Dynamic Programming Encoding for Subword Segmentation in Neural Machine Translation

(He et al., ACL 2020)

出口　祥之

deguchi@ai.cs.ehime-u.ac.jp

2020/09/11　第 2 回 NLG/MT Reading Group

Links

Paper

https://www.aclweb.org/anthology/2020.acl-main.275/

Source Code

https://github.com/xlhex/dpe

Introduction

動的計画法を用いた新たなサブワード分割法を提案

- 目的言語文の分割を潜在変数と見做し，周辺化
- “Mixed character-subword Transformer”:
  原言語文が与えられたときの目的言語文の分割を獲得

NMT におけるサブワード分割

貪欲法:	バイトペア符号化 (BPE)[1], WordPiece[2]
確率的アルゴリズム:	ユニグラム LM[3], BPE-dropout[4]
動的計画法:	本論文の提案手法

---

[1] ``Neural Machine Translation of Rare Words with Subword Units", Sennrich et al., 2016.

[2] ``Japanese and Korean voice search", Schuster et al., 2012.

[3] ``Subword Regularization: Improving Neural Network Translation Models with Multiple Subword Candidates", Kudo, 2018.

[4] ``BPE-Dropout: Simple and Effective Subword Regularization", Provilkov et al., 2020.

Related Work (Greedy Segmentation)

BPE (Sennrich et al. 2016), **WordPiece** (Schuster and Nakajima 2012)

- 隣接する頻出サブワードから順に，予め指定した語彙数に到達するまで再帰的に結合 (BPE)
- 語彙数とデコード速度はトレードオフ
  - (語彙数を小さくするだけであれば文字単位でよい)
  - テキスト圧縮の技術を利用
  - 語彙数の上限を制約とし，文長が短くなるような分割を得るアルゴリズム

例: $\text{unconscious} \rightarrow \text{un} + \text{conscious}$

Related Work (Stochastic Segmentation)

ユニグラム LM (Kudo 2018)**, BPE-dropout** (Provilkov et al. 2020)

- 複数分割候補を得られる
  - ユニグラム LM: 尤度ベースでサンプリング
  - BPE-dropout: BPE 結合時に確率的に棄却
  - NMT 訓練時に分割を確率的に得ることでデータ拡張 (Data Augumentation) の効果
    - モデルの頑健性向上

例: $\text{unconscious} \rightarrow \{\text{un} + \text{concious}, \text{uncon} + \text{scious}\}$

Related Work (Dynamic Programming Algorithms)

音声認識 (Wang et al. 2017)

- 取り得る全ての分割や入出力間のアライメントの確率を動的計画法により計算

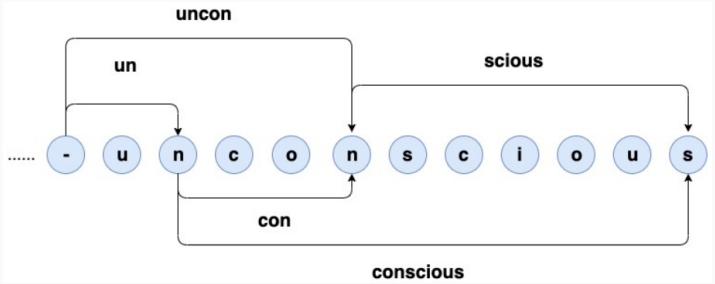
非自己回帰 NMT モデル (Chan et al. 2020; Saharia et al. 2020)

- **Imputer** (Chan et al. 2020)：
  Connectionist Temporal Classification (CTC) を用い，定数回のデコードで出力とその順序を予測
- 非自己回帰 NMT モデルに Imputer を適用 (Saharia et al. 2020)

Proposed Method

Latent Subword Segmentation - Definitions

目的言語文の分割を潜在変数とみなす



- M 個のサブワード: $\{\boldsymbol{y}_{z_i, z_{i+1}}\}_{i=1}^{M}$
  - $\boldsymbol{y} = (y_1, \ldots, y_T)$: 目的言語文の文字列
  - $\boldsymbol{z} = (z_1, \ldots, z_{M+1})$: 境界位置系列
    - $0 = z_1 < z_2 < \ldots < z_M < z_{M+1} = T$ (昇順)
  - $\boldsymbol{y}_{a,b}$: $(a+1)^{\text{th}}$ から b^{th} まで結合したサブワード

