[자연어 처리의 모든 것] 1. 자연어 처리의 시작

1. 자연어 처리 활용 분야와 트렌드

1) Natural language processing

- 딥러닝 기술의 발전을 선도하는 핵심 분야
- 주요 학회: ACL, EMNLP, NAACL
- 학문 분야

1. Low-level parsing : Tokenization, stemming
    - Tokenization: I study math라는 문장을 이해하기 위해 각 단어 단위(토큰)로 쪼개나가는 과정
    - 문장은 토큰들로 이루어진 시퀀스
    - stemming: study라는 단어의 어미가 다양하게 변해도 단어의 어근을 추출하는 것
    - 가장 low level의 task
2. Word and phrase level : NER(Named Entity Recognation), POS(Part-Of-Speech) tagging
    - NER(Named Entity Recognation): 단일 단어로 이루어진 고유 명사를 인식하는 task
    - POS(Part-Of-Speech) tagging: word들이 문장 내에서 품사나 성분인지 무엇인지 알아내는 task
3. Sentence level : 감성 분류(Sentiment Analysis), 기계 번역(Machine Translation)
    - 감성 분류(Sentiment Analysis): 주어진 문장이 긍정, 부정 어조인지 구분
    - 기계 번역(Machine Translation): 영어 문장을 한글 문장으로 번역할 때 적절한 단어, 문법으로 번역
4. Multi-sentence and paragraph level : 논리적 내포 및 모순관계 예측(Entailment Prediction), 독해기반 질의응답(question answering), 챗봇(dialog systems), 요약(summarization)
    - 논리적 내포 및 모순관계 예측(Entailment Prediction): 두 문장간의 논리적인 내포, 모순관계 예측

- 독해기반 질의응답(question answering): 질문 키워드가 포함된 문서 검색 후 독해를 통해 질문에 대한 정답을 정확하게 알아내서 제시해주는 과정
- 챗봇(dialog systems): 대화를 수행할 수 있는 기술
- 요약(summarization): 주어진 문서를 자동으로 요약하는 task

2) Text mining (텍스트 마이닝)

- 빅데이터 분석과 관련되는 경우가 많음
- 주요 학회 : KDD, The WebConf(前 WWW), WSDM, CIKM, ICWSM
- 학문 분야

1. 텍스트 및 문서 데이터에서 유용한 정보 추출
2. 문서 군집화(Document clustering): 서로 다른 키워드지만 비슷한 의미를 가지는 키워드 grouping해서 분석
   ex) 토픽 모델링
3. Highly related to computational social science : 통계적으로 사회과학적 인사이트 산출

3) Information retrieval (정보 검색)

- 검색 기술을 연구하는 분야
- 주요 학회 : SIGIR, WSDM, CIKM, Recsys
- 학문 분야

1. Highly related to computational social science
2. 정보 검색 분야, **추천 시스템**

자연어 처리 분야의 트렌드

- 자연어 처리 분야는 컴퓨터 비전 혹은 영상처리 분야와 더불어 인공지능과 딥러닝 기술이 가장 활발히 적용되며 꾸준히 발전하는 분야 중 하나
- 기존 머신러닝과 딥러닝 기술로 자연어 처리 문제를 해결하기 위해서는 주어진 텍스트 데이터를 숫자로 변환하는 워드 임베딩(Word Embedding) 과정을 거치게 됨
  - 워드 임베딩: 텍스트 데이터를 단어 단위로 분리하고 각 단어를 특정한 차원으로 이루어진 벡터로 표현하는 과정
- 텍스트 데이터는 문장을 구성하는 순서(sequence) 정보가 중요하기 때문에 이를 받아들일 수 있는 특화 모델에 대한 연구가 필요했고, 그 대표적인 예로는 'RNN(Recurrent

Neural Network)'이 있음

    - RNN 계열 모델 중 단점을 보완한 LSTM, LSTEM을 단순화하여 계산 속도를 빠르게 한 GRU 모델이 나와 사용되었음

- 2017년에는 구글에서 발표한 'Attention is all YOU need' 라는 제목의 논문이 나오면서 '셀프 어텐션(Self-Attention)' 구조를 가진 '트랜스포머(Transformer) 모델'이 각광받기 시작

    - 최근 발표된 대부분의 모델들은 대부분 트랜스포머 모델을 기반으로 하고 있음
    - 트랜스포머 모델은 주로 사용되던 기계 번역 분야를 넘어 현재는 영상처리/신약개발/시계열 예측 등에서도 다양하게 사용되고 있음

- 최근에는 자가지도 학습(self-supervised Learning)이 가능한 BERT, GPT2/GPT3 와 같은 모델의 유행하고 있음

    - 자기지도 학습: 입력 문장이 주어져 있을 때 입력 중 일부단어를 가리고 맞추게 하는 task
    - 대규모의 데이터, GPU 리소스를 필요로 함

2. 기존의 자연어 처리 기법

Bag-Of-Words (단어 가방 모형)

