

자연어처리와 Transformer

2021.6.2~3

윤형기 (hky@openwith.net)

- Intro
 - AI와 NLP
- NLP 개요
 - 일반론: 전처리
 - 전통적 기법
 - Word Embedding
 - RNN/LSTM
 - RNN + CRF
- Transformer 혁명 – BERT와 GPT
 - BERT
 - GPT
- 전처리(Text preprocessing)
 - 01) 토큰화(Tokenization)
 - 단어 (Word) 토큰화
 - 문장 (Sentence) 토큰화
 - 품사 태깅(POS tagging)
 - NER
 - 02) 정제와 정규화
 - 대, 소문자 통합
 - 불필요한 단어 제거
 - Rare words 제거
 - short length words 제거
 - 03) 어간 및 표제어 추출
 - 04) 불용어(Stopword) 제거

INTRO

NLP 개요

NLP 일반론

- 전통적 기법
 - WordNet과 Thesaurus
 - 통계기반 기법
 - 분포 가설
 - Co-occurrence matrix
 - Vector 유사도
 - 상호정보량 (Mutual Information)
 - 차원 감소와 SVD
- Word Embedding: Word2Vec
 - 추론 기반 기법과 신경망
 - 신경망에서의 단어처리
 - Simple word2vec
 - CBOW와 Skip-Gram
 - Word2Vec 개선
 - word2vec 성능 개선
 - Negative Sampling
- RNN
 - RNN?
 - Language Model과 RNNLM
 - LSTM 기반의 LM
 - RNN 이용 문장 생성
 - seq2seq
 - Encoder와 Decoder class
- Transformer
 - Attention과 Transformer

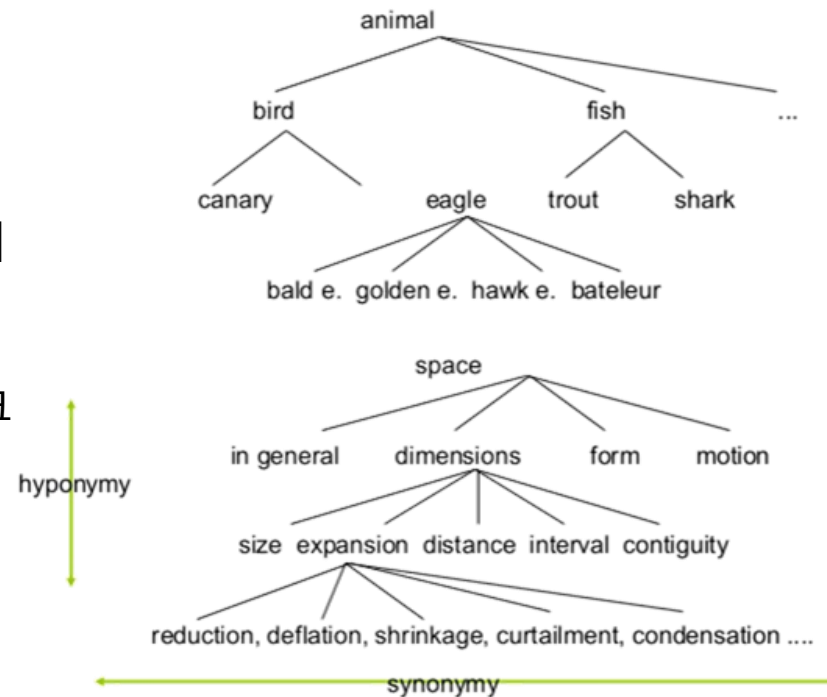
NER 일반론

- Rule 기반
- Bi-LSTM
- Bi-LSTM + CRF
- BERT 이용

NLP의 전통적 기법

WordNet과 Thesaurus

- 사전
 - 사전의 어휘는 사전 속에서 독립적으로 존재
- 시소러스(Thesaurus)
 - 유의어 사전
 - 색인언어의 어휘집으로, 개념 간 관계를 체계적으로 명시
 - WordNet
 - 어휘와 어휘 간 의미 관계를 네트워크로 나타낸 것
 - Princeton university, 1985
 - Thesaurus의 문제점
 - 시대변화에 대응하기 어렵다
 - 사람을 쓰는 비용이 크다
 - 단어의 미묘한 차이를 표현할 수 없다.



