Sgspeech_v2에 쓰인 subword algorithm

서강대학교 청각지능 연구실
박호성

서강대학교
SOGANG UNIVERSITY

소개

Sentencepiece Model

- Google에서 개발한 unsupervised text tokenizer
- 자연어 text문장들이 주어졌을 때, 알고리즘을 통해 자동으로 token을 생성해 주는 프로그램
  - Byte-pair encoding (BPE), unigram language model 등 통계 기반 알고리즘을 지원

특징

- 데이터를 기반으로 한 통계 기반 알고리즘을 사용
- 언어에 비종속적
- Python, Linux bash를 지원 (모델 호환 가능)

소개

◆ 음성인식에서의 sentencepiece

- 음성인식의 출력 단위를 결정함
- DNN-WFST 구조의 음성인식의 경우,
  디코딩 네트워크에 사용되는 단어 사전의 인식 단위를 정의하며, 10만~100만 사이의 단어 개수를 가짐
- End-to-end 구조의 음성인식의 경우
  모델의 출력 단위를 정의하며, 1,000~2,000개 사이의 단어 개수를 가짐

소개

Sentencepiece 학습

- 학습 자료의 transcription을 사용해 sentencepiece 학습을 진행

```
물 론  우 리 가  이 기 리 라  믿 습 니 다
이 후  베 이 징 으 로  자 리 를  옮 겨  남 북  특 사 간  만 남 은  계 속 됐 습 니 다
장 자 지 에 를  보 고  조 국  중 국 을  더 욱  사 랑 하 게  됐 습 니 다
전 국  기 능 경 기 대 회  냉 동 기 술 부 문  대 회 장 에 서 는  어 젯 밤 에  심 한  논 쟁 이  벌 어 졌 습 니 다
서 울 시 는  올  팔 월 에  구 체 적 인  최 종 안 을  마 련 할  방 침 입 니 다
정 부 는  오 일  남 북 정 상 회 담  대 표 단  백 삼 십 명 의  명 단 을  확 정 해  발 표 했 습 니 다
이 로 써  홈 런  일 위 는  송 지 만  이 승 엽  우 즈  세 명 이 고 요
이 어  남 측  대 표 단 을  위 해  준 비 한  학 생 들 의  특 별 공 연 이  펼 쳐 졌 습 니 다
칠 십 이 로  꺾 고  십 승  고 지 에  오 르 며  단 독  선 두 를  지 켰 습 니 다
외 국  관 광 객 들 에 게  한 국 의  초 복 은  낯 설 지  않 습 니 다
```

- Unigram language model을 이용하여 4,000개의 subword를 생성함
  - Conformer 논문상으로는 영어 단어 1,000개를 사용했다고 되어 있으나, 한국어 특성 상 음절의 개수가 많아 4,000개를 사용함

```
1986  spm_train --input=kor_txt --model_prefix=etri_kor --vocab_size=4000 --character_coverage=1.0
```

