Vol.2021-NL-250 No.6
2021/9/28

Masked Language Modeling を用いた Replaced Token Detection 型事前学習の汎化性の改善検討

麻岡 正洋[1] 坂井 靖文[1] 笠置 明彦[1] 田原 司睦[1]

概要：小規模 Masked Language Modeling（MLM）の Generator による入力の類似文生成と，Replaced Token Detection（RTD）で Generator が書き換えた場所を推定する Discriminator を組み合わせた ELECTRA が，BERT などの既存事前学習手法に比べて同程度の精度を高速に学習できることが報告されている．しかし，我々は日本語ベンチマークにおいて ELECTRA の精度が頭打ちになるという事象を観測しており，この原因として，RTD が入力を書き換えたか書き換えていないかの二択問題という簡単な問題を解いており，汎化性が低くなるからではないかという仮説を立てた．そこで，我々は回答候補の多い MLM を RTD 学習に混ぜることで汎化性を高くすることを試みた．具体的には Generator が書き換えた文に再度マスクをして，Discriminator にマスクをした箇所は MLM として，マスクをしていない箇所は RTD として学習する方法を提案する．この方法によって，RTD の事前学習モデルの精度を向上できるかどうかを検証した．

キーワード：ELECTRA，Masked Language Modeling，Replaced Token Detection，深層学習，BERT

A Study on Improvement of Generalizability of Replaced Token Detection Pre-Training Using Masked Language Modeling

MASAHIRO ASAOKA[†1] YASUFUMI SAKAI[†1] AKIHIKO KASAGI[†1]
TSUGUCHIKA TABARU[†1]

Abstract: ELECTRA is a learning method that combines similar sentence generation by generator based on masked language modeling (MLM) and replaced token detection (RTD) by discriminator. It has been reported that ELECTRA can learn the same level of accuracy faster than existing pre-training methods such as BERT. But we observed an event in which the accuracy of ELECTRA was not high in the Japanese benchmark. We hypothesized one cause for this; because RTD solves the simple two-choice problem of whether the input is rewritten or not, it reduces the generalizability of pre-training. Therefore, we tried to improve generalization by mixing MLM which has many choices with RTD. Specifically, we propose a pre-training method that masks generator's output again, makes discriminator learn the masked position as MLM and the unmasked position as RTD. We verified whether the accuracy of the pre-training model of RTD can be improved.

Keywords: ELECTRA, Masked Language Modeling, Replaced Token Detection, Deep Learning, BERT

1. はじめに

BERT[1]に代表される事前学習を必要とするモデルが多くの自然言語処理ベンチマークで最高精度達成を実現している．これらのモデルは，図 1 のように大規模なラベルなしデータを使って汎用的な事前学習済みのモデルを作成し，その事前学習済みモデルを使って用途に応じた小規模なラベルありデータへの転移学習を行う．機械翻訳や質問応答などの転移学習の認識精度は事前学習時の学習状況に依存するためデータを集める事が比較的容易なラベルなしデータを大量に与える事で高精度を実現している．

事前学習の代表的な手法は Masked Language Modeling（MLM）をベースにしており，確率的にマスクした単語を周囲の単語から推測する問題をモデルに与える．大規模なラベルなしデータから学習データを自動生成できることおよび学習の並列実行が可能である点が優れている．

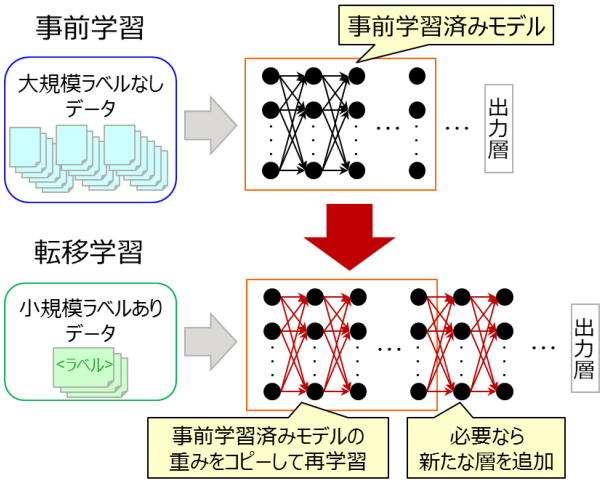

図 1 事前学習モデルに基づく転移学習のイメージ

しかし，MLM では実際に学習するのはマスクした部分

1 富士通株式会社
Fujitsu Ltd.

