1. 자연어 처리의 시작

자연어 처리 활용 분야와 트렌드

Academic Disciplines related to NLP

NLP (major conference: ACL, EMNLP, NAACL)

- Low-level parsing
  - Tokenization: 주어진 문장을 단어 단위로 쪼갬
  - Stemming: studying, studied 처럼 어미의 변화가 있어도 단어의 의미를 이해하는 것. 단어의 어근 추출하는 것
- Word and pharse level
  - Named entity recognition(NER): 단일 단어 또는 여러 단어로 이루어진 고유명사 인식
  - part-of-speech (POS) tagging: 단어의 품사나 성분이 무엇인지 알아냄 (ex. 주어, 목적어 등)
- Sentence level
  - sentiment analysis: 긍부정 판단
  - machine translation: 기계 번역. 적절한 단어와 문법 선택
- Multi-sentence and paragraph level
  - Entailment prediction: 문장 간 논리 분석, 예측
  - question answering: 질문을 이해하고 답 제공
  - dialog systems: 챗봇
  - summarization: 주어진 문서 요약

Text mining (major conferences: KDD, The WebConf, WSDM, CIKM, ICWSM)

- 키워드 관련 데이터를 수집하고 트렌드 분석

- Document clustering
  - topic modeling
  - 문서 군집화
- Highly related computational social science
  - 사회 과학과 관련 깊음
  - e.g. 소셜 미디어 데이터에 기반해 사람들의 성향 분석

Information retrieval (major conference: SIGIR, WSDM, CIKM, RecSys)

- 정보 검색 기술
  - 이미 많이 연구됨
  - 추천시스템

Trends fo NLP

- word embedding
  - 시퀀스 데이터를 단어 단위로 분리하고 단어를 숫자로 변환(벡터화)
- RNN 계열 모델 (LSTMs, GRUs) 많이 사용했었음
- Transformer models
  - replace RNNs with self-attention
  - 기계번역을 위해 고안되었으나 현재는 영상/신약개발/시계열 예측 등에서도 사용됨
- 자가지도 학습 (self-supervised training)
  - BERT, GPT-3

기존의 자연어 처리 기법

Bag-of-Words

Bag-of-Words Representation

- 단어들의 순서는 고려하지 않고 단어의 출현 빈도에만 집중하는 텍스트 데이터의 수치화 표현 방법
- Step 1. Constructing the vocabulary containing unique words
- Step 2. Encoding unique words to one-hot vectors
    - A sentence/document can be represented as the sum of one-hot vectors

NaiveBayes Classifier for Document Classification

Bayes' Rule Applied to Documents and Classes

- document에서 클래스가 나타날 확률과 클래스가 정해져 있을 때 각 단어가 나올 확률 추정 가능

- Example
    - For a document *d*, which consists of sequence of words *w*, and a class *c*

	Doc(d)	Document (words, w)	Class (c)
Training	1	Image recognition uses convolutional neural networks	CV
	2	Transformer can be used for image classification task	CV
	3	Language modeling uses transformer	NLP
	4	Document classification task is language task	NLP
Test	5	Classification task uses transformer	?

- $P(c_{CV}) = \frac{2}{4} = \frac{1}{2}$
- $P(c_{NLP}) = \frac{2}{4} = \frac{1}{2}$

- **Example**
  - For each word w_i, we can calculate conditional probability for class c
    - $P(w_k|c_i) = \frac{n_k}{n}$, where n_k is occurrences of w_k in documents of topic c_i

Word	Prob	Word	Prob
$P(w_{\text{"classification"}}\|c_{CV})$	$\frac{1}{14}$	$P(w_{\text{"classification"}}\|c_{NLP})$	$\frac{1}{10}$
$P(w_{\text{"task"}}\|c_{CV})$	$\frac{1}{14}$	$P(w_{\text{"task"}}\|c_{NLP})$	$\frac{2}{10}$
$P(w_{\text{"uses"}}\|c_{CV})$	$\frac{1}{14}$	$P(w_{\text{"uses"}}\|c_{NLP})$	$\frac{1}{10}$
$P(w_{\text{"transformer"}}\|c_{CV})$	$\frac{1}{14}$	$P(w_{\text{"transformer"}}\|c_{NLP})$	$\frac{1}{10}$

