Lecture 1. Intro & Word Vectors

Human Language and Word Meaning

- Human Language

- Rather recent key for humans to become invincible (In evolutionary terms)
- Pathway of Knowledge by 'writing'
- Adaptive form of compression (ex. simple sentences make ppl visualize corresponding circumstance)

Human Computer Network → use human language as network languages

- Word Meaning

→ Idea that is represented by a word, phrase … etc
→ what things represent (ex. chair) [Denotational Semantics]

Word2vec Intro

- WordNet
  - Thesaurus containing lists of synonyms sets and hypernyms (is-a relationship - hierarchy)
  - Used Tools) NLTK → Similar to Swiss Army Knife (Not terribly good but okay)

- WordNet - Problems

- Great as resource but missing nuance
- Requires human labor to create & adapt → can't compute accurate word similarity
→ fixed discrete synonym sets (can't measure partial resemblance btw synonyms)

- Denotational Representations

Traditional NLP - up to 2012

- Localist Representation: Represented by one-hot vectors
- No similarity relationships (orthogonal vectors)
- Tried to build table of word similarities
→ impossible to do (due to volume of vocab)

- Distributional Representations

Based on Distributional Semantics
→ Word meaning: Defined by the context it is used
- most successful idea in modern NLP
- key idea on Word2vec
- smaller size compared to localist rep (300D)

- QnA

- Dimensions of word vectors contain meaning
- closeness of vectors represent **similarity**
- Vector Dimensions & Directions in vector space contain meaning

Word2vec Overview & Objective Function Gradients

- Word2vec

- Framework for learning word vectors

1. Dataset: Big pile of continuous text (Corpus)
2. Objective: The center word being able to predict the words in the context fairly well (& vice versa)

- Very loose model

→ Due to the fact that it captures all of the words in the window size of the context in one trial

- Likelihood

- How good the job is at predicting the context of the center word
- depend on the parameter (Only one parameter in this case)

$$L(\theta) = \prod_{t=1}^{T} \prod_{\substack{-m \le j \le m \\ j \ne 0}} P\left(w_{t+j} \mid w_t; \theta\right)$$

- Probability Equation

- Softmax Equation
  - putting weight where ‘the max’ is (or list of maxes)
  - u: context word / v: center word

$$P(O = o \mid C = c) = \frac{\exp(\boldsymbol{u}_o^\top \boldsymbol{v}_c)}{\sum_{w \in \text{Vocab}} \exp(\boldsymbol{u}_w^\top \boldsymbol{v}_c)}$$

- Objective Function (cost / loss function)

- (average) negative log likelihood

$$J(\theta) = -\frac{1}{T}\log L(\theta) = -\frac{1}{T}\sum_{t=1}^{T}\sum_{\substack{-m \le j \le m \\ j \ne 0}} \log P\left(w_{t+j} \mid w_t; \theta\right)$$

- represent each word with 2 vectors
    - context & center

Minimizing objective function = Maximizing predictive accuracy

Optimization Basics

- Minimizing Objective function

Method: by calculating derivative of the objective function and updating the variable (theta)

- Resulting Derivative

$$\frac{\partial J(\theta)}{\partial v_c} = -u_o + \sum_{x=1}^{v} P(x|c) * u_x$$

- 1st element: current rep of context word
- 2nd element: expectation of what the model should look like