소개

Sentencepiece tokenizing

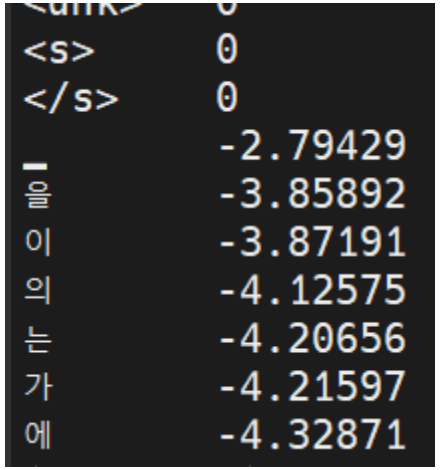
- 학습된 모델을 기반으로 분절을 실행

```
(sgspeech) hosung@blizzard:~$ head kor_txt | spm_encode --model etri_kor.model
▁물 론 ▁우 리 가 ▁이 기 리 라 ▁ 믿 습 니 다
▁이 후 ▁베 이 징 으 로 ▁자 리 를 ▁옮 겨 ▁남 북 ▁특 사 간 ▁만 남 은 ▁계 속 됐 습 니 다
▁장 자 지 에 를 ▁보 고 ▁조 국 ▁중 국 을 ▁더 욱 ▁사 랑 하 게 ▁됐 습 니 다
▁전 국 ▁기 능 경 기 대 회 ▁냉 동 기 술 부 문 ▁대 회 장 에 서 는 ▁어 젯 밤 에 ▁심 한 ▁논 쟁 이 ▁벌 어 졌 습 니 다
▁서 울 시 는 ▁올 ▁팔 월 에 ▁구 체 적 인 ▁최 종 안 을 ▁마 련 할 ▁방 침 입 니 다
▁정 부 는 ▁오 일 ▁남 북 정 상 회 담 ▁대 표 단 ▁백 삼 십 명 의 ▁명 단 을 ▁확 정 해 ▁발 표 했 습 니 다
▁이 로 써 ▁홈 런 ▁일 위 는 ▁송 지 만 ▁이 승 엽 ▁우 즈 ▁세 명 이 고 요
▁이 어 ▁남 측 ▁대 표 단 을 ▁위 해 ▁준 비 한 ▁학 생 들 의 ▁특 별 공 연 이 ▁펼 쳐 졌 습 니 다
▁칠 십 이 로 ▁꺾 고 ▁십 승 ▁고 지 에 ▁오 르 며 ▁단 독 ▁선 두 를 ▁지 켰 습 니 다
▁외 국 ▁관 광 객 들 에 게 ▁한 국 의 ▁초 복 은 ▁ 낯 설 지 ▁않 습 니 다
```

소개

◆ Sentencepiece tokenizing

- 단어 사전 예시



- _(underscore와 같은 형식이지만 unicode가 다름)
  - 본문의 띄어쓰기를 대체하는 방식
  - 각 토큰이 ' '(space)로 구분되기 때문에, 원문의 space를 표시하기 위해 사용함

알고리즘

Byte-pair encoding

- 정보 압축 알고리즘으로 제안되었음 [Gage, 1994](https://www.derczynski.com/papers/archive/BPE_Gage.pdf)
- 자연어 처리에 도입되어 OOV를 발생시키지 않는 방법으로 사용됨 [Sennrich, 2016](https://arxiv.org/pdf/1508.07909.pdf)

  * 정보 압축 -> 데이터 손실 없이 두 개의 symbol을 하나로 치환 -> character로부터 subword 생성

- 방법

  1) 문장을 분해할 수 있는 가장 작은 단위(Character)로 단어 사전을 구성함

  2) 가장 높은 frequency를 보이는 인접한 character의 쌍을 단어 사전에 추가

  2-1) 가장 높은 frequency를 가지는 쌍이 여러 개 존재할 때 먼저 등장한 것을 우선으로 함

  2-2) 새로 조합된 subword에 frequency 가 가장 높은 subword가 포함되는 경우 새로 조합된 것을 우선시 함

  3) 사용자가 원하는 단어 사이즈가 나올 때 까지 1)~2)를 반복함

알고리즘

◆ Byte-pair encoding - toy problem

- 목표: 주어진 corpus를 이용해서 20개의 subword를 가진 단어 사전을 구축하시오.

CORPUS
long: 3
longer: 2
shortest: 5
establish: 3

VOCABULARY (#12)
l, o, n, g, e, r, s, h, t, a, b, i

STEP 1.
lo: 5
on: 5
ng: 5
ge: 2
er: 2
sh: 8
ho: 5
or: 5
rt: 5
te: 5
es: 8
st: 8
ta: 3
ab: 3
bl: 3
li: 3
is: 3

알고리즘

Byte-pair encoding - toy problem

- 목표: 주어진 corpus를 이용해서 20개의 subword를 가진 단어 사전을 구축하시오.

CORPUS
long: 3
longer: 2
shortest: 5
establish: 3

VOCABULARY (#13)
l, o, n, g, e, r, s, h, t, a, b, l,
sh

STEP 2.
lo: 5
on: 5
ng: 5
ge: 2
er: 2
~~sh: 8~~
ho: 5
or: 5
rt: 5
te: 5
es: 8
st: 8
ta: 3
ab: 3
bl: 3
li: 3
is: 3

sh o: 5
i sh: 3

알고리즘

◆ Byte-pair encoding - toy problem

- 목표: 주어진 corpus를 이용해서 20개의 subword를 가진 단어 사전을 구축하시오.

