令和 5 年度茨城大学工学部情報工学科卒業研究論文

日本語 RetNet の構築と評価

所属 情報工学科

著者 齋藤駿輝 (20T4040G)

指導教員 新納浩幸教授

令和 6 年 2 月 2 日 (金)

令和 5 年度茨城大学工学部情報工学科卒業研究論文

日本語 RetNet の構築と評価

著者

齋藤駿輝 (20T4040G)

指導教員

新納浩幸教授

論文要旨

卒論のテンプレートを作成する.

目次

第 1 章　序論 **6**

第 2 章　関連研究 **7**

2.1　Transformer 7

2.2　T5 7

2.3　BART 8

第 3 章　提案手法 **9**

第 4 章　実験 **11**

4.1　構築 11

4.2　評価 12

第 5 章　考察 **14**

5.1　日本語 RetNet の性能 14

5.2　日本語 RetNet の利用可能性 15

第 6 章　結論 **16**

謝辞 **17**

参考文献 **18**

付録 **19**

A　プログラムの載せ方 19

表目次

3.1 日本語 RetNet の構築と評価 9

4.1 ？？？ 11
4.2 ？？？ 12
4.3 ？？？ 13

5.1 ？？？ 14

図目次

第１章

序論

第 2 章

関連研究

2.1 Transformer

Transformer [1] は、2017 年 6 月に Vaswani らにより発表されたエンコーダ-デコーダ構成のシーケンス変換モデルである。従来の主流は、再帰型ニューラルネットワーク (Recurrent Neural Networks；RNNs) や畳み込みニューラルネットワーク (Convolutional Neural Networks；CNNs) をベースとした Encoder-Decoder 構成のシーケンス変換モデルであった。しかし、Transformer は、再帰も畳み込みも行わない Scaled Dot-product Attention 機構を新たに提案し、その性能の良さから現在の主流となり、様々な派生モデルが多く誕生している。

$$Attention(Q, K, V) \tag{2.1}$$

2.2 T5

Replace corrupted spans

`original texts:`	太郎␣は␣花␣子␣に␣渡し␣た␣。
`masked texts:`	太郎␣<M>␣<M>␣子␣に␣<M>␣た␣。
`inputs:`	太郎␣<X>␣子␣に␣<Y>␣た␣。
`targets:`	<X>␣は␣花␣ <Y>␣渡し␣<Z>

T5(Text-to-text Transformer) は、2019 年 10 月に Raffel らにより発表された Transformer の Encoder-Decoder ベースのモデルである。入力とターゲットをテキストとし、

fine-tuning を行うことで様々なタスクに適応することができる。T5 の論文中では様々な事前学習の検証がされているが、ベースラインモデルとして以下の方法が提案されている。

2.3 BART

BART(Bidirectional and Auto-Regressive Transformers) は、2019 年 10 月に Lewis らにより発表されたモデルである。BART は、Transformer の Encoder ベースのモデルである BERT と、Transfomer の Decoder ベースのモデルである GPT を組み合わせた Encoder-Decoder モデルである。T5 と同様に、入出力をテキストとし、fine-tuning により様々なタスクを行うことができる。BART の論文中では、事前学習の方法として特定の方法がベースラインとなっているわけではないが、以下の方法が提案されている。

- Token Masking

  ランダムにトークンを選択し、マスキングを行う。
- Token Deletion

  ランダムなトークンが削除される。
- Text Infilling

  平均 3 個の連続したトークンがマスキングされる。
- Sentence Permutation

  句読点区切りの完全な一文毎に区切られ、ランダムな順序に入れ替えられる。
- Document Rotation

  ランダムにトークン選ばれ、そのトークンから始まるように文書が回転される。

第 3 章

提案手法

日本語 RetNet の構築と評価は　に示される手順で行う。構築には日本語 Wikipedia を用い、評価には DQ7 データセットを用いる。また、評価比較には事前学習済み BART ／ T5 を用い、評価指標には推論時 GPU ／推論時間／ドメイン外 PPL ／ドメイン内 PPL を用いる。

表 3.1: 日本語 RetNet の構築と評価

	構築	評価
対象	`retnet-medium`	`retnet-medium`
コーパス	日本語 Wikipedia	DQ7 データセット
比較対象	–	bart, t5
評価指標	–	推論時 GPU, 推論時間,

