Subword Regularization

1. Introduction

- 번역 모델에서 vocabulary size는 성능에 큰 영향을 줌
- 제한된 vocabulary 에서 좋은 성능을 내기 위해 subword unit을 사용
- BPE는 가장 일반적인 subword segmentation 알고리즘

1. Introduction

- BPE의 문제점

  : 같은 어휘로도 여러개의 subword sequence를 생성할 수 있는것을 고려하지 않음

Subwords (_ means spaces)	Vocabulary id sequence
_Hell/o/_world	13586 137 255
_H/ello/_world	320 7363 255
_He/llo/_world	579 10115 255
_/He/l/l/o/_world	7 18085 356 356 137 255
_H/el/l/o/_/world	320 585 356 137 7 12295

Table 1: Multiple subword sequences encoding the same sentence "Hello World"

1. Introduction

- noise나 segmentation 에러에 강한 subword segmentation 방법을 제안
  - 제안 1 : 여러개의 분할 후보를 통합
  - 제안 2 : language model 기반의 새로운 분할 알고리즘

2. NMT with multiple subword segmentation

- 일반적인 Sequential NMT 모델

$$P(\mathbf{y}|\mathbf{x};\theta) = \prod_{n=1}^{N} P(y_n|\mathbf{x}, y_{<n};\theta),$$

$$= \mathrm{P}(y_1|x) * \mathrm{P}(y_2|x,\ y_1) * \mathrm{P}(y_3|x,\ y_1, y_2) * \ldots$$

x : source sentence, y : target sentence

2. NMT with multiple subword segmentation

- 일반적인 Sequential NMT 모델의 학습 방법

$$\{\langle X^{(s)}, Y^{(s)} \rangle\}_{s=1}^{|D|} = \{\langle \mathbf{x}^{(s)}, \mathbf{y}^{(s)} \rangle\}_{s=1}^{|D|},$$

$$\theta_{MLE} = \arg\max_{\theta} \mathcal{L}(\theta)$$

$$where,\ \mathcal{L}(\theta) = \sum_{s=1}^{|D|} \log P(\mathbf{y}^{(s)} | \mathbf{x}^{(s)}; \theta). \quad (2)$$

2. NMT with multiple subword segmentation

- 논문의 가정

: source sentence X와 target sentence Y는 여러개의 subword sequence로 분리할 수 있으며, 이 sequence는 각각 P(x|X), P(y|Y) 의 확률값을 따른다.

$$\mathcal{L}_{marginal}(\theta) = \sum_{s=1}^{|D|} \mathbb{E}_{\substack{\mathbf{x}\sim P(\mathbf{x}|X^{(s)}) \\ \mathbf{y}\sim P(\mathbf{y}|Y^{(s)})}} [\log P(\mathbf{y}|\mathbf{x};\theta)] \quad (3)$$

앞에서는 subword sequence가 한 가지 였지만,
가정에 따라 논문에서는 여러개의 subword sequence를 고려하므로 학습시 marginal likelihood를 사용함?

2. NMT with multiple subword segmentation

- 문장이 길어지면 가능한 candidate가 크게 늘어 marginal likelihood를 최적화 하기 어렵기 때문에 sequence sample 수를 k로 제한

$$\mathcal{L}_{marginal}(\theta) \cong \frac{1}{k^2} \sum_{s=1}^{|D|} \sum_{i=1}^{k} \sum_{j=1}^{k} \log P(\mathbf{y}_j|\mathbf{x}_i;\theta) \quad (4)$$

$$\mathbf{x}_i \sim P(\mathbf{x}|X^{(s)}), \quad \mathbf{y}_j \sim P(\mathbf{y}|Y^{(s)}).$$

2. NMT with multiple subword segmentation

- Decoding
  - **one best decoding**

    P(x|X) 가 최대인 x 를 x* 이라고 할 때, 이 x*를 이용해 decoding

  - **n-best decoding**

    n개의 segmentation 후보들로 decoding한 결과를 y라고 할 때, 아래 score가 최대인 결과 y*를 사용

$$score(\mathbf{x}, \mathbf{y}) = \log P(\mathbf{y}|\mathbf{x})/|\mathbf{y}|^{\lambda}, \qquad (5)$$

