|  |  |
| --- | --- |
| 교육 제목 | Word Embedding |
| 교육 일시 | 12월 2일(목) |
| 교육 장소 | 강의실 |
| **교육 내용** | |
| 전체 내용 | * Word Embedding * 자연어를 컴퓨터가 이해하고 효율적으로 처리하게 하기 위해서는 컴퓨터가 이해할 수 있도록 자연어를 적절히 변환할 필요가 있다. * 단어를 표현하는 방법에 따라서 자연어 처리의 성능이 크게 달라지기 때문에 이에 대한 많은 연구가 있었고 여러가지 방법들이 있다 * 희소표현(Sparse Representation) * 원-핫 인코딩을 통해서 나온 원-핫 벡터들은 표현하고자 하는 단어의 인덱스의 값만 1이고, 나머지 인덱스에는 전부 0으로 표현되는 표현 방법 * 문제점   단어의 개수가 늘어나면 벡터의 차원이 한없이 커진다는 점   * 밀집표현(Dense Representation) * 밀집 표현은 벡터의 차원을 단어 집합의 크기로 상정하지 않음   사용자가 설정한 값으로 모든 단어의 벡터 표현의 차원을 맞춤  )이 과정에서 더 이상 0과 1만 가진 값이 아니라 실수값을 가지게 된다.   * 워드 임베딩   단어를 밀집벡터의 형태로 표현하는 방법을 워드 임베딩  이 밀집벡터를 워드 임베딩 과정을 통해 나온 결과라고 하여 임베딩 벡터  워드 임베딩 방법으로는 LSA, Word2Vec, FastText, Glove 등이 있다.   * Word2Vec   주변(context window)에 같은 단어가 나타나는 단어일수록 비슷한 벡터 값을 가져야 한다.  Context window의 사이즈에 따라 embedding의 성격이 바뀔 수 있다.   * Skip-gram   주변 단어를 예측하도록 하는 과정에서 적절한 단어의 임베딩(정보의 압축)을 할 수 있다.   * Glove   단어 x와 윈도우 내에 함께 출현한 단어들의 출현 빈도를 맞추도록 훈련  출현 빈도가 적은 단어에 대해서는 loss의 기여도를 낮춤 (따라서 출현 빈도가 적은 단어에 대해 부정확해지는 단점을 보완)   * Conclusion * 딱히 어떤 알고리즘이 더 뛰어나다고는 할 수 없다. (구현이 쉽고 빠른 오픈소스를 사용하는 것이 낫다.) * 두 개의 다른 알고리즘 결과물을 concat 하여 사용하기도 함 |