- 日本語 RetNet の構築
  1. 日本語 Wikipedia を前処理し、訓練用 (train) ／検証用 (valid) ／評価用 (test) に分割する。
  2. 訓練用日本語 Wikipedia を用いて Sentencepiece トークナイザーを構築する。
  3. 訓練用日本語 Wikipedia を用いて日本語 RetNet を構築する。

- 日本語 RetNet の評価
  1. DQ7 データセットを訓練用 (train) ／検証用 (valid) ／評価用 (test) に分割する。
  2. (ドメイン外評価) 事前学習済み RetNet ／ BART ／ T5 において評価用 DQ7

データセットでのパープレキシティ (PPL) を計測する。

3. (ドメイン適応) 事前学習済み RetNet ／ BART ／ T5 を訓練用 DQ7 データセットで追加学習する。
4. (ドメイン内評価) 追加学習した RetNet ／ BART ／ T5 において評価用 DQ7 データセットでのパープレキシティ (PPL) を計測する。

第 4 章

実験

4.1 構築

4.1.1 日本語 Wikipedia の準備

本稿では、トークナイザー及びモデルの構築に日本語 Wikipedia を使用する。ここでは、日本語 Wikipedia を訓練用データ (train) ／検証用データ (valid) ／評価用データ (test) に分割し、それぞれ一行一段落のテキストデータに変換する。各データの記事数を？？？に示す。また、日本語 Wikipedia の準備手順を？？？に示す。

表 4.1: ？？？

データ	割合	記事数
train	90%	2027087
valid	5%	112616
test	5%	112616

4.1.2 SentencePiece の構築

本稿では、トークナイザーとして Sentencepiece を採用する。Sentencepiece の構築に際しては公式の GitHub[*1]を参考に、日本語 Wikipedia の訓練用データで学習を行った。

[*1] https://github.com/google/sentencepiece

4.1.3　日本語 RetNet の構築

RetNet は、Pytorch をベースに Meta AI が開発した Sequence2Sequence モデルを学習させるためのツールキットである Fairseq を使うことを想定している。RetNet を Fairseq で構築するに際して、データセットを分かち書きする必要があるため、まず各データを構築した SentencePiece で一行毎にサブワード分割する。その後、Fairseq の fairseq-process コマンドで訓練用データセットの前処理を行い、Fairseq の公式 GitHub の言語モデリングに関する記載[*2]を参考に fairseq-train コマンドで学習する。

4.2　評価

4.2.1　DQ7 データセットの準備

本稿では、ドメイン外コーパスとしてドラゴンクエスト Ⅶ のキャラクターの発話を収集した DQ7 データセット [2] を使用する。このデータセットには、ドラゴンクエスト Ⅶ のキャラクターであるアイラ、フォズ、ガボ、キーファ、マリベル、メルビン、リーサ姫の発話が含まれており、あらかじめ訓練用／検証用／評価用でデータが分かれている。このうち評価には、訓練用に全キャラクターの発話を用い、検証用及び評価用にマリベルの発話を用いる。DQ7 データセットの概要を？？？に示す。太字部分が実際に使用するデータである。

表 4.2: ？？？

発話者名	発話数	train	valid	test
アイラ	1814	**1635**	36	143
ガボ	3061	**2757**	61	243
キーファ	1613	**1454**	32	127
フォズ	74	**68**	2	4
マリベル	4131	**3720**	**82**	**329**
メルビン	2172	**1956**	44	172
リーサ姫	147	**135**	3	9

[*2] https://github.com/facebookresearch/fairseq/tree/main/examples/language_model

4.2.2　評価対象の追加学習

評価対象は retnet-medium、および比較として t5-base、bart-base、bart-large である。追加学習は訓練用 DQ7 データセットを使用し、それぞれ事前学習と同様の方法で行う。early t5-base は、？？？の方法で、HuggingFace の Trainer クラスを利用して学習した。また、学習に係るハイパーパラメータの値は、モデル作成者の GitHub を参考にした。bart-base 及び bart-large は、？？？の方法で、モデル作成者の GitHub を参考に、Fairseq を通して学習した。retnet-medium は、4.1.3 項を参照のこと。