3. Subword segmentation with language model

1. BPE (생략)

2. Unigram language model

: 여러개의 subword segmentation과 각각의 확률값을 output으로 가짐

$$P(\mathbf{x}) = \prod_{i=1}^{M} p(x_i), \qquad (6)$$

$$\forall i \; x_i \in \mathcal{V}, \; \sum_{x \in \mathcal{V}} p(x) = 1,$$

V : pre-determined vocabulary, p(xi) : subword occurrence probability

3. Subword segmentation with language model

- S(X)가 segmentation 후보 집합일 때, 최적의 x*는 다음과 같이 구할 수 있음

$$\mathbf{x}^* = \arg\max_{\mathbf{x} \in \mathcal{S}(X)} P(\mathbf{x}), \qquad (7)$$

Viterbi 알고리즘을 이용해 x*를 구함 (Viterbi ; 최적 상태를 얻기 위한 DP방법)

3. Subword segmentation with language model

- 어휘 V가 주어졌을 때 subword xi의 등장 확률 p(xi)는 EM알고리즘을 이용해 구함

$$\mathcal{L} = \sum_{s=1}^{|D|} \log(P(X^{(s)})) = \sum_{s=1}^{|D|} \log\Big(\sum_{\mathbf{x} \in S(X^{(s)})} P(\mathbf{x}) \Big)$$

배치 내 s번째 문장 X(s) 의 확률 = 배치 내 s번째 문장 X(s)에 대한 segmentation 후보 확률들의 합?

3. Subword segmentation with language model

- 실제로는 어휘셋 V와 등장확률을 둘 다 최적화 해야하는데, 둘 다 하기는 힘듦
- 아래의 방법을 사용
  1. train corpus 내에서 충분히 큰 어휘 셋을 생성
  2. V가 적절한 크기가 될 때 까지 아래를 반복
     a. 어휘 셋을 고정
     b. subword의 등장확률을 EM 알고리즘으로 구함
     c. 각 subword를 뺀 후 구한 likelihood의 변화분 loss i 계산
     d. loss를 감소시킨 순으로 정렬 후 상위 몇% 까지 사용할지 결정
        i. 이 때 single character는 반드시 포함해야 함

V를 구하는 여러가지 방법이 있지만 주로 모든 character의 조합과 자주 등장하는 set을 사용?

3. Subword segmentation with language model

3. Subword sampling

P(x|X) 의 분포를 따르는 segmentation 후보 들 중 l개의 상위 후보를 구함

$$P(\mathbf{x}_i|X) \cong P(\mathbf{x}_i)^\alpha / \sum_{i=1}^{l} P(\mathbf{x}_i)^\alpha,$$

i번째 segmentation 후보는 위와 같은 확률분포를 따라 sampling됨

여기서 **α**는 분포의 smoothness를 나타내는데,

α가 클수록 Viterbi segmentation을 선택할 경향이 크며, α가 작을수록

uniform한 분포에서 선택될 확률이 큼

3. Subword segmentation with language model

- $l \to \infty$이면 이론적으로 모든 가능한 segmentation을 고려할 수 있으나

  문장이 길어질수록 l의 크기는 exponential하게 증가하기 때문에 실행가능하지 않음

- $l \to \infty$ (모든 가능한 segmentation 고려)를 적용하기 위해 FFPS 알고리즘을 사용

3. Subword segmentation with language model

- 결론

  : BPE 보다 Subword segmentation이 좋음

  1. 확률기반 모델을 사용 ; 유연함
  2. 여러 segmentation 후보와 그에 대한 확률을 알 수 있음

4. Related Work

- 이전의 연구들과는 달리
  - source sentence 뿐 아니라 **target sentence**에도 적용 가능
  - parameter를 업데이트할 때 **마다** sampling을 통해 segmentation을 생성

5. Experiments

Corpus	Language pair	Size of sentences			Parameters		
		train	dev	test	#vocab (Enc/Dec shared)	#dim of LSTM embedding	#layers of LSTM (Enc+Dec)
IWSLT15	en ↔ vi	133k	1553	1268	16k	512	2+2
	en ↔ zh	209k	887	1261	16k	512	2+2
IWSLT17	en ↔ fr	232k	890	1210	16k	512	2+2
	en ↔ ar	231k	888	1205	16k	512	2+2
KFTT	en ↔ ja	440k	1166	1160	8k	512	6+6
ASPEC	en ↔ ja	2M	1790	1812	16k	512	6+6
WMT14	en ↔ de	4.5M	3000	3003	32k	1024	8+8
	en ↔ cs	15M	3000	3003	32k	1024	8+8

Table 2: Details of evaluation data set

5. Experiments

- 사용한 번역 모델 : GNMT (https://norman3.github.io/papers/docs/google_neural_machine_translation.html)
- 평가 방법 : BLEU (https://brunch.co.kr/@kakao-it/154)
  - 번역 모델에서 많이 사용하는 평가 지표
  - 정답과 번역결과가 n-gram단위로 얼마나 비슷한지 보는 것
- baseline : BPE

