

進捗報告

1 あらすじ

サブワード化しよう.

2 進捗

- サブワード化したデータでの正例ラベルを喜楽・ニュートラル・驚愕としたときの2クラス感情推定

3 実験設定

BERTのトークナイザーを用いてサブワード化したデータを用いて、最終レイヤーのみのチューニング(BERT Last Layer), またはすべてのパラメータを固定させる(BERT Fixed) 条件で BERT fine-tuning を追加で行った. 正例ラベルとして喜楽・ニュートラル・驚愕の3パターンについて2クラス分類を行った. 識別器としては3層MLPを用いた.

MLPのパラメータを図1に, 学習パラメータを表2に示す. 学習率はoptunaによって最適化されたものを用いた. (試行回数5回)

先週, 意見としてあった, '!', '?' をトークンに置き換えることはせず, また入力id列の長さはサブワード化した時の最大系列長に合わせており, 固定サイズ(64, 128, ...) にはしていない.

表 1: MLP パラメータ

	MLP
(in,hidden,out)	(300,30,2)
dropout rate	0.5
activation function	tanh

表 2: 学習パラメータ

	実験
epoch	200
batch size	16
loss function	Cross Entropy Loss
optimizer	Adam

4 実験結果

表3に実験結果を載せる. 比較対象として前回までの実験結果も載せている. 表3よりサブワード化しない場合と比べて, 指定した正例ラベルによって精度の改善が若干見られたが全体的な優位性は見られなかった.

5 問題点

- train と validation のデータ分布が似ているため, 場合によっては validation の accuracy が1になっていたりする.
- 実験中 Connection reset by 192.168.0.11 port 22 が出た.

表 3: result

model		ギャグ			少女漫画			少年漫画			青年漫画			萌え系			5 タッチ平均			
		Acc	Recall	F1	Acc	Recall	F1	Acc	Recall	F1	Acc	Recall	F1	Acc	Recall	F1	Acc	Recall	F1	
Juman++	驚愕	BERT (Last Layer)	0.758	0.000	0.000	0.806	0.000	0.000	0.859	0.000	0.662	0.647	0.500	0.719	0.000	0.000	0.761	0.129	0.100	
	ニュートラル	BERT (Last Layer)	0.379	0.400	0.281	0.881	0.000	0.000	0.578	0.788	0.658	0.677	0.200	0.276	0.625	0.267	0.250	0.628	0.331	0.293
	喜楽	BERT (Last Layer)	0.833	0.400	0.421	0.567	0.579	0.603	0.797	0.083	0.133	0.800	0.357	0.435	0.656	0.455	0.476	0.731	0.375	0.414
		BERT (Fixed)	0.818	0.500	0.455	0.463	0.421	0.471	0.766	0.000	0.000	0.769	0.429	0.444	0.625	0.409	0.429	0.688	0.352	0.360
Subword	驚愕	BERT (Last Layer)	0.803	0.000	0.000	0.866	0.000	0.000	0.875	0.222	0.333	0.739	0.235	0.320	0.734	0.000	0.000	0.803	0.091	0.131
		BERT (Fixed)	0.758	0.000	0.000	0.836	0.000	0.000	0.875	0.444	0.500	0.677	0.353	0.364	0.750	0.167	0.200	0.779	0.193	0.213
	ニュートラル	BERT (Last Layer)	0.561	0.650	0.473	0.896	0.000	0.000	0.516	0.727	0.608	0.662	0.200	0.267	0.729	0.333	0.357	0.672	0.382	0.341
		BERT (Fixed)	0.485	0.450	0.346	0.910	0.000	0.000	0.563	0.697	0.622	0.631	0.250	0.294	0.703	0.200	0.240	0.658	0.319	0.300
	喜楽	BERT (Last Layer)	0.818	0.200	0.250	0.627	0.737	0.691	0.734	0.000	0.000	0.754	0.500	0.467	0.563	0.273	0.300	0.699	0.342	0.342
		BERT (Fixed)	0.758	0.200	0.200	0.552	0.684	0.634	0.797	0.000	0.000	0.785	0.357	0.417	0.719	0.455	0.526	0.722	0.339	0.355
ベースライン		0.85	0	0	0.43	0	0	0.81	0	0	0.78	0	0	0.66	0	0	0.71	0	0	