例:

- 辞書 $\mathcal{V} = \{\text{c, a, t, ca, at}\}$
- 目的言語文 $\boldsymbol{y} = \text{cat}$

$\boldsymbol{z}$	サブワード列
$(0, 1, 3)$	(c, at)
$(0, 2, 3)$	(ca, t)
$(0, 1, 2, 3)$	(c, a, t)

Latent Subword Segmentation - Likelihood

連鎖律を用いてサブワード列の対数尤度を表現

- 各サブワード位置において語彙の確率分布を生成

$$\log p(\boldsymbol{y}, \boldsymbol{z}|\boldsymbol{x}) = \sum_{i=1}^{|\boldsymbol{z}|} \log p(\boldsymbol{y}_{z_i,z_{i+1}}|\boldsymbol{y}_{z_1,z_2}, \ldots, \boldsymbol{y}_{z_{i-1},z_i}, \boldsymbol{x})$$

※ $\boldsymbol{x}$: 原言語文

- 殆どの NMT では $\boldsymbol{z}$ を暗黙的に $\log p(\boldsymbol{y}, \boldsymbol{z}) \approx \log p(\boldsymbol{y})$ と仮定

Latent Subword Segmentation - Latent Variable

$\boldsymbol{z} \in \mathcal{Z}_y$($\boldsymbol{y}$の分割集合) を潜在表現とみなす

- $p(\boldsymbol{y}|\boldsymbol{x}) = \sum_{\boldsymbol{z}} p(\boldsymbol{y}, \boldsymbol{z}|\boldsymbol{x})$ とする

$$\log p(\boldsymbol{y}|\boldsymbol{x}) = \log \sum_{\boldsymbol{z} \in \mathcal{Z}_y} \exp \sum_{i=1}^{|\boldsymbol{z}|} \log p(\boldsymbol{y}_{z_i, z_{i+1}} | \boldsymbol{y}_{z_1, z_2}, \ldots, \boldsymbol{y}_{z_{i-1}, z_i}, \boldsymbol{x})$$

※対数周辺尤度の下限: $\log p(\boldsymbol{y}|\boldsymbol{x}) \geq \log p(\boldsymbol{y}, \boldsymbol{z}|\boldsymbol{x})$

- 各サブワードの確率が条件部のコンテキストの分割に依存するため，巨大な空間 $\mathcal{Z}_y$ 上での厳密な周辺化は組み合わせ爆発を起こす
  - コンテキストが次に来るサブワードの確率に影響しないモデルが必要

A Mixed Character-Subword Transformer

文字に基づいてサブワードを生成する Transformer

- 条件部のコンテキストを文字のみに

$$\log p(\boldsymbol{y}, \boldsymbol{z}|\boldsymbol{x}) = \sum_{i=1}^{|\boldsymbol{z}|} \log p(\boldsymbol{y}_{z_i, z_{i+1}}|y_{z_1}, \ldots, y_{z_i}, \boldsymbol{x})$$

- $\boldsymbol{y}$ の各文字位置 t において，次に来るサブワード $w \in \mathcal{V}$ の分布を以下に基づいて生成

$$p(w|y_1, \ldots, y_t, \boldsymbol{x}) = \frac{\exp(f(y_1, \ldots, y_t)^\top e(w))}{\sum_{w' \in \mathcal{V}} \exp(f(y_1, \ldots, y_t)^\top e(w'))}$$

