Insight campus SeSAC

임베딩 (Embedding)

실무형 인공지능 자연어처리

3 FastText

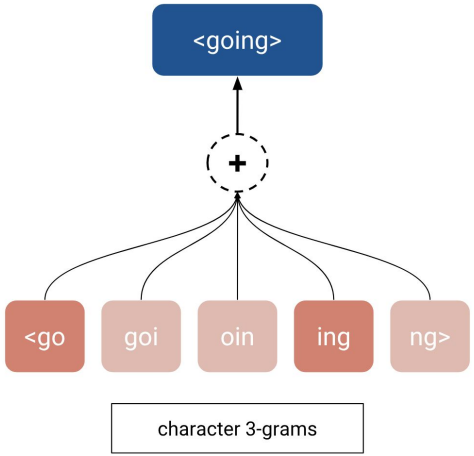
FastText

- Facebook에서 발표한 Word Embedding 기법
- Word2vec나 GloVe의 경우 언어의 형태학적(Morphological)인 특성을 반영하지 못하고, 또 희소한 단어에 대해서는 Embedding이 되지 않음
- FastText에서는 단어를 **Bag-of-Characters**로 보고, 개별 단어가 아닌 n-gram의 characters를 Embedding함 (Skip-gram model 사용)
    - 각 단어는 Embedding된 n-gram의 합으로 표현됨, 그 결과 빠르고 좋은 성능을 보임
- Word2Vec와 FastText의 가장 큰 차이점은 Word2Vec은 단어를 쪼개질 수 없는 단위로 생각한다면, FastText는 하나의 단어 안에도 여러 단어들이 존재하는 것으로 간주(= **Subword**)
- 내부 단어(subword)를 고려하여 학습
    - 내부 단어(subword)를 통해 모르는 단어(OOV)에 대해서도 다른 단어와의 유사도를 계산할 수 있음

OOV (Out Of Vocabulary)

- FastText에서 각 단어를 글자의 n-gram으로 나타냄
- 예를 들어, tri-gram의 경우, apple은 app, ppl, ple로 분리하고 임베딩
- FastText에서 birthplace(출생지)란 단어를 학습하지 않은 상태라고 해보자.
  - 다른 단어 n-gram으로서 birth와 place를 학습한 적이 있다면 birthplace의 임베딩 벡터(Embedding Vector)를 만들어낼 수 있음

〈ap, app, ppl, ple, le〉 # n = 3 이므로 길이가 3
〈apple〉 # 특별 토큰



Rare Word

- Word2Vec 경우 단어 등장 빈도가 높을 수록 정확하게 임베딩 되지만, 희소 단어(rare word) 경우 임베딩 정확도가 높지 않음
- FastText는 희소 단어(rare word)의 경우 문자로 n-gram을 하는 특성학 학습 경우의 수가 많아지므로 Word2Vec와 비교하여 정확도가 높은 경향 (=FastText가 노이즈가 많은 코퍼스에 강점)
- Word2Vec에서는 오타가 섞인 단어는 임베딩이 되지 않음(OOV). FastText는 이 경우도 일정 수준 성능을 보임

한국어 FastText

- 글자 단위 - 글자 단위로 임베딩하는 경우
예를 들어서 글자(Character) 단위의 임베딩의 경우에 n=3일때 '자연어처리'라는 단어에 대해 n-gram을 만들어보면 다음과 같다.

〈자연, 자연어, 연어처, 어처리, 처리〉

- 자모 단위 - 자모 단위(초성, 중성, 종성 단위)로 임베딩하는 경우.
예를 들어 '자연어처리'라는 단어에 대해서 초성, 중성, 종성을 분리하고, 만약, 종성이 존재하지 않는다면 '_'라는 토큰을 사용한다고 가정한다면 '자연어처리'라는 단어는 아래와 같이 분리가 가능.