- 단어들의 순서는 전혀 고려하지 않고, 단어들의 출현 빈도(frequency)에만 집중하는 텍스트 데이터의 수치화 표현 방법
- vocabulary 상에서 word 별로 가방을 준비하고 특정 문장에서 나타난 word들을 가방에 넣어준 후, 각 가방에 들어간 word 수를 세서 벡터로 나타냄
- Step 1. unique한 단어들 사전(vocabulary) 형태로 저장

    - 저장된 단어들은 각각 유니크한(중복x) 카테고리 변수

- Step 2. categorical variable인 unique words를 one-hot vectors로 인코딩하여 벡터로 표현

    - 이를 통해 주어진 문장을 원-핫 벡터의 합, 즉 숫자로 표현할 수 있게 됨

Naive Bayes Classifier for Document Classification

- bag-of-words 벡터로 나타낸 문서를 정해진 카테고리/클래스로 분류하는 방법

- For a document d and a class c

$$c_{MAP} = \underset{c \in C}{\operatorname{argmax}}\ P(c|d)$$ MAP is "maximum a posteriori" = most likely class

$$= \underset{c \in C}{\operatorname{argmax}}\ \frac{P(d|c)P(c)}{P(d)}$$ Bayes Rule

$$= \underset{c \in C}{\operatorname{argmax}}\ P(d|c)P(c)$$ Dropping the denominator

- 카테고리 C개

예시

- Example
  - For a document d, which consists of sequence of words w_i 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	?

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

- 학습 데이터로 주어진 Traing 1~4 번 문장을 통해 우리는 Test data(5번 문장)을 CV, NLP 두 클래스 중에 한 곳으로 분류
- 5번 문장에 있는 각 단어들이 1~4번 문장에 몇 번 등장했는지를 조건부 확률로 계산

- **For a test document d_5 = "Classification task uses transformer"**
  - 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
$P(w_{\text{'classification'}}\|c_{CV})$	$\frac{1}{14}$	$P(w_{\text{'classification'}}\|c_{NLP})$	$\frac{1}{10}$
$P(w_{\text{'task'}}\|c_{CV})$	$\frac{1}{14}$	$P(w_{\text{'task'}}\|c_{NLP})$	$\frac{2}{10}$
$P(w_{\text{'uses'}}\|c_{CV})$	$\frac{1}{14}$	$P(w_{\text{'uses'}}\|c_{NLP})$	$\frac{1}{10}$
$P(w_{\text{'transformer'}}\|c_{CV})$	$\frac{1}{14}$	$P(w_{\text{'transformer'}}\|c_{NLP})$	$\frac{1}{10}$

- 이와 같은 파라미터 추정 방식은 최대우도법(MLE)을 기반으로 유도

3. Word Embedding - (1)Word2Vec

Word Embedding

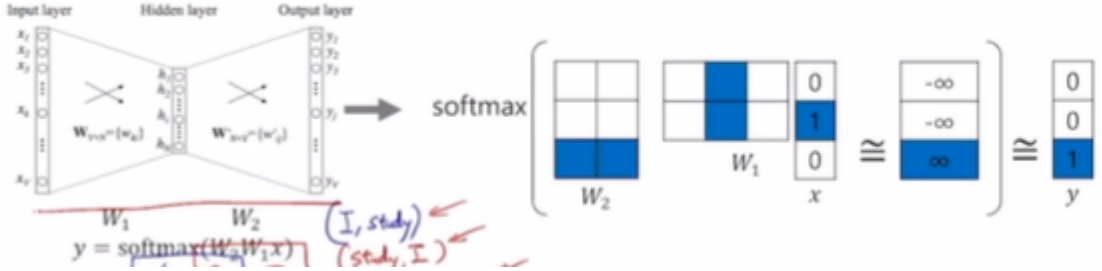
워드 임베딩: 각 단어를 좌표공간 상의 한 점, 또는 그 점의 좌표를 나타내는 벡터로 표현하는 기법
ex) (1,3,1) -> cat

- 비슷한 의미를 가지는 단어가 좌표공간 상의 비슷한 위치의 최적의 좌표값으로 mapping
  - cat-kitty는 비슷한 위치, hamburger는 먼 위치
  - 감정을 분류할 때 love-like는 비슷한 위치