5. Experiments

- decoding 방법 두 가지
  - one-best decoding
  - n-best decoding (n=64)
- l : 상위 몇 개의 segmentation 후보를 고려할 것인지
  - $l = 1$: 하나의 후보만 고려 (BPE 와 비교했을 때 빈도 vs language model 중 뭐가 좋은지 평가 가능)
  - $l = 64$
  - $l = \infty$
- α : sampling 할 분포의 smoothing을 고려
  - 작을수록 uniform한 분포에서 sampling 되는 것
  - $\alpha = 0.1, 0.2, 0.5$

5. Experiments

n-best decoding의 성능이 전체적으로 좀 더 좋음

Corpus	Language pair	baseline (BPE)	Proposed (one-best decoding)			Proposed (n-best decoding, $n=64$)		
			$l=1$	$l=64$ $\alpha=0.1$	$l=\infty$ $\alpha=0.2/0.5$	$l=1$	$l=64$ $\alpha=0.1$	$l=\infty$ $\alpha=0.2/0.5$
IWSLT15	en → vi	25.61	25.49	27.68*	27.71*	25.33	28.18*	28.48*
	vi → en	22.48	22.32	24.73*	26.15*	22.04	24.66*	26.31*
	en → zh	16.70	16.90	19.36*	20.33*	16.73	20.14*	21.30*
	zh → en	15.76	15.88	17.79*	16.95*	16.23	17.75*	17.29*
IWSLT17	en → fr	35.53	35.39	36.70*	36.36*	35.16	37.60*	37.01*
	fr → en	33.81	33.74	35.57*	35.54*	33.69	36.07*	36.06*
	en → ar	13.01	13.04	14.92*	15.55*	12.29	14.90*	15.36*
	ar → en	25.98	27.09*	28.47*	29.22*	27.08*	29.05*	29.29*
KFTT	en → ja	27.85	28.92*	30.37*	30.01*	28.55*	31.46*	31.43*
	ja → en	21.37	21.46	22.33*	22.04*	21.37	22.47*	22.64*
ASPEC	en → ja	40.62	40.66	41.24*	41.23*	40.86	41.55*	41.87*
	ja → en	26.51	26.76	27.08*	27.14*	27.49*	27.75*	27.89*
WMT14	en → de	24.53	24.50	25.04*	24.74	22.73	25.00*	24.57
	de → en	28.01	28.65*	28.83*	29.39*	28.24	29.13*	29.97*
	en → cs	25.25	25.54	25.41	25.26	24.88	25.49	25.38
	cs → en	28.78	28.84	29.64*	29.41*	25.77	29.23*	29.15*

Table 3: Main Results (BLEU(%)) (l: sampling size in SR, α: smoothing parameter). * indicates statistically significant difference ($p < 0.05$) from baselines with bootstrap resampling (Koehn, 2004). The same mark is used in Table 4 and 6.

https://donghwa-kim.github.io/BLEU.html

5. Experiments

subword regularization을 한 결과가 더 좋음

Corpus	Language pair	baseline (BPE)	Proposed (one-best decoding)			Proposed (n-best decoding, $n=64$)		
			$l=1$	$l=64$ $\alpha=0.1$	$l=\infty$ $\alpha=0.2/0.5$	$l=1$	$l=64$ $\alpha=0.1$	$l=\infty$ $\alpha=0.2/0.5$
IWSLT15	en → vi	25.61	25.49	27.68*	27.71*	25.33	28.18*	28.48*
	vi → en	22.48	22.32	24.73*	26.15*	22.04	24.66*	26.31*
	en → zh	16.70	16.90	19.36*	20.33*	16.73	20.14*	21.30*
	zh → en	15.76	15.88	17.79*	16.95*	16.23	17.75*	17.29*
IWSLT17	en → fr	35.53	35.39	36.70*	36.36*	35.16	37.60*	37.01*
	fr → en	33.81	33.74	35.57*	35.54*	33.69	36.07*	36.06*
	en → ar	13.01	13.04	14.92*	15.55*	12.29	14.90*	15.36*
	ar → en	25.98	27.09*	28.47*	29.22*	27.08*	29.05*	29.29*
KFTT	en → ja	27.85	28.92*	30.37*	30.01*	28.55*	31.46*	31.43*
	ja → en	21.37	21.46	22.33*	22.04*	21.37	22.47*	22.64*
ASPEC	en → ja	40.62	40.66	41.24*	41.23*	40.86	41.55*	41.87*
	ja → en	26.51	26.76	27.08*	27.14*	27.49*	27.75*	27.89*
WMT14	en → de	24.53	24.50	25.04*	24.74	22.73	25.00*	24.57
	de → en	28.01	28.65*	28.83*	29.39*	28.24	29.13*	29.97*
	en → cs	25.25	25.54	25.41	25.26	24.88	25.49	25.38
	cs → en	28.78	28.84	29.64*	29.41*	25.77	29.23*	29.15*