〈ㅈㅏ, ㅈㅏ _, ㅏ _ ㅇ, ... 중략〉

4 단어 임베딩 비교

Word Embedding 방법론

임베딩 비교

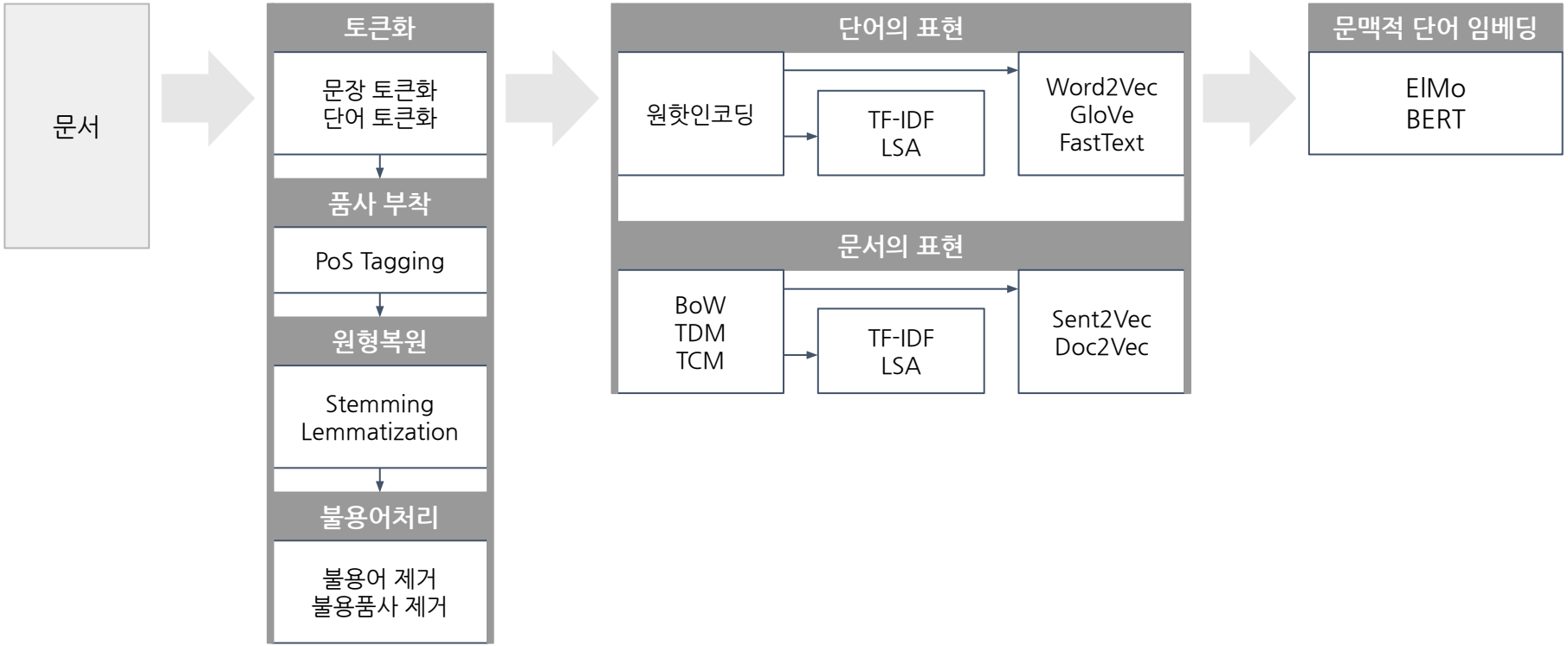
- Word2Vec
  - Word2Vec는 사용자가 설정한 window내에 등장한 단어의 벡터 요소를 업데이트 하며 단어를 임베딩
  - window 내 등장하지 않는 단어는 멀어지도록 (내적값이 줄어들도록 = 유사도가 작아지도록),
    동시 등장하는 단어는 가까워지도록(내적값이 커지도록 = 유사도가 커지도록) 학습
- Glove
  - GloVe는 2014년 미국 스탠포드대학 연구팀에서 개발한 단어 임베딩 방법론
  - 단어 동시 등장 여부를 사용 (Term-Context matrix, Co-occurrence matrix)
  - GloVe로 임베딩된 단어 벡터끼리의 내적은 동시 등장확률의 로그값과 같습니다.
    (their dot product equals the logarithm of the words' probability of co-occurrence)
- FastText
  - 페이스북이 2016년 발표
  - 단어를 내부단어(subword)의 벡터들로 표현한다는 점을 제외하고는 Word2Vec와 유사
  - 노이즈가 많은 말뭉치에 강함.

세 방법론의 한계

- 의미상 아무런 관련이 없어 보이는 단어간에도 벡터공간에 가깝게 임베딩(=코사인유사도가 큰) 결과를 보일 수 있음 (예. 소프트웨어, 하드웨어)
- 임베딩시 단어간 동시 등장 정보를 사용한다는 면에서는 Word2Vec, GloVe, FastText 모두 count based 방법과 본질적으로 유사
- 하지만 동시 등장 정보를 벡터로 표현했을때 의미를 보존하고 있다는 면에 있어서, 기존 count based 방법인 TF-IDF, LSA 보다 월등히 개선되었기 때문에 각광을 받음

Insight campus SeSAC

임베딩 절차



처리 의존도를 고려하여 성능을 높이는 작업을 진행

감사합니다.

Insight campus SeSAC

daniel@fins.ai