の周辺のみであることからマスクする確率に学習効率が依存し，またマスクする確率を高めるとヒントとなる周辺部分の単語が少なくなって問題として成立しなくなるため，学習効率を上げるのが難しいという課題があった．

その問題を解消するために，ELECTRA[2]が提案されている．ELECTRA は，BERT と同様の構成となる Generator と Discriminator の 2 種類のニューラルネットワークで構成される. ELECTRA の事前学習では，小規模 MLM である Generator による入力の類似文生成と，Generator が書き換えた場所を推定する Replaced Token Detection（RTD）を行う Discriminator を組み合わせる．Discriminator は後の転移学習に用いるため，Discriminator を学習させるのがこの学習手法の目的である．RTD は与えられた全ての単語に対して書き換え有無を検討するため，MLM などの既存手法に比べて学習効率が高く，高速に学習できることが報告されている．

しかし，我々は日本語ベンチマークタスクにおいて ELECTRA の認識精度が頭打ちになり, MLM に及ばないという事象を観測した．この原因として，RTD では Generator が入力を書き換えたかどうかの二択問題という単純な問題を解いているため，事前学習モデルの汎化性が低くなるからではないかという仮説を立てた．

仮説を確かめるため，回答の選択肢が多く複雑な問題でなる MLM を RTD 学習に混ぜて, 同時に学習を行うことで入力全体を学習しつつ汎化性を高くすることを試みた．具体的には Generator が穴埋めした文に再度マスクをして，Discriminator にはマスクをした箇所は MLM として，マスクをしていない箇所は RTD として学習するハイブリッドな方法を提案する．この方法によって，学習速度を保ちつつ転移学習の高精度化を実現できるかどうかを検証した．

2. Replaced Token Detection 型事前学習

ELECTRA が採用している事前学習手法である RTD について説明する．RTD のイメージを図 2 に示す．

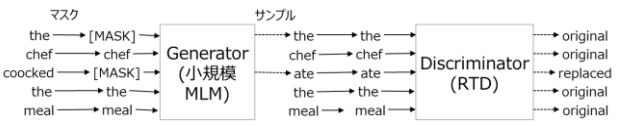

図 2　Replaced Token Detection のイメージ

図 2 のように RTD では Generator と Discriminator の 2 種類のニューラルネットワークを学習する．

Generator は MLM であり, 入力のラベルなしデータのトークン（単語）の一部をマスクしたデータを入力し，マスクしたトークンの元々のトークンを推定するように学習する．なお，BERT などでは通常 15%のトークンをマスクするが, ELECTRA では Generator が正しくマスクに対して穴埋めする事が目的ではないため，Base サイズでは 15%, Large サイズでは 25%のトークンをマスクするとよい，と報告されている[2].

Discriminator では，Generator で推定した結果を入力し，トークンが元々の入力データと置き換わっているか（replaced）または置き換わっていないか（original）を推定するように学習する．

RTD の事前学習は以下のように定式化される．

$$\min_{\theta_G,\theta_D} \sum_{x\in X} L_{MLM}(x,\theta_G) + \lambda L_{Disc}(x,\theta_D)$$

ここで, xは入力データXに含まれるトークンで, θ_G,θ_Dはそれぞれ Generator と Discriminator のパラメータであり，$L_{MLM}(x,\theta_G)$ が Generator の損失関数，$L_{Disc}(x,\theta_D)$ が Discriminator の損失関数で，λは損失関数の重みである．つまり，RTD の事前学習は Generator と Discriminator の損失関数の和を最小化するパラメータを求める問題と言える．

Generator があまり精度よく推定すると Discriminator の入力の大部分が original となってしまい，学習できなくなってしまうので，Generator はある程度誤りを含む推定をすることが望ましい．そこで，Generator は Discriminator よりもモデルサイズを小さくするようにする．ELECTRA では，Generator のモデルサイズを Base サイズでは Discriminator の 1/3，Large サイズでは 1/4 の規模にしている．

