

## 進捗報告

### 1 今週やったこと

- manga109 のセリフ 147918 データ を抽出後 Juman で分かち書きし, Doc2Vec のモデルを作成.
- Self-Attention ネットワークの実装. 動作確認をした.
- 情報工学実験 II のコードの書き直し中.

### 2 manga109 d2v モデル

情報工学実験 II での実験で, Embedding 用のモデルとして, Wikipedia や 青空文庫, 小説家になろう から抽出した文書から学習した d2v モデル用いたが, この中には漫画特有の口語表現などが学習されていない可能性から, manga109 のセリフを用いる. 一旦は追加学習ではなく manga109 のセリフのみから作成したモデルで Embedding を行う. 学習に用いたパラメータは図 1 のようである.

表 1: d2v parameters

parameter	value
vec_size	300
epochs	30
win_size	8
min_cnt	3
alpha	0.05
workers	4

### 3 Self-Attention ネットワークの実装

ひとまずは情報工学実験 II での実験設定とおおよそ同じで, 図 1 のように同一 4 コマに含まれているセリフを, 1 文ずつ Bidirectional LSTM に入力し, 各隠れ層での出力を Self-Attention 層で受け取り, Attention を計算し, タスクとしても同様に感情の 2 値分類 (喜楽 or その他) を解くことを目標とする. 学習パラメータは図 2, ネットワークパラメータ

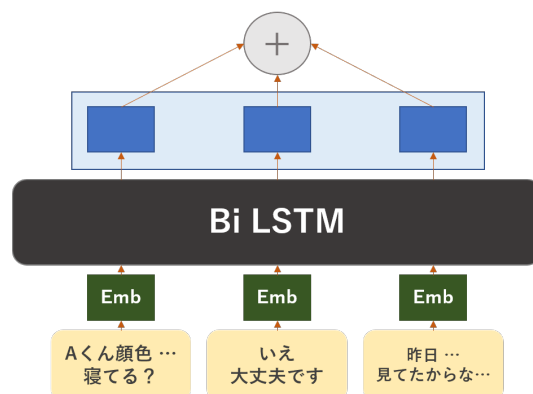


図 1: Self-Attention

は図 3 である. クラス重みは設定していない. 時系列長を 2 - 6 として実験を行った.

表 2: learning parameters

parameter	value
epoch	100
batch size	16
loss	CrossEntropyLoss
optimizer	Adam
learning rate	0.00001

表 3: network parameters

parameter	value
lstm_in	300
lstm_hidden	128
self_attn_in	128 × 2
attn_num_layers	3

### 4 結果

学習時間については 1 つの時系列長につき平均 7 分 / 100 epoch だった.

実験結果を表 4 にまとめた. そして, 情報工学実験 II での LSTM + 3 層 MLP を用いた実験結果との比較を示したのが図 2 である.

時系列長 N	LSTM + MLP			Self-Attention		
	5 タッチ平均			5 タッチ平均		
	Acc	Recall	F 値	Acc	Recall	F 値
2	0.51	0.64	0.41	<b>0.62</b>	0.42	<b>0.53</b>
3	0.50	0.62	0.39	0.56	0.52	0.51
4	0.50	0.65	0.40	0.58	0.50	<b>0.53</b>
5	0.48	0.66	0.39	0.55	<b>0.57</b>	0.52
6	<b>0.52</b>	<b>0.71</b>	<b>0.44</b>	0.55	0.52	0.51
ベースライン	<b>0.71</b>	<b>0.00</b>	<b>0.00</b>	<b>0.71</b>	<b>0.00</b>	<b>0.00</b>

図 2: compare LSTM and Self-Attention

## 5 考察

どの場合も約 20 epoch 目付近から学習が止まっていた. (result ディレクトリ参照)

LSTM + MLP と比べて, 正例 (喜楽ラベル) の recall は落ち込んだが, Accuracy, F1 値ともに向上した. しかし, 各タッチ・時系列長ごとに見ると, 性能の増減の傾向に差がありその原因については今後詳しく考えていく必要がある.

## 6 その他 問題点

- train データを train と validation に分ける際, test は 各タッチの 5 - 10 話を用い, train と validation は 1 - 5 話からランダムで分けているが, データセットの仕様上, 中身が似たようなものになる可能性が高く, 汎化性能に欠ける.
- Data Augmentation の手法で日本語 WordNet を用いることで発生する, 意味の乖離.
- optuna を用いたハイパーパラメータのチューニングをしていない.

## 7 課題

- 森先生と大工大の上野先生に 30 話までであるらしい追加データをお願いして, train と validation を話区切りで分けたい.
- Data Augmentation の手法の改善案
- optuna でチューニングできるようにする
- BERT の論文等で構造の理解
- 現状タッチごとに異なるモデルを作っているの  
で, タッチ情報を表す one-hot-vector 等を用いて  
1 つのモデルで性能を確かめることも考える.

## 8 質問

Attention 層を複数にする意味

## 9 次に読むべき論文

- BERT について書かれているもの.
- pointer-generator network について書かれているもの.

表 4: result

seq_len	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
2	0.58	0.30	<u>0.45</u>	0.72	0.82	<u>0.70</u>	0.53	0.42	0.45	0.71	0.21	0.53	0.58	0.36	0.53	0.62	0.42	<u>0.53</u>
3	0.36	0.50	0.33	0.70	0.87	0.68	0.48	0.42	0.42	0.69	0.43	0.59	0.56	0.41	0.52	0.56	0.52	0.51
4	0.39	0.50	0.36	0.66	0.58	0.66	0.53	0.58	<u>0.48</u>	0.69	0.36	0.57	0.63	0.50	<u>0.59</u>	0.58	0.50	<u>0.53</u>
5	0.42	0.60	0.39	0.64	0.61	0.64	0.44	0.67	0.42	0.74	0.57	<u>0.65</u>	0.53	0.41	0.50	0.55	0.57	0.52
6	0.39	0.70	0.37	0.61	0.45	0.61	0.52	0.67	<u>0.48</u>	0.66	0.36	0.54	0.56	0.41	0.52	0.55	0.52	0.51
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