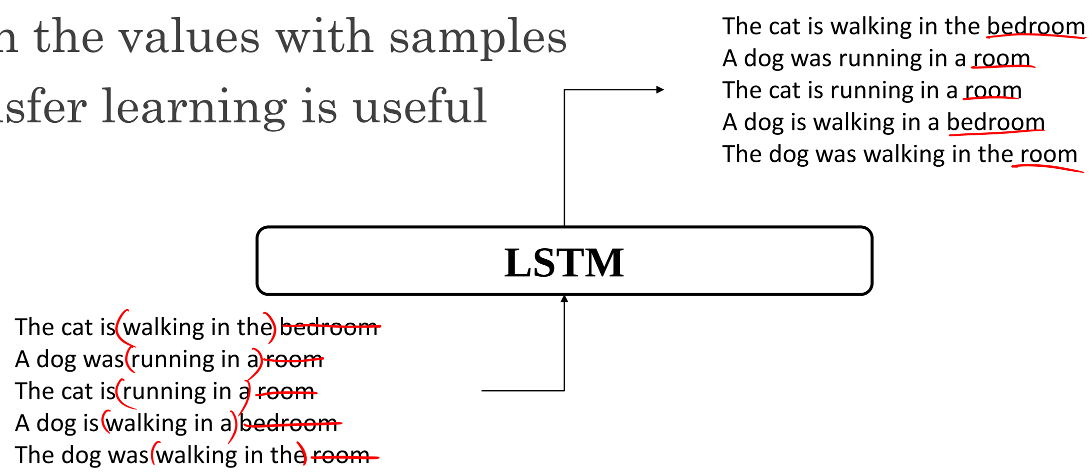
8

n-grams

- The earliest successful language models were based on models of fixed-length sequences of tokens called n-grams. An n-gram is a sequence of n tokens.
- Ex) Can we predict the blanks?
  - The cat is walking in the bedroom
  - A dog was running in a room
  - The cat is running in a room
  - A dog is walking in a bedroom
  - The dog was walking in the room

3-grams Learning

- Initially, (as usual) set the values of embedding vectors randomly
- Train the values with samples
- Transfer learning is useful



Negative sampling

- Training data set includes only positive samples
  - The cat is **walking in the** bedroom.
  - The dog was **walking in the** room.
- We need to have negative samples
  - The cat is **walking in the** __drawer__.
  - The dog was **walking in the** __fridge__.
- Total number of negative samples is similar to the number of positive samples
- Negative sampling is proportion to the frequency of the words

11

11

Sentiment Analysis

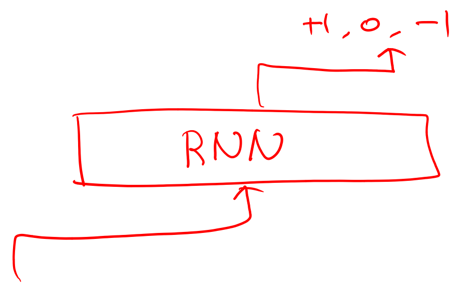
- Determining whether a comment expresses positive or negative sentiment.
  - All of the car-based crap is just a huge product placement for Audi, and it really hurts credibility.
  - The movie is full of other great visuals like V.I.K.I., the whole robotized city of Chicago, and of course Smith's cool Audi.

12

12

RNN for sentiment analysis

- A sentiment predictor trained on customer reviews of media content.
- There is an underlying function that tells whether any statement is positive, neutral, or negative



All of the car-based crap is just a huge product placement for Audi, and it really hurts credibility.

Context-Free grammars

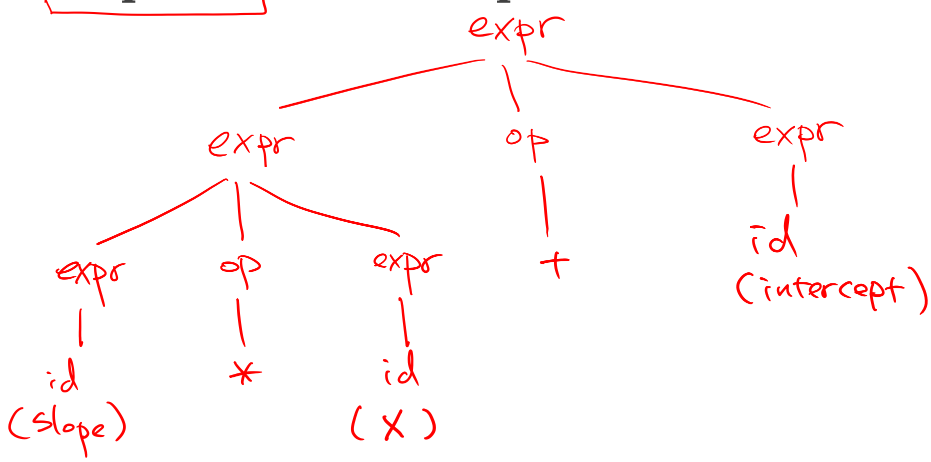
- The notation for context-free grammars (CFG) is sometimes called Backus-Naur Form (BNF)
- A CFG consists of
  - A set of terminals T
  - A set of non-terminals N
  - A start symbol S (a non-terminal)
  - A set of productions X → Y

