

1. Natural language processing (자연어 처리)

- 주요 학회 : ACL, EMNLP, NAACL

학문 분야

- Low-level parsing : Tokenization, stemming *token으로 쪼개기*
- Word and phrase level : NER(Named Entity Recognition), POS(Part-Of-Speech) tagging *NER: 고유명사를 인식. POS: 품사·성분 파악.*
- Sentence level : 감성 분류(Sentiment Analysis), 기계 번역(Machine Translation) *공감·부감 여부 파악. (이중부감구 등...) 문법 반영*
- Multi-sentence and paragraph level : 논리적 내포 및 모순관계 예측(Entailment Prediction), 독해기반 질의응답(question answering), 챗봇(dialog systems), 요약(summarization) *ex) 구문 잡는 문단 level*

2. Text mining (텍스트 마이닝)

빅데이터

- 주요 학회 : KDD, The WebConf(前 WWW), WSDM, CIKM, ICWSM

학문 분야

- Extract useful information and insights from text and document data *→ 다루는 거. 비슷한 거야.*
- 문서 군집화(Document clustering) *ex) 토픽 모델링*
- Highly related to computational social science : 통계적으로 사회과학적 인사이트 산출 *ex) facebook.*

3. Information retrieval (정보 검색)

기술발전 느리다(이제 상용) 구글·네이버.

- 주요 학회 : SIGIR, WSDM, CIKM, Recsys

학문 분야

- Highly related to computational social science
- 정보 검색 분야, 추천 시스템

광고, 2대, 유튜브, 영향, impact ↑↑.

자연어 처리 분야의 트렌드

1/3

- 자연어 처리 분야는 컴퓨터 비전 혹은 영상처리 분야와 더불어 인공지능과 딥러닝 기술이 가장 활발히 적용되며 꾸준비 발전하는 분야 중 하나입니다. 기존 머신러닝과 딥러닝 기술로 자연어 처리 문제를 해결하기 위해서는 주어진 텍스트 데이터를 숫자로 변환하는 '워드 임베딩(Word Embedding)' 과정을 거치게 됩니다.
- 텍스트 데이터는 문장을 구성하는 순서 정보가 중요하기 때문에 이를 받아들일 수 있는 특화 모델에 대한 연구가 필요했고, 그 대표적인 예로는 'RNN(Recurrent Neural Network)'이 있습니다. 이후 단점을 보완한 LSTM, GRU 모델이 나와 사용되었습니다.

- 2017년에는 구글에서 발표한 'Attention is all YOU need' 라는 제목의 논문이 나오면서 '셀프 어텐션(Self-Attention)' 구조를 가진 '트랜스포머(Transformer) 모델'이 각광받기 시작했습니다. 최근 발표된 대부분의 모델들은 트랜스포머 모델을 기반으로 하는 것이 많으며, 트랜스포머 모델은 주로 사용되던 '기계 번역' 분야를 넘어 현재는 영상/신약개발/시계열 예측 등에서도 다양하게 사용되고 있습니다.
- 최근에는 자가지도 학습(self-supervised Learning)이 가능한 BERT, GPT 와 같은 모델의 유행하고 있습니다.

기초의 처리 방법.

Bag-Of-Words (단어 가방 모형)

- 단어들의 순서는 전혀 고려하지 않고, 단어들의 출현 빈도(frequency)에만 집중하는 텍스트 데이터의 수치화 표현 방법입니다.
- 단어를 벡터로 표현하기 위해서는 주어진 문장에 쓰인 단어들을 사전(Vocabulary) 형태로 저장하며, 이때 주의할 점은 단어들의 중복을 허용하지 않아야 한다
- 저장된 단어들은 각각 유니크한 카테고리 변수(Categorical variable)이므로, 원-핫 인코딩(One-hot Encoding)을 이용해 벡터로 표현할 수 있습니다. 이를 통해 주어진 문장을 원-핫 벡터의 합, 즉 숫자로 표현할 수 있게 됩니다.

단어 수 만큼 dimension 1.

→ 단어별 작용.

쉽게 표현하면, Bag-Of-Words는 '가방'이라는 개념을 통해 문장을 구성하고 있는 단어들을 가방에 순차적으로 정리하는 것입니다. 가방에 있는 각 단어들은 각각 원-핫 벡터를 통해 숫자로 변환하며, 주어진 문장은 벡터의 합으로 표현됩니다.