Discriminator ではすべてのトークンについて，置き換わっているかどうかの判定を行うため，マスクした部分のみの判定を行う MLM に比べて学習効率がよいことが報告されている. 具体的には, 1/4 以下の計算量で RoBERTa[3] や XLNet[4] と同等の GLUE[5]スコアを達成している．

3. RTD の課題

我々は，大規模な日本語データセットによる事前学習の性能を評価するために，PyTorch ベースの並列分散フレームワークである Megatron-LM[6] 上に BERT および ELECTRA を実装した. 日本語データセットには, Wikipedia 日本語版[7]，青空文庫[8]，OSCAR[9]日本語データの 3 種類のオープンデータを使用した．データセットのサイズは 78.1GB，約 2,700 万サンプルである．そのデータセットから 1,000 万サンプルを無作為に抽出し，Sentencepiece[10]を利用して 24,000 サイズの語彙ファイルを作成した．

転移学習には，オープンデータである Twitter 評判分析[11]のデータセットの 114,955 件のうち，10001 カテゴリのデータ 3,761 件を使用した．Twitter 評判分析には 1 つのツイートに対して複数のラベルが付与されているが，今回は positive & negative, positive, negative, neutral の 4 種類に分ける問題とした．4 種類に分けられないデータは削除した．なお，10001 カテゴリを選んだ理由は，positive, negative, neutral のサンプル数の偏りが比較的小さく，正当に評価しやすいと考えたからである．また比較的小規模のデータで実施することで評価サイクルを増やすことも意図している．

MLM で事前学習を行った BERT-Large モデルと RTD で事前学習を行った ELECTRA-Large モデルに対し，Twitter 評判分析の転移学習を行った結果を表 1 に示す．

表 1　BERT-Large, ELECTRA-Large の精度

モデル	Twitter 評判分析精度
BERT-Large	76.12%
ELECTRA-Large	69.19%

表 1 より，BERT-Large にくらべて ELECTRA-Large は精度が低くなっていることが分かる．

BERT-Large と ELECTRA-Large の事前学習の試行回数と転移学習の検証精度の関係を図 3 に示す．ここで横軸の iteration は事前学習でミニバッチデータを処理した回数を表す．

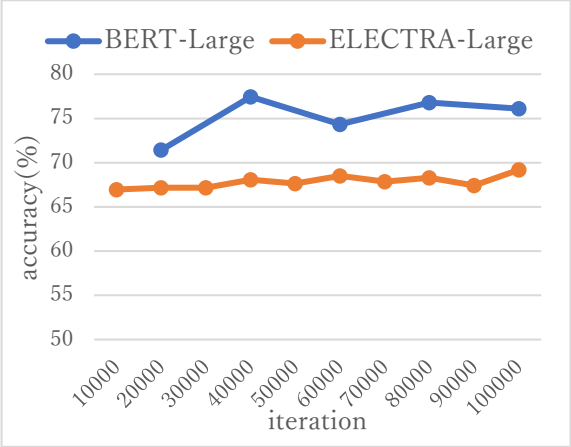

図 3　BERT, ELECTRA の事前学習の試行回数と検証精度の関係

図 3 のように，学習効率については，ELECTRA-Large は 10,000 iteration でほぼ最高精度近くまで到達しているのに対し，BERT-Large は 40,000 iteration で最高精度近くに到達していることが分かり，ELECTRA-Large のほうが学習効率はよいと思われる．しかし，いずれの地点においても BERT-Large の精度には届いていない．

ELECTRA の精度が向上しない原因として，我々は RTD では Generator が入力を書き換えたかどうかの二択問題という単純な問題を Discriminator が解いているため，語彙サイズだけの選択肢があり複雑な問題を解いている MLM に比べて事前学習モデルの汎化性が低くなるからではないかという仮説を立てた．また，MLM ではマスク化したトークンに対し正しいトークン情報を与えているため，その誤差から各トークンの多次元空間上での特徴量を修正していくが，RTD では多次元空間を面で判定できるようにしか各トークンの多次元空間上での特徴量を修正しないため，最終的に高精度な事前学習を行う場合は多くの試行回数が必要と考えられる．

4. 提案手法

我々はその仮説を検証するために，RTD の Discriminator にも複雑な MLM の問題を解かせることによって，学習効率を保ちつつ事前学習モデルの汎化性を向上させる手法を提案する．提案手法のイメージを図 4 示す．