통계기반 기법

- 확률 그래프 모델
 - HMM, SVM, CRF

- TF-IDF

- Vector space model

$$\text{tf}(t, d) = \frac{\text{count}(t)}{\text{count}(d)}$$

- Zipf's Law, Topic Modeling

$$\text{idf}(t, D) = \log \frac{\text{number of documents}}{\text{number of documents containing } t}$$

$$\text{tfidf}(t, d, D) = \text{tf}(t, d) * \text{idf}(t, D)$$

- 단어의 분산 표현

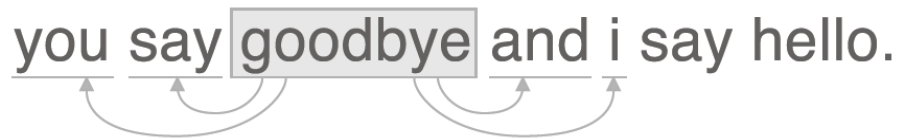
- 단어를 vector로 표현 (흡사: RGB)해서 의미파악 (dense vector)

- ANN을 이용해서 얻은 단어 vector = 단어의 밀집 vector 표현

- 분포 가설 (Distributional hypothesis)

- “단어 자체가 아닌 그 단어가 사용된 맥락 (context)이 의미를 형성”

you say goodbye and i say hello.



-
- Co-occurrence matrix
 - 단어의 빈도를 중심으로 분석

you say goodbye and i say hello.

	you	say	goodbye	and	i	hello	.
you	0	1	0	0	0	0	0
say	1	0	1	0	1	1	0
goodbye	0	1	0	1	0	0	0
and	0	0	1	0	1	0	0
i	0	1	0	1	0	0	0
hello	0	1	0	0	0	0	1
.	0	0	0	0	0	1	0

- Vector간 유사도

$$\text{similarity}(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x} \cdot \mathbf{y}}{\|\mathbf{x}\| \|\mathbf{y}\|} = \frac{x_1 y_1 + \dots + x_n y_n}{\sqrt{x_1^2 + \dots + x_n^2} \sqrt{y_1^2 + \dots + y_n^2}}$$

```
def cos_similarity(x, y, eps=1e-8):  
    nx = x / (np.sqrt(np.sum(x ** 2)) + eps)  
    ny = y / (np.sqrt(np.sum(y ** 2)) + eps)  
    return np.dot(nx, ny)
```

- 유사단어의 Ranking 표시

- Relative ranking

$$\cos \Theta = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|}$$

- 통계기반 기법 개선 – Mutual Information

- (배경) 고빈도 단어의 문제

$$\text{PMI}(x,y) = \log_2 \frac{P(x,y)}{P(x)P(y)} = \log_2 \frac{\frac{C(x,y)}{N}}{\frac{C(x)}{N} \frac{C(y)}{N}} = \log_2 \frac{C(x,y) \cdot N}{C(x)C(y)}$$

– (where C: 동시발생행렬, C(x,y); x,y의 동시발생 횟수)

$$\text{PMI}(\text{"the"}, \text{"car"}) = \log_2 \frac{10 \cdot 10000}{1000 \cdot 20} \approx 2.32$$

- PPMI

$$\text{PPMI}(x,y) = \max(0, \text{PMI}(x,y))$$

$$\text{PMI}(\text{"car"}, \text{"drive"}) = \log_2 \frac{5 \cdot 10000}{20 \cdot 10} \approx 7.97$$

- 단, PPMI의 문제점
 - ① 단어 (어휘)수의 증가 시 차원 증가
 - ② Sparse matrix
 - ③ Noise에 약하다

- 의미 (semantic) 분석

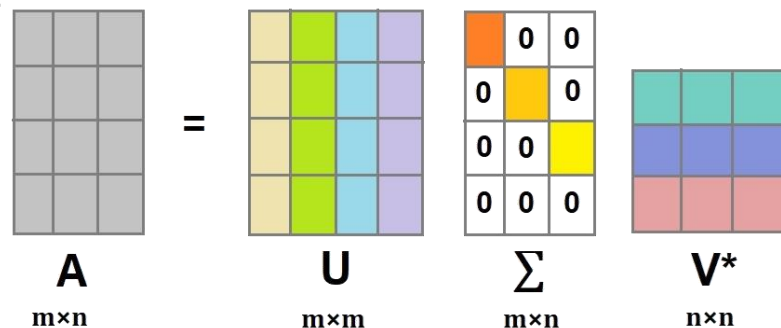
- LSA (Latent Semantic Analysis)

- reveals the meaning of word combinations and computing vectors to represent this meaning.

- PCA (Principal Components Analysis)

- 차원축소

- SVD



- LDA (Latent Dirichlet Allocation)

- assumes that each document is a mixture of some arbitrary number of topics that you select when you begin training the LDiA model.

– SVD



- U 와 V 는 직교행렬 - 열벡터는 서로 직교. (직교행렬은 어떤 공간의 축(기저)을 형성)
- U 는 '단어공간'
- S 는 대각행렬 - 대각성분은 특이값 순서. (= '해당 축'의 중요도)
- 중요도 낮은 원소를 제거할 수 있다.

전처리



To Do's

- 설치
 - Anaconda + ...
 - VS code
 - Konlpy
 - 형태소분석기
 - Okt(Open Korea Text), 메캅(Mecab), 코모란(Komoran), 한나눔(Hannanum), 꼬꼬마(Kkma)