|  |  |
| --- | --- |
| 교육 제목 | AI 프로젝트 기반 S/W 전문가 양성과정 |
| 교육 일시 | 22.01.21 |
| 교육 장소 | 학원 |
| **교육 내용** | |
| 오전 | 1. Introduction to Word Embedding Word : Discrete, not Continuous 어휘는 계층적 의미 구조를 지니고 있으며, 이에 따라 단어 사이의 유사성을 지님 🡪 ex) 파랑과 핑크 중에서 빨강에 가까운 단어는 무엇인가? OHE으로 표현된 값은 유사도나 모호성을 표현할 수 없다. Dense Vector로 표현하는 것이 유리(합이 1이어야 하는 것이 아님) 2. Feature Vectors Feature(특징) 🡪 샘플을 잘 설명하는 특징, 특징을 통해 우리는 특정 샘플을 수치화 할 수 있다. Feature Vector 🡪 각 특징들을 모아서 하나의 vector로 만드는 것 컴퓨터는 나이, 머리, 얼굴, 신장 등 특징을 수치화해서 Feature를 만듦 3. Representation Learning via Dimension Reduction 신경망은 x와 y사이의 관계를 학습하는 과정에서 자연스럽게 x의 feature를 추출하는 방법을 학습함 레이어 중간의 hidden representation은 y의 값을 구하기 위해 x에서 필요한 정보를 더 작은 차원에 압축 표현(Auto Encoder)한 것이라 할 수 있음 4. Word Embedding 딥러닝의 시대에 들어와 신경망의 이러한 특성을 활용하여 단어를 연속적인 값으로 표현하고자 하는 시도가 이어짐 이전에 비해 훌륭한 Dense vector를 얻을 수 있게 되어, 단어의 필요한 특징을 잘 표현할 수 있게 되었음 🡪 유사도 등의 연산에 유리함 5. Wordnet 🡪 전통적 NLP에서 단어들을 분류 계층적 구조를 지님(대충 반시계 방향으로 90도 누운 트리구조.jpg) 단어와 단어 사이의 거리를 구함 similarity(w, w’) = -log distance(w, w’) 6. Summary WordNet - 단어의 계층적 구조를 파악할 수 있음, 동의어 집합(synset)을 구할 수 있음, 단어 사이의 유사도를 계산할 수 있음 🡪 코퍼스 없이도 가능! BUT - 특정 도메인(또는 수집 데이터)에 특화된 수치를 계산하고 싶을 때 신조어나 사전에 등록되지 않은 단어를 처리할 때 Data-driven방식이 필요 7. Data-driven Methods Thesaurus기반 방식은 사전에 대해 의존도가 높으므로, 활용도가 떨어질 수 있음 데이터에 기반한 방식은(데이터가 충분하다면) task에 특화된 활용이 가능 8. 단어의 표현의 카테고리화 9. Data-driven Methods 2 Local Representation - 해당 단어 그 자체만 보고, 특정값을 맵핑하여 단어를 표현하는 방법 - = discrete representation - ex) puppy : 1, cute(귀여운) : 2, lovely(사랑스러운) : 3 🡪 숫자 맵핑 Distributed Representation - 그 단어를 표현하고자 주변을 참고하여 단어를 표현하는 방법 - = continuous representation - ex) puppy(강아지)라는 단어 근처에 주로 cute(귀여운), lovely(사랑스러운)이라는 단어가 자주 등장하므로, puppy는 cute, lovely한 느낌이다 로 단어 정의! 10. WordFeature Vectors: Traditional Methods BOW(Bag of Words) 🡪 단어들의 순서는 전혀 고려하지 않고 출현 빈도에만 집중하는 텍스트 데이터의 수치화 표현 방법 만드는 과정 🡪 1. 각 단어에 고유한 정수 인덱스를 부여, 2. 각 인덱스의 위치에 단어 토큰의 등장 횟수를 기록한 벡터를 만듦 11. 문서 단어 행렬(Document Term Matrix, DTM) 다수의 문서에서 등장하는 각 단어들의 빈도를 행렬로 표현한 것 각 문서에 대한 BOW를 하나의 행렬로 만드는 것 BOW표현을 다수의 문서에 대해서 행렬로 표현하고 부르는 용어 각 문서에서 등장한 단어의 빈도를 행렬의 값으로 표기 문서 단어 행렬은 문서들을 서로 비교할 수 있도록 수치화할 수 있다는 점에 의의 12. DTM의 한계 13. 