Table 3: Main Results (BLEU(%)) (l: sampling size in SR, α: smoothing parameter). * indicates statistically significant difference ($p < 0.05$) from baselines with bootstrap resampling (Koehn, 2004). The same mark is used in Table 4 and 6.

특히 비교적 작은 데이터 셋에서 더 좋음

5. Experiments

Domain (size)	Corpus	Language pair	Baseline (BPE)	Proposed (SR)
Web (5k)	IWSLT15	en → vi	13.86	17.36*
		vi → en	7.83	11.69*
		en → zh	9.71	13.85*
		zh → en	5.93	8.13*
	IWSLT17	en → fr	16.09	20.04*
		fr → en	14.77	19.99*
	WMT14	en → de	22.71	26.02*
		de → en	26.42	29.63*
		en → cs	19.53	21.41*
		cs → en	25.94	27.86*
Patent (2k)	WMT14	en → de	15.63	25.76*
		de → en	22.74	32.66*
		en → cs	16.70	19.38*
		cs → en	23.20	25.30*
Query (2k)	IWSLT15	en → zh	9.30	12.47*
		zh → en	14.94	19.99*
	IWSLT17	en → fr	10.79	10.99
		fr → en	19.01	23.96*
	WMT14	en → de	25.93	29.82*
		de → en	26.24	30.90*

Table 4: Results with out-of-domain corpus ($l = \infty$, $\alpha = 0.2$: IWSLT15/17, $l = 64$, $\alpha = 0.1$: others, one-best decding)

결론 : open domain에도 유용함

http://www.cs.toronto.edu/~pekhimenko/tbd/datasets.html

5. Experiments

Model	BLEU
Word	23.12
Character (512 nodes)	22.62
Mixed Word/Character	24.17
BPE	24.53
Unigram w/o SR ($l = 1$)	24.50
Unigram w/ SR ($l = 64,\ \alpha = 0.1$)	25.04

Table 5: Comparison of different segmentation algorithms (WMT14 en→de)

5. Experiments

일반적으로 l이 클수록 좋지만,
α에 민감하므로 충분한 크기의 데이터 셋이라면 l=64 써도 될듯

고려해야 할 후보는 많은데,
uniform한 분포에서 sampling

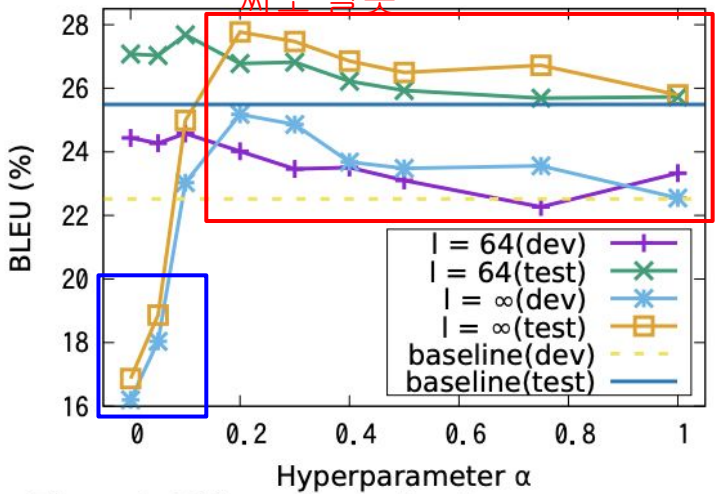

Figure 1: Effect of sampling hyperparameters

5. Experiments

Regularization type	en→vi	vi→en	en→ar	ar→en
No reg. (baseline)	25.49	22.32	13.04	27.09
Source only	26.00	23.09*	13.46	28.16*
Target only	26.10	23.62*	14.34*	27.89*
Source and target	27.68*	24.73*	14.92*	28.47*

Table 6: Comparison on different regularization strategies (IWSLT15/17, $l = 64$, $\alpha = 0.1$)

source 와 target 문장 모두에 적용한 결과가 가장 좋지만, 한쪽에만 적용한 것도 안한 결과보다는 더 좋으니 인코더/디코더 를 사용하는 여러 NLP task에 적용할 수 있겠다