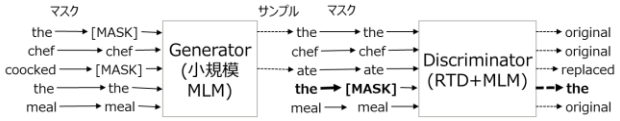

図 4　提案手法のイメージ

図 4 のように，Generator 出力の一部を再度マスクした結果を Discriminator に入力し，マスクしていない箇所については RTD で，マスクした箇所については MLM で学習することにした．なお，Generator で推定した箇所，つまり Generator 入力で既にマスクした箇所は Discriminator の入力ではマスクしないようにした．これは，Generator が書き換える箇所をマスクすることによって，書き換えられた箇所が減り RTD の学習が遅くなることを避ける意図がある．

提案手法の事前学習は以下のように定式化した．

$$\min_{\theta_G,\theta_D} \sum_{x\in X} L_{MLM}(x,\theta_G) + \lambda L_{Disc}(x,\theta_D) + \mu L_{Disc2}(x,\theta_D)$$

ここで，$L_{Disc2}(x,\theta_D)$ は Discriminator の MLM の損失関数，μは損失関数の重みである．つまり，提案手法の事前学習では，RTD の 2 つの損失関数の和に Discriminator の MLM の損失関数を加算したものを最小化することになる．

5. 実験

提案手法の有効性と仮説の検証を行うために，実験を行った．

事前学習の入力データセットと転移学習の入力データおよび事前学習のモデルサイズは 3 節と同様である．また，提案手法と ELECTRA で共通する部分，すなわち Generator, Discriminator の RTD 部分のハイパーパラメータも 3 節のものと同様として，提案手法固有のハイパーパラメータである Discriminator の MLM の損失関数の重みμと Discriminator 入力のマスク率を変更して検証を行った．

結果を表 2 に示す．

表 2　提案手法の精度

Twitter 評判分析精度		Discriminator 入力のマスク率			
		0.10	0.15	0.20	0.25
重みμ	0.5	60.94%	**63.17%**	60.71%	60.49%
	1	**62.72%**	61.83%	58.71%	60.04%
	2	60.27%	**62.50%**	61.38%	60.94%
	3	**63.62%**	61.61%	61.83%	**63.39%**

表 2 のとおり，いずれのパラメータにおいても Electra-Large の精度 69.19%を超えることができなかったことが分かる．また，重みμとマスク率と精度の関係ははっきりとした傾向は見られないが，強いて言えば，重みμは 3 が，マスク率は 0.15 や 0.10 といった小さい値の精度が若干よい．つまり，少量マスクしたデータを入力として Discriminator を学習して，MLM を重視した損失関数を使うとよい精度を達成できる可能性が少しあると言える．

提案手法（重みμを 1，マスク率を 0.15 としたもの）と，BERT, ELECTRA の事前学習の試行回数と転移学習の検証精度の関係を図 5 に示す．

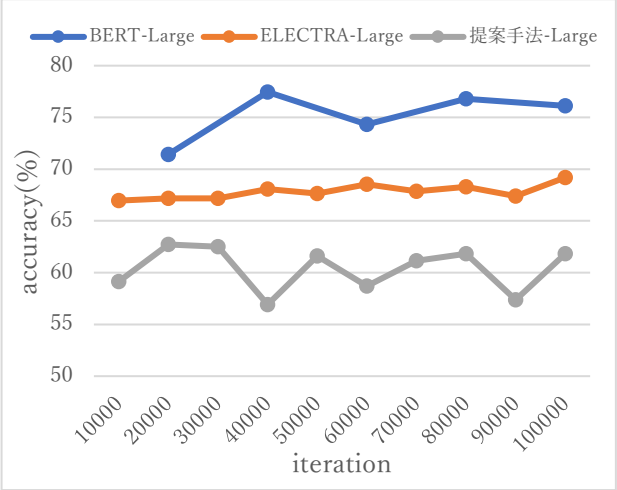

図 5　提案手法の事前学習の試行回数と検証精度の関係

図 5 のように，提案手法も 20,000 iteration で最高精度近くに到達しており，ELECTRA と同様に BERT よりも学習効率は高いと思われる．一方で，提案手法は BERT や ELECTRA に比べて精度のバラつきが大きく，試行回数を進めても精度が向上せず，かえって下がってしまうことがあることもわかる．