CORPUS
long: 3
longer: 2
shortest: 5
establish: 3

VOCABULARY (#14)
l, o, n, g, e, r, s, h, t, a, b, i,
sh, es

STEP 3.

lo: 5	sh o: 5
on: 5	i sh: 3
ng: 5	**es t: 8**
ge: 2	t es: 5
er: 2	
~~sh: 8~~	
ho: 5	
or: 5	
rt: 5	
te: 5	
~~es: 8~~	
st: 8	
ta: 3	
ab: 3	
bl: 3	
li: 3	
is: 3	

알고리즘

Byte-pair encoding - toy problem

- 목표: 주어진 corpus를 이용해서 20개의 subword를 가진 단어 사전을 구축하시오.

CORPUS
long: 3
longer: 2
shortest: 5
establish: 3

VOCABULARY (#15)
l, o, n, g, e, r, s, h, t, a, b, i,
sh, es, est

STEP 4.

lo: 5	sh o: 5
on: 5	i sh: 3
ng: 5	~~es t: 8~~
ge: 2	t es: 5
er: 2	t est: 5
~~sh: 8~~	est a: 3
ho: 5	
or: 5	
rt: 5	
te: 5	
~~es: 8~~	
~~st: 8~~	
ta: 3	
ab: 3	
bl: 3	
li: 3	
is: 3	

알고리즘

◆ Byte-pair encoding - toy problem

- 목표: 주어진 corpus를 이용해서 20개의 subword를 가진 단어 사전을 구축하시오.

CORPUS
long: 3
longer: 2
shortest: 5
establish: 3

VOCABULARY (#16)
l, o, n, g, e, r, s, h, t, a, b, i,
sh, es, est, lo

STEP 5.

~~lo: 5~~
on: 5
ng: 5
ge: 2
er: 2
~~sh: 8~~
ho: 5
or: 5
rt: 5
te: 5
~~es: 8~~
~~st: 8~~
ta: 3
ab: 3
bl: 3
li: 3
is: 3
sh o: 5
i sh: 3
~~es t: 8~~
t es: 5
t est: 5
est a: 3
lo n: 5

알고리즘

◆ Byte-pair encoding - toy problem

- 목표: 주어진 corpus를 이용해서 20개의 subword를 가진 단어 사전을 구축하시오.

CORPUS
long: 3
longer: 2
shortest: 5
establish: 3

VOCABULARY (#17)
l, o, n, g, e, r, s, h, t, a, b, i,
sh, es, est, lo, lon

STEP 6.

~~lo: 5~~
~~on: 5~~
ng: 5
ge: 2
er: 2
~~sh: 8~~
ho: 5
or: 5
rt: 5
te: 5
~~es: 8~~
~~st: 8~~
ta: 3
ab: 3
bl: 3
li: 3
is: 3
sh o: 5
i sh: 3
~~es t: 8~~
t es: 5
t est: 5
est a: 3
~~lo n: 5~~
lon g: 5

알고리즘

◆ Byte-pair encoding - toy problem

- 목표: 주어진 corpus를 이용해서 20개의 subword를 가진 단어 사전을 구축하시오.

CORPUS
long: 3
longer: 2
shortest: 5
establish: 3

VOCABULARY (#18)
l, o, n, g, e, r, s, h, t, a, b, i,
sh, es, est, lo, lon, long

STEP 7.

~~lo: 5~~
~~on: 5~~
~~ng: 5~~
ge: 2
er: 2
~~sh: 8~~
ho: 5
or: 5
rt: 5
te: 5
~~es: 8~~
~~st: 8~~
ta: 3
ab: 3
bl: 3
li: 3
is: 3

sh o: 5
i sh: 3
~~es t: 8~~
t es: 5
t est: 5
est a: 3
~~lo n: 5~~
~~lon g: 5~~
long e: 2

알고리즘

◆ Byte-pair encoding - toy problem

- 목표: 주어진 corpus를 이용해서 20개의 subword를 가진 단어 사전을 구축하시오.

CORPUS
long: 3
longer: 2
shortest: 5
establish: 3

VOCABULARY (#19)
l, o, n, g, e, r, s, h, t, a, b, i,
sh, es, est, lo, lon, long, ho

STEP 8.

