**@Pytorch**

Sequence를 다룰 수 있는 RNN 모듈을 많이 씀

자연어처리의 가장 핵심은 첫 번째 레이어에서 시작되는 임베딩

임베딩이 잘 되는 것이 매우 중요한 부분

자연어처리는 긴 역사를 가지고 있음

* 구글의 기계번역기 (트랜스포머 이후로 발전)  
  제 2차 세계대전 때부터 많은 투자를 받음  
  자연어처리의 역사는 곧 기계번역의 역사  
  처음에는 문법 기반 -> 통계 기반 -> 인공신경망 기반 -> 트랜스포머 기반
* 시리
* 챗봇…
* 요약기

컴퓨터는 우리의 말을 이해하지 못하기 때문에 이해할 수 있도록 만들어야 함  
 -> 컴퓨터가 이해할 수 있는 representation으로 전달  
 인간의 언어가 어떻게 만들어져 있는지 이해하여야 함

Speech(음향적 전달) -> 음소 분석 -> 통사 분석 -> 의미 분석 -> 문맥 추론

많은 경우 각각의 요소의 합은 전체와 다를 수 있음  
ex) 여기 왜 이렇게 더워요? -> 에어컨을 켜달라는 의미를 내포할 수 있음

#컴퓨터가 언어를 이해할 수 있는가?

품사분석기, 구문분석기 등은 많이 개발되었지만, 컴퓨터가 이해를 할 수 있는가?

우리가 원하는 그대로 이해하는 것은 어려움

하지만 다른 접근으로 진행 중

인간의 언어는 **중의성**이 매우 큼 (ambiguity)

#파이프라인 만들기

* 하나 하나 작은 것을 수행하여 전체를 연결하여 수행
* 1. 문장 단위로 나누기 (sentence segmentation)  
   -> 늘 마침표 기준으로 나눠지지 않음 (소수점, 약어)
* 2. 단어 단위로 나누기 (word tokenization)  
   -> 늘 띄어쓰기 기준으로 나눠지지 않음 (New York)
* 3. 품사 예측 (predicting POS)  
   -> 형태소 분석기를 사용해서 품사를 지정함
* 4. Lemmatization (원형 복원)  
   -> 형태는 변했지만, 같은 의미를 같은 단어를 원형으로 복원함  
   한국어는 원형복원 모듈이 없음
* 5. 불용어 지정 (identifying stop words)
* 6. 문장 구조의 의존도 분석 (dependency parsing)  
   -> 영어엔 Stanford parser, 구글 파서 등이 있음  
   한국어엔 없음  
   영어의 경우 어순이 정해져 있고, 형태소가 매우 복잡함 (먹으셨겠으다만은)  
   하지만 한국어는 어순에 있어 비교적 자유로움
* 6-2. 명사구 찾기 (finding noun phrase)
* 7. 개체명 인식 (Named Entity Recognition)
* 8. 대용어 해소 (Coreference Resolution)  
   -> 영어에서는 대명사가 활발하게 쓰이기 때문에 매우 중요함  
   한국어는 상대적으로 중요하지 않음 (그, 그녀 등을 쓰지 않음)
* 모두를 작은 것으로 나눠서 이후 파이프라인으로 통합함

#spaCy

#textacy

트위터에서는 정형화된 글을 쓰지 않음

#분류 모델을 사용하여 의미 추출하기

* 놀이공원 리뷰?  
  자연어처리의 대부분의 문제는 분류 문제  
  리뷰 텍스트를 넣고, 분류 모델을 만듦 -> 텍스트를 넣으면 별로 평점이 나옴
* 이게 어떻게 작동하는 지?  
  분류모델이 작동을 잘 하는 이유가 많은 리뷰가 계속 생성되고, 텍스트에 숨어있는 feature를 잘 찾아내서 작동함
* 의미 파악을 매우 빠르게 함  
  파이프라인에 따라 진행하려면 시간이 오래 걸리지만, 분류모델은 선형회귀나 로지스틱회귀로 계산하기 때문에 매우 빠르게 답을 얻을 수 있음
* 큰 학습 데이터가 필요함  
  작은 데이터 셋으로는 특징을 학습하기 힘듦  
  데이터 셋이 클수록 정확도가 높아짐
* 리뷰 분석기, 스팸 분류기 등…
* **의미 파악, 기계 번역 등 모든 것은 분류 문제라고 볼 수 있음**
* 단어의 의미를 분석할 수 있는 임베딩도

TPU의 경우 병렬계산을 더 잘 수행함

구글에서 소량 빌려줌