We calculate the conditional probability of the document for each class

We can choose a class that has the highest probability for the document

$$P(c_{CV}|d_5) = P(c_{CV}) \prod_{w \in W} P(w|c_{CV}) = \frac{1}{2} \times \frac{1}{14} \times \frac{1}{14} \times \frac{1}{14} \times \frac{1}{14}$$

$$P(c_{NLP}|d_5) = P(c_{NLP}) \prod_{w \in W} P(w|c_{NLP}) = \frac{1}{2} \times \frac{1}{10} \times \frac{2}{10} \times \frac{1}{10} \times \frac{1}{10}$$

Word	Prob	Word	Prob
$P(w_{\text{"classification"}}\|c_{CV})$	$\frac{1}{14}$	$P(w_{\text{"classification"}}\|c_{NLP})$	$\frac{1}{10}$
$P(w_{\text{"task"}}\|c_{CV})$	$\frac{1}{14}$	$P(w_{\text{"task"}}\|c_{NLP})$	$\frac{2}{10}$
$P(w_{\text{"uses"}}\|c_{CV})$	$\frac{1}{14}$	$P(w_{\text{"uses"}}\|c_{NLP})$	$\frac{1}{10}$

Word Embedding - Word2Vec

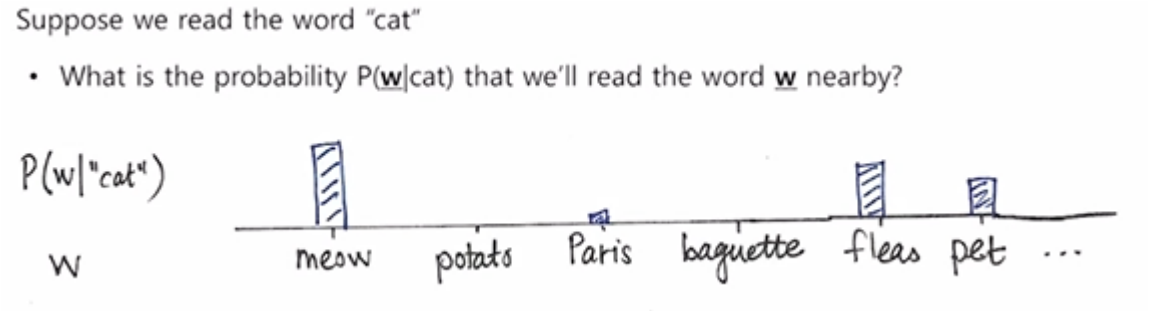
What is Word Embedding?

- Express a words as a vector
- 비슷한 의미의 단어가 좌표 상에서 비슷한 위치를 가짐
  - ex. ‘cat’ and ‘kitty’ are similar wors, so they have similar vector representations → short distance

- ex. ‘hamburger’ is not similar with ‘cat’ or ‘kitty’, so they have different vector representations → far distance