  - $f(\cdot)$: Transformer により条件部の計算
  - $e(\cdot)$: ソフトマックス層の重み

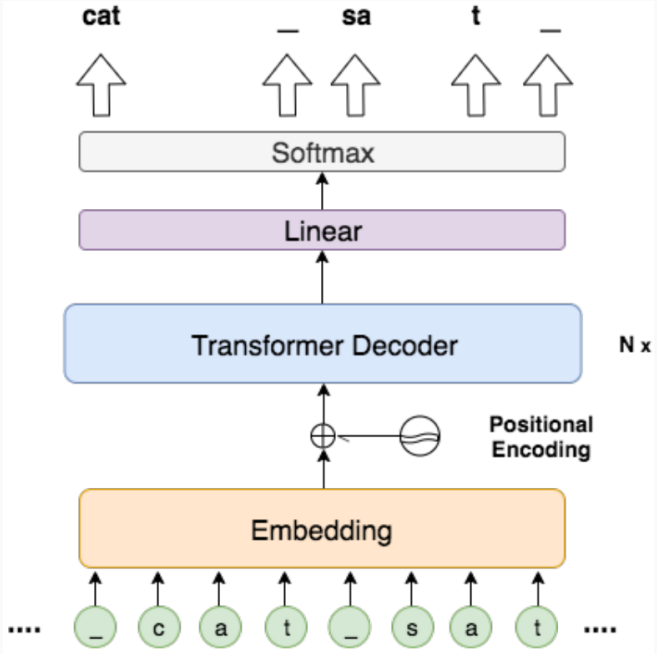
A Mixed Character-Subword Transformer

t ステップ目のモデル出力

(1) t ステップ目でサブワード w を生成

(2) サブワード w の文字をデコーダに入力 ($t+1$ から $t+|w|$ まで)

(3) $t+|w|$ ステップ目で次のサブワードを生成



Optimization

目的関数 $\mathcal{L}(\theta)$ を最大化

$$\mathcal{L}(\theta) = \sum_{(\boldsymbol{x},\boldsymbol{y})\in\mathcal{D}} \log P_\theta(\boldsymbol{y}|\boldsymbol{x})$$

必要な計算

- 周辺尤度の計算
- 対数周辺尤度の勾配計算

Exact Marginalization

動的計画法を用いて周辺尤度を計算

- サブワードの出力確率が文字のみによって得られるため動的計画法によって対数周辺尤度が計算可能

Algorithm 1 Dynamic Programming (DP) for Exact Marginalization

Input: $\boldsymbol{y}$ is a sequence of T characters, V is a subword vocabulary, m is the maximum subword length
Output: $\log p(\boldsymbol{y})$ marginalizing out different subword segmentations.

$\alpha_0 \leftarrow 0$
for $k = 1$ **to** T **do**
  $\alpha_k \leftarrow \log \sum_{j=k-m}^{k-1} \mathbb{1}[\boldsymbol{y}_{j,k} \in V] \exp\left(\alpha_j + \log P_\theta(\boldsymbol{y}_{j,k}|y_1, .., y_j)\right)$
end for
return α_T ▷ the marginal probability $\log p(\boldsymbol{y}) = \log \sum_{\boldsymbol{z} \in \mathcal{Z}_y} p(\boldsymbol{y}, \boldsymbol{z})$

- 計算量: $\mathcal{O}(mT)$
  - m : 語彙に含まれる最長の単語の文字数

Gradient Computation

計算量に関する問題点

- 通常の Transformer デコーダより 8 倍遅く，メモリ使用量も増加 [5]
  - DP アルゴリズムと文字レベルでの演算による系列長の増加が原因

対処法

- Transformer のレイヤ数を 6 から 4 に削減
- 16 ステップ分勾配蓄積 (Gradient Accumulation) してからパラメタ更新

[5] PyTorch での著者実装で比較

Segmenting Target Sentences

Dynamic Programming Encoding (DPE):
最大事後確率を持つ目的言語文の分割を探索

Algorithm 2 Dynamic Programming Encoding (DPE) for Subword Segmentation

Input: $\boldsymbol{y}$ is a sequence of T characters, V is a subword vocabulary, m is the maximum subword length
Output: Segmentation $\boldsymbol{z}$ with highest posterior probability.