6. 考察

提案手法で精度を向上しなかった原因を考察した．

まず，提案手法（重みμが 1，マスク率が 0.15）の事前学習の損失関数と転移学習の精度の関係を図 6 に示す．

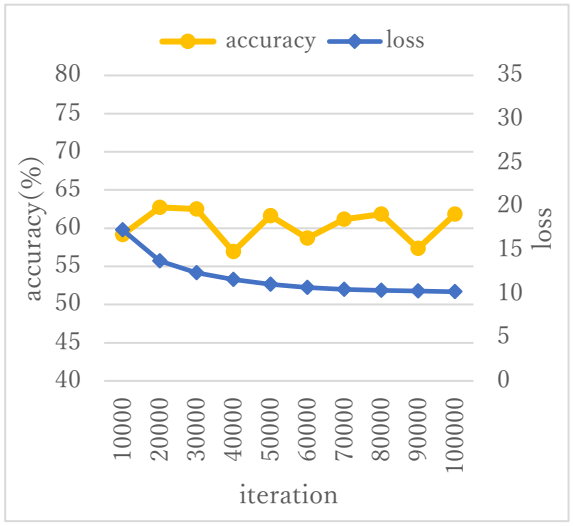

図 6　提案手法の事前学習の損失関数と検証精度の関係

図 6 より，損失関数（loss）は滑らかに低下しているのに対し，精度には大きな揺れがあることが分かる．このことから，提案手法の損失関数全体では検証精度が向上せず下がってしまうことがある原因はわからない．

そこで，検証精度と，Generator の損失関数（g loss）と Discriminator の RTD の損失関数（d loss），Discriminator の MLM の損失関数（d2 loss）の 3 種類の損失関数の変動の関係を詳細に確認した．その結果を示したものが表 3 である．ここで，acc diff は 1 つ上の行の iteration の検証精度（accuracy）との差分（つまり変動），g diff %，d diff %，d2 diff %はそれぞれ 1 つ上の行の iteration の g loss, d loss $\times\lambda$, d2 loss $\times\mu$との差分の割合（つまり変動率）を示す．

表 3　検証精度と各損失関数の変動の関係

iteration	accuracy	acc diff	g loss	g diff %	d loss × λ	d diff %	d2 loss × μ	d2 diff %
10000	59.1518		6.33052		10.00954		0.9624071	
20000	62.7232	3.5714	5.36627	15%	7.525892	25%	0.8670025	10%
30000	62.5	-0.2232	4.97679	7%	6.602409	12%	0.8287973	4%
40000	56.9196	-5.5804	4.741827	5%	6.081474	8%	0.7987067	4%
50000	61.6071	4.6875	4.58593	3%	5.699658	6%	0.7820277	2%
60000	58.7054	-2.9017	4.486702	2%	5.477981	4%	0.7535393	4%
70000	61.1607	2.4553	4.419371	2%	5.319149	3%	0.7476106	1%
80000	61.8304	0.6697	4.389909	1%	5.247385	1%	0.7494839	0%
90000	57.3661	-4.4643	4.361089	1%	5.202833	1%	0.7444608	1%
100000	61.8304	4.4643	4.338639	1%	5.132121	1%	0.7419026	0%

表 3 より，Generator の損失関数（g loss）と Discriminator の RTD の損失関数（d loss）に関しては滑らかに低下しているのに対し，Discriminator の MLM の損失関数（d2 loss）は精度の低下幅が上下する傾向がみられる．また，50,000, 70,000, 100,000 iteration のように d2 loss があまり低下しない場合に精度（accuracy）が上がり，40,000, 60,000, 90,000 iteration のように d2 loss が他の loss と同程度かそれ以上低下する場合に精度が下がる傾向が読み取れる．

このことから，Discriminator の MLM の推定を学習するときに何らかの悪影響が起きていると思われる．

その悪影響の原因として，以下の 2 つが考えられる．

1 つ目は，Discriminator の RTD を学習する際に，一部分がマスクされてしているため，転移学習では存在しないマスクがある前提で学習してしまうこと，およびヒントとなる周辺のトークンが減ってしまうことによって，精度の向上しないことが考えられる．この例を図 7 に示す．