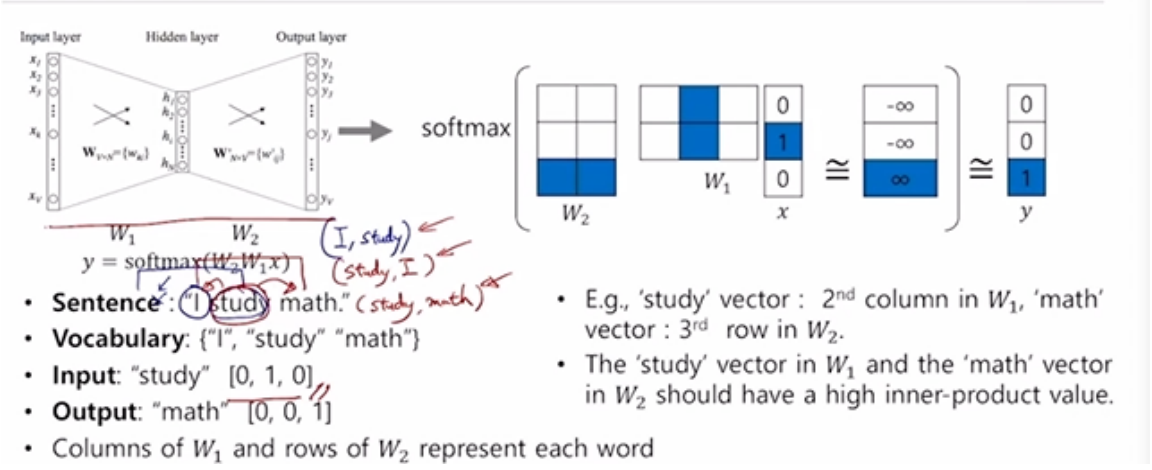
Word2Vec

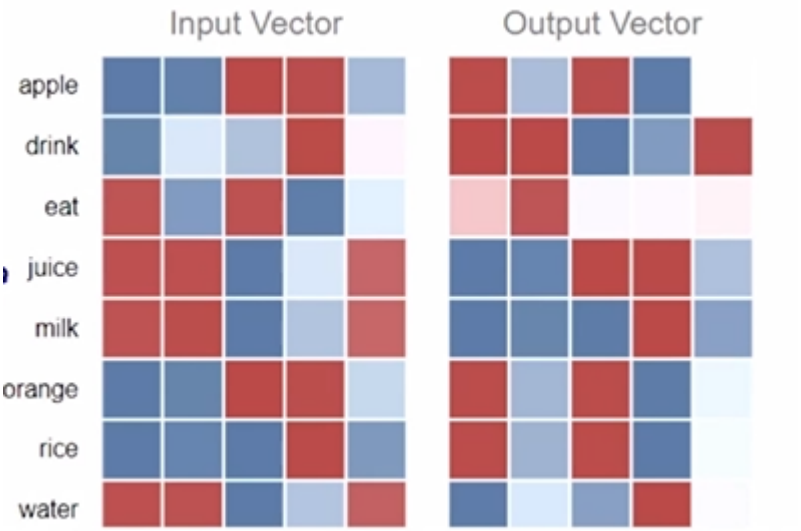
- 같은 문장에서 나타나는 인접한 단어들이 유사한 의미를 가질 것이라고 가정
  - ex. The cat purrs.
  - ex. This cat hunts mice.
- 주어진 데이터를 바탕으로 한 단어의 주변에 나타나는 단어들의 확률분포를 예측



How Word2Vec Algorithm Works

- Step 1. Tokenization 후 사전 구성
- Step 2. 문장을 one-hot vector 형태로 나타냄
- Step 3. 문장별로 sliding window을 적용해 입출력 단어 쌍 만듦
- Step 4. 만들어진 입출력 단어 쌍들에 대해 예측을 진행하는 neural net을 만듦





→ juice와 drink, milk, water가 유사한 벡터를 가짐

Property of Word2Vec

- The word vector, or the relationship between vector points in space, represents the relationship between the words
- The same relationship is represented as the same vectors
- 각 단어들간의 의미론적인 관계 학습
- Intrusion Detection
    - 가장 상이한 단어 찾기 가능

Application of Word2Vec

- Word similarity
- Machine translation
- PoS tagging
- 고유명사 인식
- 감정 인식
- Image Captioning