for $k = 1$ **to** T **do**

$$\beta_k \leftarrow \max_{\{j \in [k-m,k-1] \,|\, \boldsymbol{y}_{j,k} \in V\}} \beta_j + \log P_\theta(\boldsymbol{y}_{j,k} | y_1, .., y_j)$$

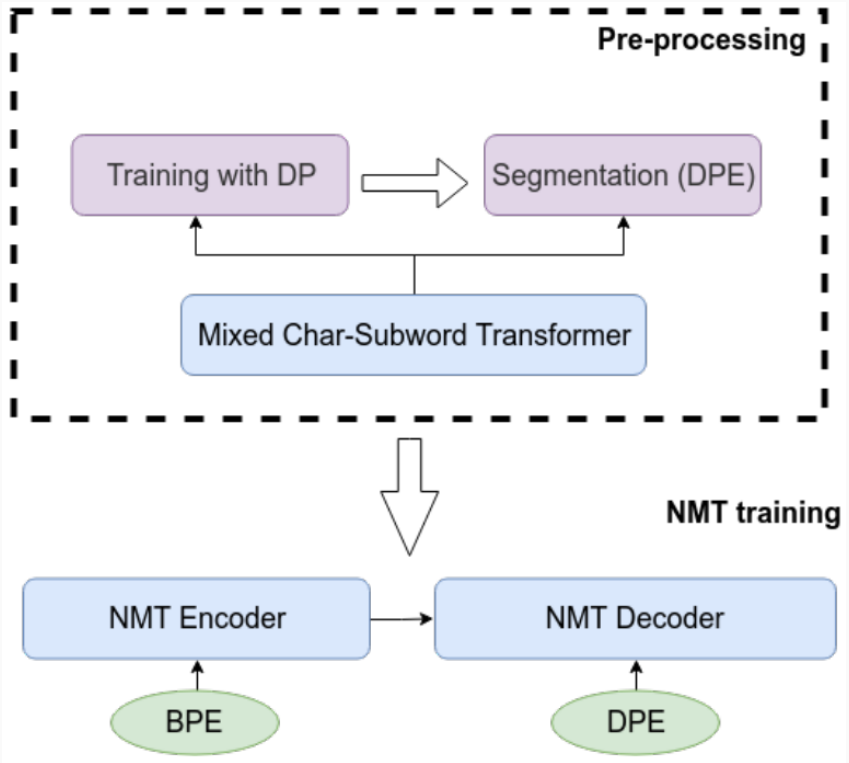
$$b_k \leftarrow \operatorname{argmax}_{\{j \in [k-m,k-1] \,|\, \boldsymbol{y}_{j,k} \in V\}} \beta_j + \log P_\theta(\boldsymbol{y}_{j,k} | y_1, .., y_j)$$

end for

$\boldsymbol{z} \leftarrow \text{backtrace}(b_1, .., b_T)$ ▷ backtrace the best segmentation using $\boldsymbol{b}$

Segmenting Target Sentences

- Mixed character-subword Transformer は
  訓練データの目的言語文の分割のためのみに使用
- 分割した文で通常のサブワード Transformer を訓練



Experiments

データセット WMT09 En-Hu, WMT14 En-De, WMT15 En-Fi, WMT16 En-Ro, WMT18 En-Et

モデル

NMT アーキテクチャ	Transformer base
分割 (原言語側)	BPE-dropout $(p = 0.05)$
(目的言語側)	DPE (提案手法)

Main Results

Method	BPE	BPE dropout	Δ_1	This paper	Δ_2
Source segmentation	BPE	BPE dropout		BPE dropout	
Target segmentation	BPE	BPE dropout		DPE	
En→De	27.11	27.27	+0.16	27.61	+0.34
En→Ro	27.90	28.07	+0.17	28.66	+0.59
En→Et	17.64	18.20	+0.56	18.80	+0.60
En→Fi	15.88	16.18	+0.30	16.89	+0.71
En→Hu	12.80	12.94	+0.14	13.36	+0.42
De→En	30.82	30.85	+0.03	31.21	+0.36
Ro→En	31.67	32.56	+0.89	32.99	+0.43
Et→En	23.13	23.65	+0.52	24.62	+0.97
Fi→En	19.10	19.34	+0.24	19.87	+0.53
Hu→En	16.14	16.61	+0.47	17.05	+0.44
Average	22.22	22.57	+0.35	23.12	+0.55