~~lo: 5~~
~~on: 5~~
~~ng: 5~~
ge: 2
er: 2
~~sh: 8~~
~~ho: 5~~
or: 5
rt: 5
te: 5
~~es: 8~~
~~st: 8~~
ta: 3
ab: 3
bl: 3
li: 3
is: 3

sh o: 5
i sh: 3
~~es t: 8~~
t es: 5
t est: 5
est a: 3
~~lo n: 5~~
~~lon g: 5~~
long e: 2
s ho: 5
ho r : 5

알고리즘

Byte-pair encoding - toy problem

- 목표: 주어진 corpus를 이용해서 20개의 subword를 가진 단어 사전을 구축하시오.

CORPUS
long: 3
longer: 2
shortest: 5
establish: 3

VOCABULARY (#20)
l, o, n, g, e, r, s, h, t, a, b, i,
sh, es, est, lo, lon, long, ho,
hor

STEP 9.

~~lo: 5~~	sh o: 5
~~on: 5~~	i sh: 3
~~ng: 5~~	~~es t: 8~~
ge: 2	t es: 5
er: 2	t est: 5
~~sh: 8~~	est a: 3
~~ho: 5~~	~~lo n: 5~~
~~or: 5~~	~~lon g: 5~~
rt: 5	long e: 2
te: 5	s ho: 5
~~es: 8~~	~~ho r : 5~~
~~st: 8~~	S hor: 5
ta: 3	Hor t: 5
ab: 3	
bl: 3	
li: 3	
is: 3	

알고리즘

Byte-pair encoding의 문제점

- 가장 작은 단위에서 조합하는 방식이기 때문에, 하나의 문장에 대해 여러 개의 조합법이 나올 수 있음
  - 예를 들어, ‘hello’의 경우에도 BPE를 사용하면 ‘h ello’, ‘he llo’, ‘he l l o’ 등 여러 가지가 나올 수 있음

- 같은 단어가 여러 pieces로 나누어지는 경우,
  학습에 있어서 동일한 의미를 가진 문자가 다른 segment로 나누어지는 ambiguity발생

알고리즘

Unigram language model

- 기존 BPE 기반 알고리즘에 subword regularization을 적용하여 기존 BPE의 문제인 multiple segmentation을 극복함
- 단어 사전을 생성해 내는 것이 아닌, heuristic한 단어 사전을 최적화하여 지정한 단어 개수가 될 때까지 반복함
- Unigram language model의 occurrence probabilities을 기반으로 상위 n%의 단어들만 남김 (character 제외)

- 가정

  - 단어 사전 구축 시 모든 subword의 확률은 독립적이며, subword sequence의 확률은 subword 확률의 곱으로 나타낼 수 있음

- 방법

  1) 학습 corpus로부터 heuristic한 단어 사전을 만듬

  2) 각 단어에 대해 unigram 확률을 계산

  3) 각 단어에 대해 entropy를 계산

  4) entropy 순서대로 모든 단어들을 정렬한 다음, 상위 n%의 단어만 남김 (n은 일반적으로 0.8)

  5) 사용자가 원하는 단어 사이즈가 나올 때 까지 2)~4)를 반복함

알고리즘

◆ Unigram language model – toy problem

CORPUS
long: 3
longer: 2
shortest: 5
establish: 3

VOCABULARY (#20)
l, o, n, g, e, r, s, h, t, a, b, i, sh, es, est, lo, lon, long, ho, hor

1) Vocabulary 중 'sh'를 선택

2) training corpus 예시: 'shortest'

3) Training corpus에 대해 'sh'에 대한 likelihood를 구함
3-1) 'sh'를 포함한 분절 실시: 'sh_o_r_t_est'
3-2) log(p(sh)*p(o)*p(r)*p(t)*p(est)) = -5.96465

4) Training corpus에 대해 'sh'가 없는 경우에 대한 likelihood를 구함
4-1) log(p(s)*p(h)*p(o)*p(r)*p(t)*p(est)) = -7.16536