Word2Vec Idea

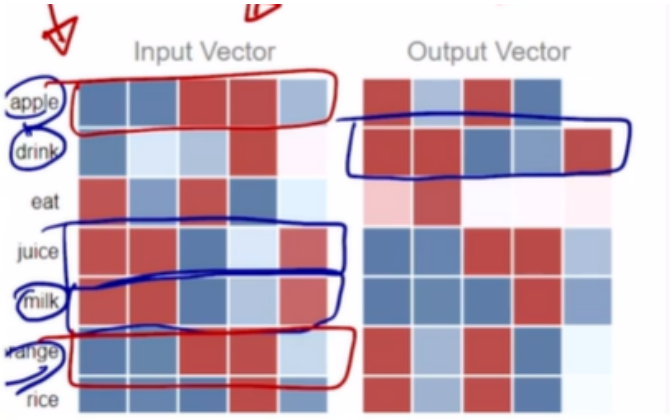
- 같은 문장 내에서 인접한 단어들간의 의미가 비슷할 것이라고 가정
- 주변에 등장하는 단어들을 통해 중심 단어의 의미가 표현될 수 있다는 것을 착안
  - 확률 분포를 예측하여 학습 진행
- 우선 워드를 Tokenization 해준 후, 유니크한 단어만 모아서 사전(Vocabulary)을 구축
  - 사전의 각 단어는 사전 사이즈만큼의 dimension을 가지는 one-hot vector
- 문장에서 중심단어를 위주로 학습 데이터를 구축
  - sliding window 기법 사용
  - “I study math”라는 문장의 중심단어가 study 라고 한다면 (I study), (study I) , (study math) 와 같은 단어쌍을 학습 데이터로 구축
- 입출력 단어쌍들에 대해 예측 task 수행하는 2 layer neural network를 만들게 됨
- 입출력 노드는 모두 3차원 one-hot vector

Word2Vec의 계산

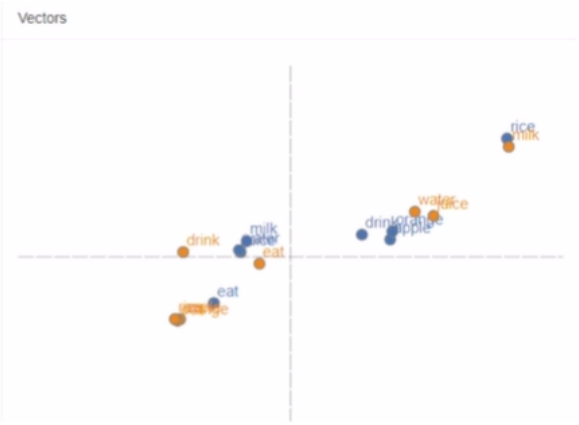


- 문장의 단어의 갯수만큼에 Input, Output 벡터 사이즈를 입력/출력
  - 연산에 사용되는 은닉 층의 차원(dim)은 사용자가 파라미터로 지정 가능
- 임베딩 레이어와의 연산은 0이 아닌 1인 부분만 추출해서 계산
  - [0,0,1] 벡터인 경우는 3번째 원소와 곱해지는 부분의 column만 추출하여 계산해줌
  - one-hot vector와 첫번째 선형 변환 (W1) matrix가 곱해지는 과정 -> 임베딩 레이어라고 함
- 마지막 결과값으로 나온 벡터는 softmax 연산을 통해 가장 큰 값이 1, 나머지는 0으로 출력
- 내적 연산을 반복되면서, 같이 등장하는 단어들 간의 벡터표현이 유사도가 커지게 함
- W1, W2에 속한 파라미터들을 조정하며 학습 진행

Word2Vec의 특성



- juice의 input vector와 drink의 output vector는 거의 유사한 형태를 가짐
  - 둘 간의 내적값은 최대한 커짐
- drink와 milk, water의 input vector도 유사한 형태를 가짐
- eat, apple, orange도 유사한 벡터 표현형을 가짐
- 입력, 출력 단어 2차원으로 차원 축소한 후 시각화한 예시



- vec[queen]-vec[king] = vec[woman]-vec[man]
  - > 남성에서 여성으로의 변화를 의미하는 벡터 관계를 효과적으로 학습
- 아이폰-휴대폰+노트북=아이패드
- Word intrusion detection: 의미가 가장 다른 단어 찾아내기
  - 단어들간의 유클리드 거리를 계산한 후 평균을 구한 후 가장 큰 단어를 고름
- 기계번역, 감정분석, 이미지 캡션 등에서 임베딩 벡터 사용

4. Word Embedding - (2)GloVe

Glove : Global Vectors for Word Representation

Glove와 Word2Vec의 차이점

- 각 입출력 단어쌍들에 대해 학습 데이터에서 두 단어가 한 window에서 몇번 동시에 등장했는지 사전에 계산을 미리하고 log 값을 취해 Ground Truth로 취급
- 단어 간의 내적값이 가까워질 수 있도록 하는 loss 함수 사용

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

- Word2Vec보다 빠르게 동작하며 더 적은 데이터에서도 잘 동작함

GloVe의 특징

- 의미 차이가 일정한 방향과 크기의 벡터로 나타남
- 형용사들에 대해 원형, 비교급/최상급 간에도 일정한 크기와 방향을 가진 벡터들이 나타남

-> 문법적 의미와 관계도 효과적으로 학습함

사전 학습된 Glove 모델

- 사전에 이미 대규모 데이터로 학습된 모델이 오픈소스로 공개되어 있음
- 위키피디아 데이터를 기반으로 하여 6B token만큼 학습 되었으며, 중복 제거 후 사전을 구축할 때도 단어의 개수가 무려 40만개(400k)에 달함
- uncased: 대문자 소문자를 구분 x <-> cased: 대소문자를 구분

해당글은 부스트코스의 [자연어 처리의 모든 것] 1. 자연어 처리의 시작 강의를 듣고 작성한 글입니다.

velog 링크