Segmentation Examples

BPE source:
Die G@@ le@@ is@@ anlage war so ausgestattet , dass dort elektr@@ isch betrie@@ bene Wagen eingesetzt werden konnten .
DPE target:
The railway system was equipped in such a way that electrical@@ ly powered cart@@ s could be used on it .
BPE target:
The railway system was equipped in such a way that elect@@ r@@ ically powered car@@ ts could be used on it .

BPE source:
Normalerweise wird Kok@@ ain in kleineren Mengen und nicht durch Tunnel geschm@@ ug@@ gelt .
DPE target:
Normal@@ ly c@@ oca@@ ine is sm@@ ugg@@ led in smaller quantities and not through tunnel@@ s .
BPE target:
Norm@@ ally co@@ c@@ aine is sm@@ ugg@@ led in smaller quantities and not through tun@@ nels .

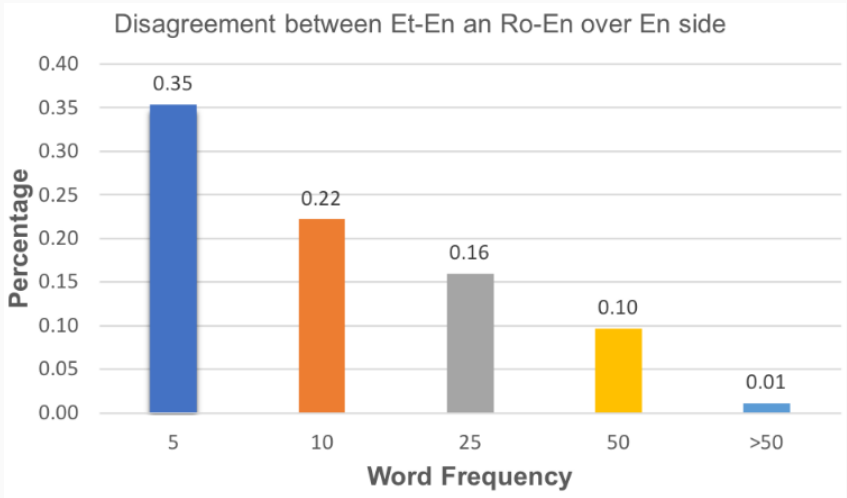
- 他の例は論文参照

Conditional Subword Segmentation

原言語文を条件部に入れず，LM で分割

Source Target	BPE drop BPE drop	BPE drop LM DPE	BPE drop DPE
En→Ro	28.07	28.07	28.66
En→Hu	12.94	12.87	13.36
Ro→En	32.56	32.57	32.99
Hu→En	16.61	16.41	17.05

同一の目的言語文で原言語側を変えて違いを比較



Conditional Subword Segmentation

原言語文が BPE-dropout によって変化することの有効性

Source Target	BPE drop DPE Fixed	BPE drop DPE On The Fly
En→Ro	28.58	28.66
En→Hu	13.14	13.36
En→Et	18.51	18.80
Ro→En	32.73	32.99
Hu→En	16.82	17.05
Et→En	24.37	24.62

DPE vs BPE

目的言語側の分割アルゴリズムを変えて比較

Source Target	BPE drop BPE	BPE drop BPE drop	BPE drop DPE
En→Ro	28.04	28.07	28.66
En→Et	18.09	18.20	18.80
Ro→En	32.40	32.56	32.99
Et→En	23.52	23.65	24.62

Conclusion

新たなサブワード分割法

Dynamic Programming Encoding を提案

- Mixed character-subword Transformer により
  目的言語文を分割
  - 目的言語文の分割を潜在変数と見做して周辺化
  - 条件部のコンテキストを文字にすることで
    動的計画法が適用可能に
  - 分割時は事後確率が最大となる分割を出力

- BPE だけでなく BPE-dropout と比較しても
  翻訳性能が向上