4.2.3　評価指標と実験結果

評価には、評価用 DQ7 データセットを用いる。評価指標は、推論時 GPU、推論時間、ドメイン外 PPL、ドメイン内 PPL である。推論時 GPU は、推論時に使用された GPU の最大値である。また、ドメイン外 PPL は DQ7 データセットでの追加学習前における PPL で、ドメイン内 PPL は追加学習後における PPL である。実験結果を？？？に示す。

表 4.3: ？？？

モデル	推論時 GPU	推論時間	ドメイン外 PPL	ドメイン内 PPL
bart-base(140M)	1301	1.1	2.73	2.57
bart-large(400M)	3485	1.5	2.96	2.19
t5-base(220M)	2483	4.5	34.09	2.04
retnet(270M)	2917	2.6	48.91	19.67

第 5 章

考察

5.1 日本語 RetNet の性能

表 5.1: ？？？

モデル	1M 当たりの推論時 GPU	1M 当たりの推論時間
bart-base(140M)	9.29	0.00786
bart-large(400M)	8.71	0.00375
t5-base(220M)	11.29	0.0204
retnet(270M)	10.80	0.00963

実験結果より、日本語 RetNet の性能を評価する。まず推論時 GPU と推論時間であるが、パラメータ数の違いを考慮し、1M 当たりの値に直したものを？？？に示す。？？？より、推論時 GPU はモデルサイズにサイズに伴って大きい値を示した。1M 当たりの推論時 GPU を比較すると、それほど差はなく、retnet が特に優位であるという結果にはならなかった。また推論時間は、1M 当たりの推論時間では t5-base が最も大きく、bart-large が小さい値を示した。しかし、推論時間全体としては秒単位の差で、あまり変わらなかった。これは、単に評価用データセットのサイズが小さいためと考えられる。

次に PPL に関して見ていく。はじめに bart-base、base-large のドメイン外 PPL が異常に低い値になっている。原因不明であるが、追加学習前よりこのような値を示すのは現実的に考えにくいため、評価から除外する。よって、t5-base と retnet で比較すると、ドメイン

外 PPL 及びドメイン内 PPL においての両者において、retnet よりパラメータ数の小さい t5-base が良い値を示した。総合的に判断すると、推論時間にやや差があるものの、t5-base が最も有用である判断して良いだろう。

5.2　日本語 RetNet の利用可能性

？？？らは、経験的事実として、検証用データセットでのパープレキシティにおいて、モデルサイズが 2B より大きい場合に RetNet が Transformer を上回ると示している。また、モデルサイズ 6.8B における推論時 GPU に関して、入力シーケンス長が大きくしていくと、Transformer の推論時 GPU は直線的に増加していくが、RetNet の推論時 GPU はほぼ変化しない。l 同様に、モデルサイズ 6.8B における推論時間に関して、バッチサイズが大きくなるにつれて、Transformer の推論時間は大きくなるが、RetNet の推論時間は、ほとんど変わらないことが示されている。本稿では、モデルや評価に用いたデータセットが小規模であったため、RetNet の論文と同様の規模で行えるのであれば、日本語 retnet の有用性を示せる可能性はある。しかし、本稿と同等の規模での使用ならば、あまり有用性はないと考えられる。

第 6 章

結論

謝辞

参考文献

[1] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017.

[2] 岸野望叶, 古宮嘉那子, 新納浩幸. T5 による特定キャラクター風発話への変換とその言語モデルの構築. Technical Report 13, 茨城大学大学院理工学研究科情報工学専攻, 東京農工大学大学院工学研究院先端情報科学部門, 茨城大学大学院理工学研究科情報科学領域, sep 2022.

付録

A　プログラムの載せ方

C 言語のソースコードを A.1 に示す.

ソースコード A.1: hallo.c

```
1 #include<stdio.h>
2
3 int main(void){
4  printf("Hallo,␣world!\n");
5  return 0;
6 }
```

また，Python のソースコードを A.2 に示す.

ソースコード A.2: numpy.py

```
1 import numpy as np
2
3 a = np.array([1, 2, 3])
4
5 print(a)
```