FastText

박소영

01. FASTTEXT

Word2Vec의 단점

1. Out of Vocabulary
2. Infrequent words

01. FASTTEXT

기존의 제안

WPM (Word Piece Model)
Appear라는 단어를 app + ear로부터 유추
- App, ear로부터 appear을 만들어 내기가 어렵다

02. FASTTEXT

FastText란?

Bag of character n-grams

Ex) character 3-grams
- Should
  - <should>
  - <sh, sho, hou, oul, uld, ld>
  - 6개의 3글자 짜리 subwords
- 실제 단어에서는 3-6grams + <,>를 더한 단어 추가

Ex)
- Should
  - <sh, sho, hou, oul, uld, ld>
  - <sho, shou, houl, ould, uld>
  - <shou, shoul, hould, ould>
  - <shoul, should, hould>
  - <should>

02. FASTTEXT

FastText란?

Should의 vector는 subword vectors의 평균으로 표현

V(should) = (V(<sh) + V(sho) + V(hou) + … + V(hould>) + V(<should>))/n

이를 이용해 비슷한 단어 vector를 얻을 수 있다

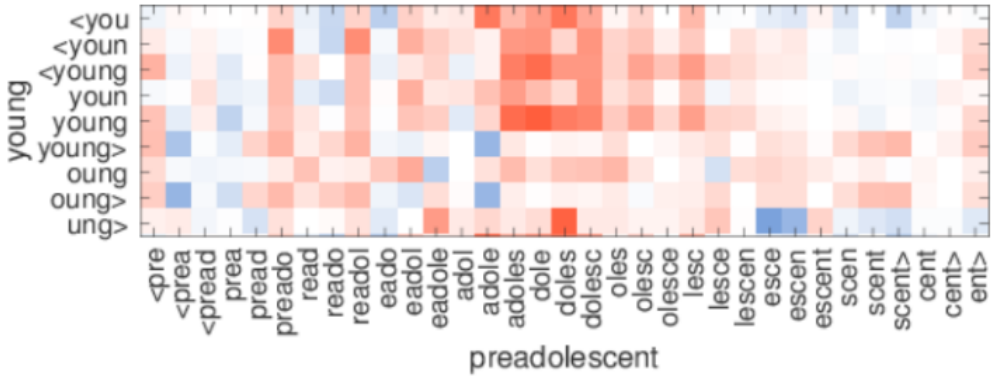
Should, shoulder는 대부분 subwords가 공통으로 존재하기 때문

Trainnig 땐 subwords의 vector를 학습

02. FASTTEXT

FastText란?

빨간 색일 수록 두 subword vectors 간의 cosine similarity가 크다
Young, adole, adoles, doles는 높은 similarity를 갖는다



02. FASTTEXT

FastText란?

'잘가' -> 'ㅈㅏㄹㄱㅏ-'
FastText는 띄어쓰기를 기준으로 학습

Word2Vec처럼 CBOW, Skip-gram으로 학습 가능

03. FASTTEXT

FastText with Korean

- 과거시제 '-았', '-었 ' , '-율','-률 ' 을 학습하기 위해선 Stride 1이 필요
- 비문이 많은 txt에서는 -를 추가하지 않는 경우가 존재
- 이처럼 어휘를 자소까지 분리해 학습하는 것은 일부 어휘의 semantic 특성 학습에 도움이 되지 않음
  - semantic 유사도 파악에서는 word2vec보다 성능이 떨어짐
- fastText는 문법적 요소가 n-gram을 통해 분석되므로 문법적 유사도 파악에는 좋다

04. FASTTEXT

FastText의 구성

.bin
- parameter
- 모든 n-gram에 대한 vector 포함
- 바이너리 형태

.vec
- 라인에 있는 한 단어의 단어 벡터를 포함하고 있는 text file
- 일반 텍스트로 집계된 단어 벡터

04. FASTTEXT

FastText의 구성

Word2int

- Word string을 hash로 indexing
- MAX_VOCAB_size를 미리 설정해서 size 넘으면 효율 감소

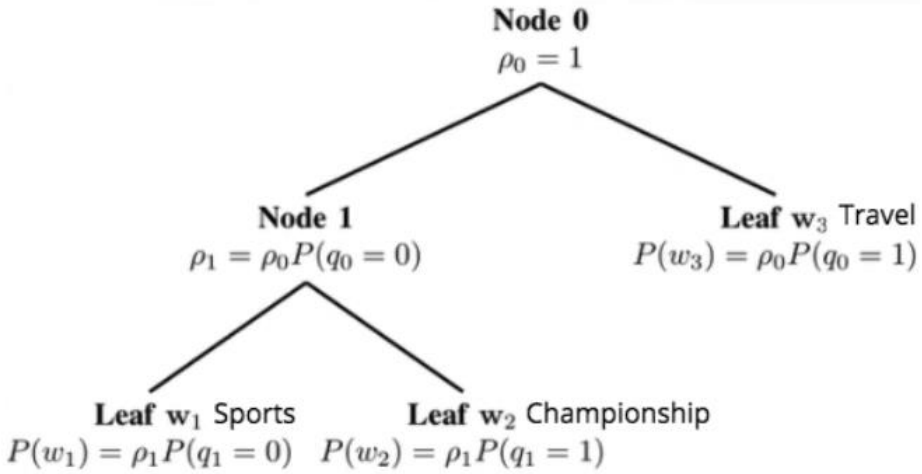
Words_

- 문장을 단어로 바꿔준다
- 각 단어에 대한 struct entry를 함께 저장
  - Word : 단어를 나타내는 string
  - Count : 그 단어의 total count
  - Entry_type : {word, label}
  - Subwords : subword에 대한 vector

```
struct entry {
  std::string word;
  int64_t count;
  entry_type type;
  std::vector<int32_t> subwords;
};
```

05. FASTTEXT

FastText를 이용한 classification




```
(tensorflow) soyoung ~/fastText master ./fasttext supervised -input ../CRC_soyoung/data/label_fastText/train.txt -output fastText_twitter_up
Read 1M words
Number of words:  71627
Number of labels: 2
Progress: 100.0% words/sec/thread:  241716 lr:  0.000000 loss:  0.207601 ETA:   0h 0m
(tensorflow) soyoung ~/fastText master ./fasttext test fastText_twitter_up.bin ../CRC_soyoung/data/label_fastText/valid.txt
N       21629
P@1     0.664
R@1     0.664
(tensorflow) soyoung ~/fastText master ./fasttext test fastText_twitter_up.bin ../CRC_soyoung/data/label_fastText/test.txt
N       21644
P@1     0.668
R@1     0.668
```