희소표현(Sparse representation)  * 각 문서 벡터의 차원은 OHV와 마찬가지로 전체 단어 집합의 크기를 가짐, 많은 문서벡터가 대부분의 값이 0을 가질 수도 있음 🡪 희소벡터, 희소 행렬, 희소벡터는 많은 양의 저장공간과 계산을 위한 리소스가 필요  1. 단순 빈도 수 기반 접근  * 예를 들어 영어에 대한 DTM을 만들었을 때, 불용어인 The는 어떤 문서든 자주 등장하는데 다른 문서에도 The가 많이 들어가는데 빈도수가 비슷하다고 해서 유사한 문서라고 판단하면 안됨  1. TF-IDF 텍스트 마이닝(Text Mining)에서 중요하게 사용 어떤 단어 w가 문서 d내에서 얼마나 중요한지 나타내는 수치 2. TF(Term Frequency) - 단어의 문서 내에 출현한 횟수, 숫자가 클수록 문서 내에서 중요한 단어(라고 가정), 하지만 the와 같은 단어도 TF값이 매우 클 것 3. IDF(Inverse Document Frequency) - 그 단어가 출현한 문서의 숫자의 역수(inverse) 값이 클수록 the와 같이 일반적으로 많이 쓰이는 단어 TF-IDF(w, d) = TF(w, d)/DF(w) 🡪 w = 단어, d = documents TF-IDF 값이 높으면 해당 단어가 문서에서 중요한 내용을 차지한다 4. DTM(Document-Term-Matrix) 실습 5. TF-IDF를 feature로 사용할 수 없을까?? TF-IDF는 문서에 해당 단어가 얼마나 중요한지 수치화 중요한 문서가 비슷한 단어들은 비슷한 의미를 지닐까?(가정) 각 문서에서의 중요도를 feature로 삼아서 vector를 만든다면? 6. TF-IDF Matrix 🡪 단어의 각 문서(문장, 주제)별 TF-IDF 수치를 vector화 row : 단어, column : 주제 7. Based on Context Window (Co-occurrence) 함께 나타나는 단어들을 활용 - 가정: 의미가 비슷한 단어라면 쓰임새가 비슷할 것 쓰임새가 비슷하기 때문에, 비슷한 문장 안에서 비슷한 역할로 사용될 것 따라서 함께 나타나는 단어들이 유사할 것 - Context Window를 사용하여 windowing을 실행 window의 크기라는 hyper-parameter 추가 적절한 window 크기를 정하는 것이 중요 8. summary thesaurus 기반 방식에 비해 코퍼스(or 도메인) 특화된 표현 가능 여전히 sparse한 vector로 표현 됨 PCA를 통해 차원 축소를 하는 것도 한 방법 9. reuters dataset을 가지고 실습 10. from sklearn.naive\_bayes import MultinomialNB 추론 대상의 사전 확률과 추가적인 정보를 기반으로 해당 대상의 사후 확률을 추론하는 통계적 방법을 베이즈 추정(Bayesian Estimation)이라고 합니다. 베이즈 정리에 기반한 통계적 분류 기법입니다. 가장 단순한 지도 학습 (supervised learning) 중 하나입니다. 나이브 베이즈 분류기는 빠르고, 정확하며, 믿을만한 알고리즘입니다. 정확성도 높고 대용량 데이터에 대해 속도도 빠릅니다. |
| 오후 | 1. 오전 수업 이어서 2. 실제 뉴스 기사 크롤링 및 분류 !pip install beautifulsoup4 !pip install newspaper3k !pip install konlpy 3. Mecab 설치 !pip install konlpy !git clone <https://github.com/SOMJANG/Mecab-ko-for-Google-Colab.git> %cd Mecab-ko-for-Google-Colab/ !bash install\_mecab-ko\_on\_colab190912.sh %cd ../ 4. BeautifulSoup을 이용한 HTML 읽어오기 5. Tag로 분류된 HTML contents 읽기 soup = BeautifulSoup(html, "html.parser") print(soup.select("body")) # body 내용을 싹 긁어옴 6. newspaper 모듈 🡪 사용자가 지정한 url에서 text를 추출해주는 모듈 from newspaper import Article # Article 인스턴스 만들기, url 변수에 주소 할당하고 lang은 ko ㅇㅇ article = Article(url, language = "ko") article.title # 제목 불러오기 article.contents # 내용 불러오기 7. newspaper 실습 – 크롤러 만들기 8. 전처리하고 분류모델 만들기 |