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
index	id	original	story_main_num	story_sub_num	koma	who	inner	speaker	what	wakati	emotion	subword					
0	0	0	TRUE	0	0	0	B	FALSE	B	AくんってAくんって	ニュートラル	Aくんっていつも	クールね				
1	0	0	FALSE	0	0	0	B	FALSE	B	AくんってAくんって	ニュートラル	Aくんって聞断断なく	クールね				
2	0	0	FALSE	0	0	0	B	FALSE	B	AくんってAくんって	ニュートラル	Aくんって常##常	クールね				
3	0	0	FALSE	0	0	0	B	FALSE	B	AくんってAくんって	ニュートラル	Aくんって毎毎	クールね				
4	0	0	FALSE	0	0	0	B	FALSE	B	AくんってAくんって	ニュートラル	Aくんってつね##に	クールね				
5	0	0	FALSE	0	0	0	B	FALSE	B	AくんってAくんって	ニュートラル	Aくんって年##から##年##じゅう	クールね				
6	0	0	FALSE	0	0	0	B	FALSE	B	AくんってAくんって	ニュートラル	Aくんって必ず	クールね				
7	0	0	FALSE	0	0	0	B	FALSE	B	AくんってAくんって	ニュートラル	Aくんって朝夕	クールね				
8	0	0	FALSE	0	0	0	B	FALSE	B	AくんってAくんって	ニュートラル	Aくんって絶えず	クールね				
9	0	0	FALSE	0	0	0	B	FALSE	B	AくんってAくんって	ニュートラル	Aくんって四##六##時##中	クールね				

図 1: dataset

6 現状の 4 コマストーリーデータセットの加工

図 1 に、現状独自に加工してある 4 コマストーリーデータセット [1] の一部を示す。

- id : セリフの区別用の数字. 同じ id なら元は同じセリフから生成されている.
- original : オリジナルのセリフかどうか
- story main num : 第 n 話
- story sub num : 第 n 話の左右どちらか
- koma : 何コマ目に属するか
- who : 主体
- speaker : 話者
- inner : 傍白 (主体にしか聞こえていないセリフ) かどうか
- what : セリフ
- wakati : what を juman++ で分かち書きにしたもの
- subword : wakati をさらに BERT のトークナイザーで分割したもの ([UNK] の場合は元の単語に戻している)

7 連続 n 文を入力とする実験のネットワークモデル

図 2 にネットワークの概略図を示す. s_i はそれぞれ連続する n 文のセリフを BERT の id 列に変換したものである. また, 組み合わせとしては同一の 4 コマに属し, かつ連続しているものを扱う. 各 4 コマ

の序盤に現れるセリフには参照できる過去のセリフが無いいため, 便宜上のセリフ [pad] を置くことで対処した. よって, 単語 id 列長を w とすると入力次元は $(batch \times n \times w)$ となる. このままでは BERT の入力次元に対応していないので, まず $(n \times batch \times w)$ へと軸を入れ替え, これを 1 次元目について各要素に分解し, これら n 個の次元数 $(batch \times w)$ のベクトルをそれぞれ BERT への入力とし, BERT の出力から [CLS] トークンに相当するベクトルのみを抜き取り, 先と逆の手順を踏むことで次元数 $(batch \times n \times 768)$ のテンソルを得ることができる. これを Bi LSTM, Self-Attention への入力とすることで末尾のセリフの感情推定を行う. このモデルの形で将来的に期待できることは, BERT の出力を stack する前に, 各 [CLS] Vector に元のセリフに対応するコマの画像の分散表現を concat でき, マルチモーダルでかつシーケンシャルな推定が可能となることであるが, 不安点としては BERT の利点を上手く利用できているのかという問題がある.

BERT は最終層のみをチューニングし, その他実験設定などは, 1 文のみを入力とする実験と同じである.

現状は, $n = 3$ として, テスト実験を回している. loss が下がっていることは確認できた. 計算時間は約 $3200sec/200epoch$.

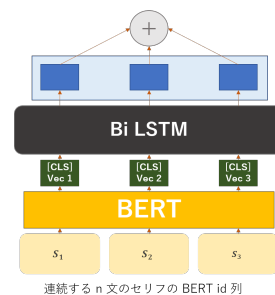


図 2: seq net

8 今後の実験予定

- 単語 id 列長を 64, 128 として実験
- 連続 n 文を入力とする実験

参考文献

- [1] 上野 未貴. 創作者と人工知能: 共作実現に向けた創作過程とメタデータ付与 4 コマ漫画ストーリーデータセット構築. 人工知能学会全国大会論文集, 2018.