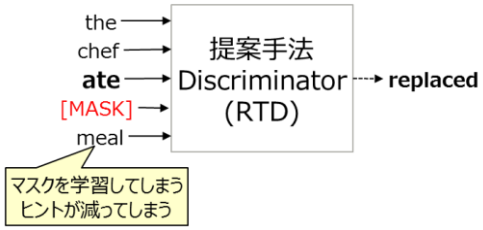

図 7　提案手法の悪影響の原因（その 1）の例

2 つ目は，Discriminator の MLM を学習する際に，一部分が Generator で置き換えられているため，ヒントとなるマスク箇所周辺の誤ったトークンを学習してしまうことによって，精度の向上しないことが考えられる．この例を図 8 に示す．

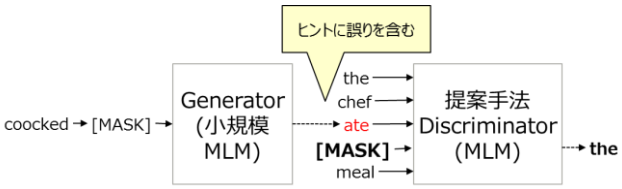

図 8　提案手法の悪影響の原因（その 2）の例

つまり，提案手法では RTD と MLM において転移学習の精度向上に結びつかない，無駄な学習をしてしまっている可能性があると言える．

このように，Discriminator に MLM の学習を単純に組み合わせる提案手法では，RTD 型学習の汎化性を上げる利点よりも，副作用のほうが大きかったと考えている．

これらの問題に対して，RTD ではマスクしないデータを，MLM では Generator で置き換える前のデータを使えば解消すると思われるが，学習効率を保つためには現在の Discriminator の構造を保ったまま RTD と MLM の入力を区別するような工夫が必要となり，簡単ではない．

7. おわりに

ELECTRA で採用している RTD 型の事前学習モデルの精度が頭打ちになっている課題に対して，Discriminator に MLM の学習を組み合わせることによって RTD の高い学習効率を保ったまま Twitter 評判分析での精度向上を試みた．しかし，精度向上を実現することはできなかった．

結果について考察したところ，Discriminator の MLM 学習の進行と精度低下の相関がみられ，MLM を組み合わせたことによる 2 種類の副作用が結果に影響しているのではないかと推測した．結果として，学習効率が高いが単純な問題である RTD に複雑な問題を混ぜることで汎化性を向上できるのではないか，という仮説を立証することはできなかった．

しかし，本稿ではまだ Twitter 評判分析の一部という小規模かつただ 1 種類のタスクで検証したに過ぎない．今後は他の日本語タスクや GLUE などの英語ベンチマークテストにおいてどういった挙動を示すかの確認や，精度低下の原因究明や対策の検討を行っていく予定である．それによって，本稿の仮説の立証を行う予定である．

参考文献

[1] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. NAACL-HLT, 2019.
[2] K. Clark, M.-T. Luong, Q. V. Le and C. D. Manning. ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators. ICLR, 2020.
[3] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. RoBERTa: A robustly optimized BERT pretraining approach. arXiv preprint arXiv:1907.11692, 2019.
[4] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, and Q. V Le. XLNet: Generalized autoregressive pretraining for language understanding. NeurIPS, 2019.
[5] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. R Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. ICLR, 2019.
[6] M. S. M. Patwary, R. Puri, P. LeGresley, J. Casper, and B. Catanzaro, Megatron-LM: Training Multi-Billion Parameter Language Models Using Model Parallelism. arXiv preprint arXiv: 1909.08053, 2019.
[7] 日本語 Wikipedia データベース スナップショット. (閲覧日：2021 年 8 月 30 日) https://dumps.wikimedia.org/jawiki/
[8] 青空文庫データセット. (閲覧日：2021 年 8 月 30 日) https://github.com/aozorabunko/aozorabunko
[9] OSCAR. Open super-large crawled aggregated corpus. (閲覧日：2021 年 8 月 30 日) https://oscar-corpus.com/
[10] T. Kudo and J. Richardson. SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing. arXiv preprint arXiv:1808.06226, 2018.
[11] Y. Suzuki. Filtering Method for Twitter Streaming Data Using Humanin-the-Loop Machine Learning, Journal of Information Processing (JIP), vol. 27, 2019, pp. 404–410.