Word Embedding - GloVe

GloVe

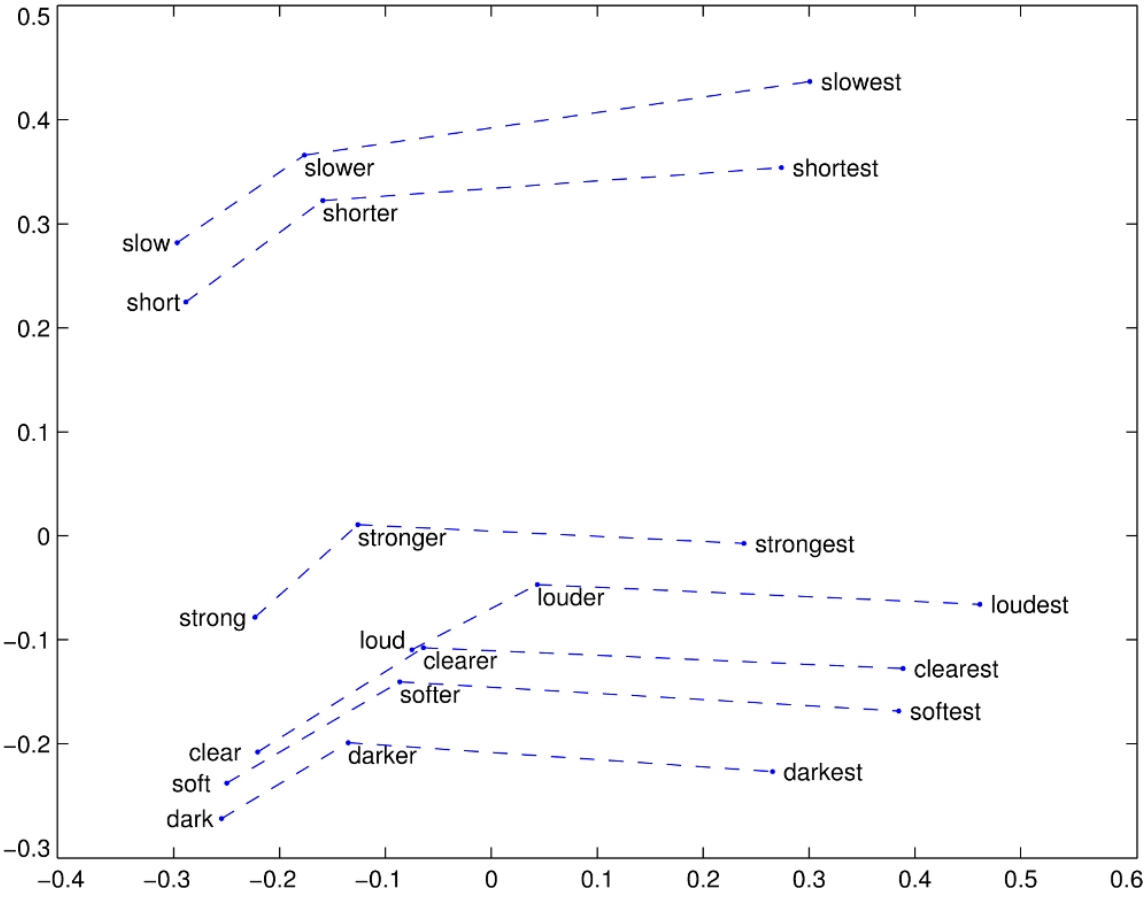
- First computes the co-occurrence matrix, to avoid training on identical word pairs repetively
- Loss function

$$J(\theta) = \frac{1}{2}\sum_{i,j=1}^{W} f(P_{ij})(u_i^T v_j - \log P_{ij})^2$$

- 특정한 입출력 단어쌍이 등장하는 횟수를 먼저 계산해 학습에 사용 → 중복된 계산 방지
- word2vec보다 빠른 학습, 적은 데이터에도 잘 학습함

Propery of Glove

- Linear Substructure


slowest
slower
shortest
shorter
slow
short
stronger
strongest
louder
loudest
strong
loud
clearer
clearest
softer
softest
clear
darker
darkest
soft
dark
0.5
0.4
0.3
0.2
0.1
0
−0.1
−0.2
−0.3
−0.4
−0.3
−0.2
−0.1
0
0.1
0.2
0.3
0.4
0.5
0.6