

進捗報告

1 今週やったこと

- BERT[1] の勉強と動作確認.
- optuna の動作確認.
- d2v モデルの再構築.
- B3 実験のコードの手直し.

2 BERT の勉強と動作確認

Transformer の構造から入って、今は BERT のモデルの構造について勉強中。また、“transformers” ライブラリ (元 “pytorch-pretrained-bert”) を元に簡単な動作確認を行った。モデルは日本語 Wikipedia から全 1,800 万文を用いて事前学習させたモデル¹を利用した。

入力は、4 コマデータセットのセリフを Juman++ で分かち書きしたものに、“[CLS]”、“[SEP]” トークンを付加したもの。出力として、入力文全体の特徴量を持っている “[CLS]” トークンに対応する 768 次元のベクトルを得た。

以下は自分用のメモである。

2.1 BERT の特徴

2.1.1 文脈に依存した単語ベクトル表現が得られる

同じ単語であれば、初めの Embedding によって同じベクトルとして表されるが、12 段の Transformer を経る内に単語の位置情報ベクトルが変化していくため、最終的には同じ単語であっても文脈によって異なる単語ベクトル表現が得られる。

2.1.2 自然言語処理タスクで fine-tuning が容易

BERT をベースに様々な自然言語処理タスクを行わせるためには、まず事前学習した重みパラメータを設定し、BERT モデルから出力したテンソルを実施したいタスクに応じてアダプターモジュールを追加し、タスクに応じた出力を得る。

例えば、2 値分類の感情分析であれば、アダプターモジュールとしては 1 つの全結合層を付加するだけで、文章の感情分析が可能となる。学習時にはベースとなる BERT と付加した全結合層の両方を fine-tuning で学習させる。

2.1.3 Attention によって可視化と解釈が容易

ディープラーニングにおいて、モデルの出力についての解釈性・説明性が問われる昨今において、Attention による可視化によって我々が推論結果の説明について考えやすい。

3 d2v モデルの再構築

B3 実験で用いた Wikipedia + ならう + 青空文庫からなる大規模コーパスは、その大きさながらも良い識別結果が得られなかったため、一旦約 10 分の 1 のサイズで Wikipedia のみのコーパスを用いたモデル (model wiki)、前回報告した manga 109 のセリフコーパスを用いたモデル (model manga109)、これら 2 つのコーパスを両方用いたモデル (model wiki + manga109)、の計 3 つの Doc2Vec のモデルを作成した。学習パラメータは表 1 で共通している。

表 1: d2v parameters

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

これらのモデルを用いて、B3 実験をやり直して結果をまとめることを課題とする。

4 B3 実験のコードの手直し

1 文を単語ごとに分散表現にしてから LSTM に入力する手法の追加。入力列長は 1 セリフ内の単語数

¹<http://nlp.ist.i.kyoto-u.ac.jp>

の最大値に合わせ, それ未満のセリフについては, 前側にゼロパディングする. また, 未知語の場合も, その分散表現をゼロベクトルとする “<unk>” トークンに置き換える. (または, 予測値を用いる?)

5 課題 優先度準

- BERT の fine-tuning を行う.
- 新しく作成した d2v モデルと手法を用いた B3 実験の再実験および結果の比較.
- 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する.
- Data Augmentation の手法の改善案.

参考文献

- [1] Chang M.-W. Lee K. Devlin, J. and K Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv:1810.04805*, 2018.