5) 'shortest' corpus에 대한 'sh'의 likelihood = -5.96465 – (-7.16536) = 1.20074

6) 해당 작업을 모든 training corpus와 characte를 제외한 단어에 적용함

7) Likelihood 순서로 정렬한 뒤, 상위 80%를 취함

8) 원하는 단어 사전의 개수가 될 때까지 반복

symbol	count	probability
l	8	0.062992
o	10	0.07874
n	5	0.03937
g	5	0.03937
e	10	0.07874
r	7	0.055118
s	8	0.062992
h	8	0.062992
t	8	0.062992
a	3	0.023622
b	3	0.023622
i	3	0.023622
sh	8	0.062992
es	8	0.062992
est	8	0.062992
lo	5	0.03937
lon	5	0.03937
long	5	0.03937
ho	5	0.03937
hor	5	0.03937

알고리즘

Unigram language model - toy problem

- 목표: 주어진 corpus와 단어 사전을 이용해서 17개의 subword를 가진 단어 사전이 되도록 정규화하세요

CORPUS
long: 3
longer: 2
shortest: 5
establish: 3

VOCABULARY (#18)
l, o, n, g, e, r, s, h, t, a, b, i,
sh, ~~es~~, ~~est~~, lo, lon, long, ho,
hor

symbol	count	probabilitie	entropy
a	3	0.02702703	0.042384
b	3	0.02702703	0.042384
i	3	0.02702703	0.042384
n	5	0.04504505	0.060647
g	5	0.04504505	0.060647
lo	5	0.04504505	0.060647
lon	5	0.04504505	0.060647
long	5	0.04504505	0.060647
ho	5	0.04504505	0.060647
hor	5	0.04504505	0.060647
r	7	0.06306306	0.07569
l	8	0.07207207	0.082323
s	8	0.07207207	0.082323
h	8	0.07207207	0.082323
t	8	0.07207207	0.082323
sh	8	0.07207207	0.082323
o	10	0.09009009	0.094173
e	10	0.09009009	0.094173

VOCABULARY (#17)
l, o, n, g, e, r, s, h, t, a, b, i,
~~sh~~, ~~es~~, ~~est~~, lo, lon, long, ho,
hor

검증

Experimental setup [Park, 2020]

- Dataset

  - KsponSpeech corpus[Bang, 2020]

    * Large-scale spontaneous speech corpus of Korean.
      In this experiment, we'll just transcription of this corpus for language modeling.

    * It contains 622,545 utterances.

    * Training set: 621,545 utterances.

    * Test set: 1,000 utterances.

- Language modeling

  - n-gram language modeling[Li, 2018]

    * It calculates the occurrence of n consecutive words.

    * Implemented by SRILM toolkit [Stolcke, 2002].

검증

Experimental result [Park, 2020]

- Metric

  - perplexity (ppl)

    * Average branching factor needed to represent information on test corpus

Table 1. Performance of BPE and unigram language model-based subword tokenization

Subword tokenization	Perplexity(ppl)
Byte pair encoding (BPE)	977.73
Unigram	860.72

- Result

  - Unigram language model-based subword tokenization shows better performance than BPE-based.

결론

- Subword 생성 알고리즘 두 가지를 설명하고, 이를 n-gram에 적용하여 성능 확인함
- 원래 e2e에 하려고 했는데 아직 학습이 덜 끝…났어요
- 끝나는대로 공유함

Reference

[Sennrich, 2016] Sennrich, R., Haddow, B., Birch, A., (2016). Neural Machine Translation of Rare Words with Subword Units, *Proceedings of the 54th annual Meeting of the Association for Computational Linguistices*, 1, 1715-1725

[Gage, 1994] Gage, P., (1994). A new algorithm for data compression, *C Users Journal*, 12, 2, 23-38

[Kudo, 2018] Kudo, T., (2018). Subword Regularization: Improving Neural Network Translation Models with Multiple Subword Candidates, *arXiv preprint arXiv:1804.10959*

[Bang, 2020] Bang, J., Yun, S., Kim. S., Choi, M., Lee, M., Kim. Y., Kim. D., Park, J., Lee, Y., and Kim, S., (2020), KsponSpeech: Korean Spontaneous Speech Corpus for Automatic Speech Recognition, *Applied Sciences*, 10, 19, 6936

[Li, 2018] Li, S., Xu, J., (2018). A Recurrent Neural Network Language Model Based on Word Embedding, *Proceedings of APWeb-WAIM*, 368-377

[Stolcke, 2002] Stolcke, A., (2002). SRILM-an extensible language modeling toolkit, *Proceedings of 7th International Conference on Spoken Language Processing*, 901-904