단어가 이 예미라 상관없이 벡터 사이 관계 0

Naive Bayes Classifier for Document Classification

나이브 베이즈 분류기는 인공 신경망 알고리즘에는 속하지 않지만, 머신러닝의 주요 알고리즘으로 분류에 있어 준수한 성능을 보여주는 것으로 알려져 있습니다. 나이브 베이즈 분류기를 이해하기 위해서는 우선 베이즈의 정리(Bayes' theorem)를 이해할 필요가 있습니다. 베이즈 정리는 조건부 확률을 계산하는 방법 중 하나입니다.

$$C_{MAP} = \underset{c \in C}{\operatorname{argmax}} P(c|d) = \underset{c \in C}{\operatorname{argmax}} \frac{P(d|c)P(c)}{P(d)} = " P(d|c)P(c) \quad \text{MAP: maximum a posterior}$$

$$P(d|c)P(c) = P(w_1, \dots, w_n | c)P(c) \rightarrow P(c) \prod_{w_i \in w} P(w_i | c)$$

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

$$P(C_{CV}) = \frac{1}{2}, \quad P(C_{NLP}) = \frac{1}{2} \quad \left| \quad P(W_{task} | C_{CV}) = \frac{1}{14}, \quad P(W_{task} | C_{CV}) = \frac{2}{10}$$

학습 데이터로 주어진 Training 1~4 번 문장을 통해 우리는 Test data(5번 문장)을 CV, NLP 두 클래스 중에 한 곳으로 분류하려 합니다. 방법은 간단합니다. 5번 문장에 있는 각 단어들이 1~4번 문장에 몇 번 등장했는지를 조건부 확률로 계산하면 쉽게 알 수 있습니다. 다만 이 방식의 맹점은 다른 단어들이 분류하고자 하는 문장에 많이 등장했는지라도, Training data 에서 1번이라도 등장하지 않았다면 모든 단어들의 확률 곱으로 인해 0으로 수렴한다는 점이 있습니다.

이와 같은 파라미터 추정 방식은 최대우도법(MLE)을 기반으로 유도됨으로, 더 깊이 공부하고 싶은 분께서는 추가적으로 더 학습하시길 권해드립니다.

Word Embedding 이란?

'워드 임베딩'은 각 단어를 좌표공간에 최적의 벡터로 표현하는(임베딩하는) 기법
그렇다면 표현된 벡터값이 '최적'인지를 어떻게 알 수 있을까요? 예를 들어 알아보시다.

- kitty : 아기 고양이, cat : 고양이, hamburger : 햄버거

위 단어들을 벡터를 통해 좌표공간으로 표현한다면, 'kitty'와 'cat'은 비슷한 위치할 것입니다. 그러나 'hamburger'는 꽤 먼 거리에 표현되겠지요? 이와 같이 유사한 단어는 가까이, 유사하지 않은 단어는 멀리 위치하는 것을 '최적의 좌표값'으로 표현할 수 있습니다.

의미 유사도를 잘 표현한 벡터 표현.

또 다른 예로 감정을 분류를 한다고 했을 때에

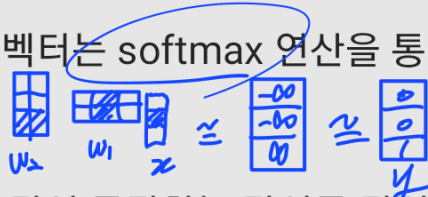
- "기쁨", "환희"는 긍정적인 감정을 나타내는 단어들과 함께,
- "분노", "증오"는 부정적인 감정을 나타내는 단어들과 비슷한 위치에 맵핑이 될 것입니다.

Word2Vec Idea

- '워드 투 벡터'의 아이디어는 "문장 내에서 비슷한 위치에 등장하는 단어는 유사한 의미를 가질 것이다" 에서 출발합니다. 즉, 주변에 등장하는 단어들을 통해 중심 단어의 의미가 표현될 수 있다는 것으로 추정을 시작합니다.
- 이를 위해 우선 워드를 토큰나이징(Tokenizing)해준 후, 유니크한 단어만 모아서 사전(Vocabulary)을 만들어주어야 합니다. 그 이후 문장에서 중심단어를 위주로 학습 데이터를 구축해줍니다.
- 예를 들어 "I study math"라는 문장의 중심단어가 study 라고 한다면, (I study), (study I), (study math) 와 같은 단어쌍을 학습 데이터로 구축합니다.