Parse tree

$expr \longrightarrow$ `id` | `number` | `-` $expr$ | `(` $expr$ `)`
| $expr\ op\ expr$

$op \longrightarrow$ `+` | `-` | `*` | `/`

- Parse tree for expression grammar for
  "slope * x + intercept"



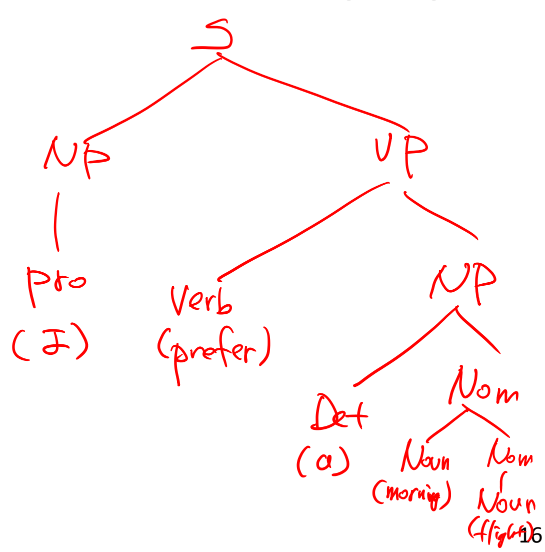
15

15

CFG

- S → NP VP
- NP → Pronoun
  | Proper-Noun
  | Det Nominal
- Nominal → Noun Nominal
  | Noun
- VP → Verb | Verb NP
  | Verb NP PP
  | Verb PP
- PP → Preposition NP

- I prefer a morning flight

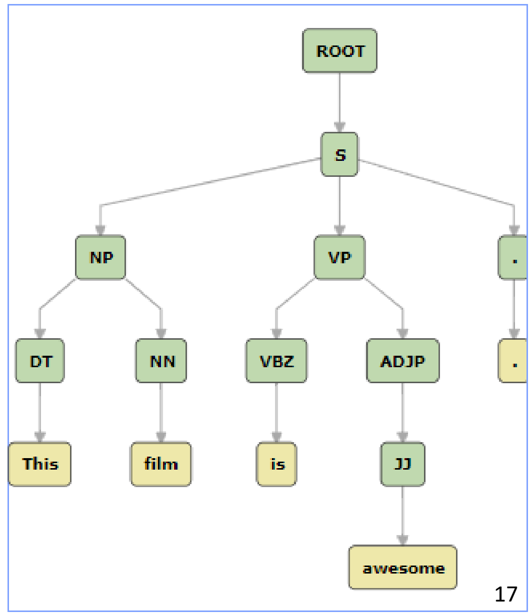


16

16

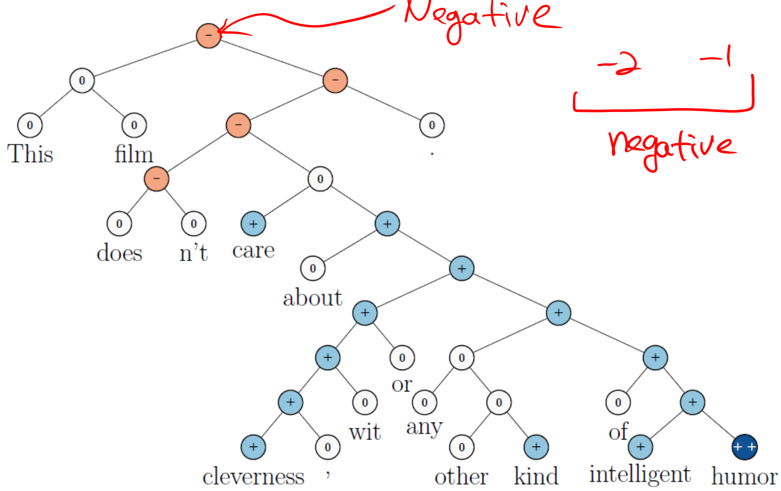
Recursive Neural Network

- Recursive neural networks represent another generalization of recurrent networks which is structured as a deep tree
  - This film is awesome.



Recursive Neural Network

- This file doesn't care about cleverness, wit or any other kind of intelligent humor.



Negative

$\underbrace{-2 \quad -1}_{\text{negative}} \quad 0 \quad \underbrace{+1 \quad +2}_{\text{positive}}$

Recursive Neural Network

- To classify d-dimensional semantic vectors into five sentiment classes, the posterior probability is computed by $y_j = \frac{e^{(W_s \vec{x})_j}}{\sum_{i=1}^{5} e^{(W_s \vec{x})_i}}$

$|W_s| = 1 \times d$

$|\vec{x}| = d \times 1$

- The semantic vectors of internal nodes are computed by

$$\vec{x}_p = \tanh\left(\begin{bmatrix}\vec{x}_L \\ \vec{x}_R\end{bmatrix}^T \vec{V} \begin{bmatrix}\vec{x}_L \\ \vec{x}_R\end{bmatrix} + \vec{W}\begin{bmatrix}\vec{x}_L \\ \vec{x}_R\end{bmatrix}\right)$$