** 토큰나이징(Tokenizing)이란? : 말그대로 문자(Text)를 컴퓨터가 이해할 수 있는 Token 이라는 숫자 형태로 바꿔주는 행위

Word2Vec의 계산

- 문장의 단어의 갯수만큼 Input, Output 벡터 사이즈를 입력/출력해줍니다. 이 때 연산에 사용되는 히든 레이어(hidden layer, 은닉 층)의 차원(dim)은 사용자가 파라미터로 지정할 수 있습니다.
- 실제로 Tensorflow나 Pytorch와 같은 프레임워크에서는 임베딩 레이어와의 연산은 0이 아닌 1인 부분, 예를 들어 $[0,0,1]$ 의 벡터인 경우는 3번째 원소와 곱해지는 부분의 컬럼(column)만 뽑아서 계산해줍니다.
- 마지막 결과값으로 나온 벡터는 softmax 연산을 통해 가장 큰 값이 1, 나머지는 0으로 출력됩니다.

- 위의 연산이 반복되면서, 같이 등장하는 단어들 간의 벡터표현이 유사해지는 것을 아래 사이트에서 확인하실 수 있습니다.

Word2Vec의 특성

- 워드투벡터를 통해 단어를 임베딩하면 queen - king 그리고 woman - man , 마지막으로 aunt - uncle 의 벡터가 비슷한 것을 볼 수 있습니다. 해당 결과가 의미하는 것은 여성과 남성의 관계성을 잘 학습했다는 것을 의미합니다.
- 연세대 축제 이름인 "아카라카"에서 연세대를 빼고 고려대를 더해주면, 고려대의 축제인 "입실렌티"가 나오는 것도 확인 할 수 있습니다.

** 워드투벡터 성능 확인하기 : <http://w.elnn.kr/search>

Application of Word2Vec

- Word2Vec은 그 자체로도 의미가 있지만, 뿐만 아니라 다양한 테스트에서 사용되고 있습니다.
- Machine translation : 단어 유사도를 학습하여 번역 성능을 더 높여줍니다.
- Sentiment analysis : 감정분석, 긍부정분류를 돕습니다.
- Image Captioning : 이미지의 특성을 추출해 문장으로 표현하는 테스트를 돕습니다.

Glove : Global Vectors for Word Representation

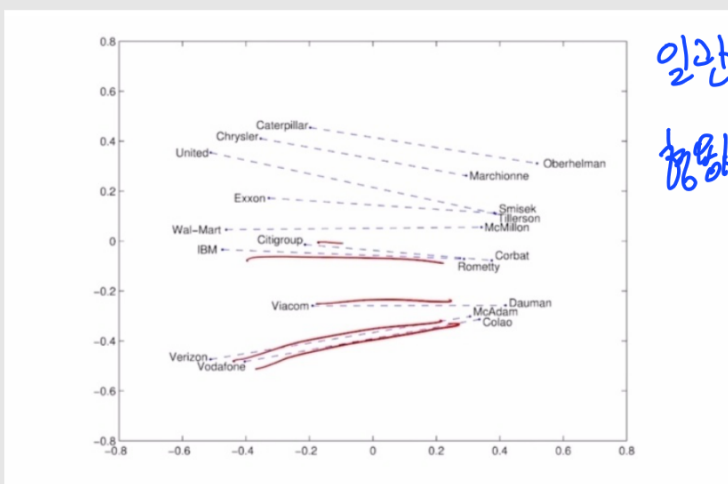
- Glove는 Word2Vec과 다르게 사전에 미리 각 단어들의 동시 등장 빈도수를 계산하며,
단어간의 내적값과 사전에 계산된 값의 차이를 줄여가는 형태로 학습합니다.
- Word2Vec는 모든 연산을 반복하지만, Glove는 사전에 계산된 Ground Truth를 사용해 반복계산을 줄일 수 있습니다.
- 따라서 Word2Vec보다 더 빠르게 동작하며, 더 적은 데이터에서도 잘 동작합니다.

사전 학습된 Glove 모델

- 사전에 이미 대규모 데이터로 학습된 모델이 오픈소스로 공개되어 있습니다. 해당 모델은 위키피디아 데이터를 기반으로 하여 6B token만큼 학습 되었으며, 중복 제거 시에도 단어의 개수가 무려 40만개(400k)에 달합니다.
- 학습된 모델을 나타낼 때 뒤에 붙는 "uncased"는 대문자 소문자를 구분하지 않는다는 의미이며, 반대로 "cased"는 대소문자를 구분한다는 의미입니다. 예를 들어 Cat과 cat이 uncased에서는 같은 토큰으로 취급되지만, cased에서는 다른 토큰으로 취급됩니다.
- Glove 깃헙 주소 : <https://github.com/stanfordnlp/GloVe>

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

↗ 가짜지도록!



일관된 관계 학습.

형태 - 비교 - 학습 → 일정한 크기 방향 벡터.

