

日本語大規模ブログコーパス YACIS に基づいた ELECTRA 事前学習済み言語モデルの作成及び性能評価

柴田 祥伍¹, プタシンスキ ミハウ¹, エロネン ユーソ¹, ノヴァコフスキ カロル², 梶井 文人¹

Shogo Shibata¹, Michal Ptaszynski¹, Juuso Eronen¹, Karol Nowakowski², Fumito Masui¹

¹ 北見工業大学 テキスト情報処理研究室

Kitami Institute of Technology, Text Information Processing Laboratory

² 東北公益文科大学 Tohoku University of Community Service and Science

f1812201003@std.kitami-it.ac.jp {michal,f-masui}@mail.kitami-it.ac.jp

karol@koeki-u.ac.jp eronen.juuso@gmail.com

概要

本研究では現在最大規模の日本語ブログコーパス YACIS を用いて、事前学習済み言語モデル ELECTRA を作成し、その性能評価について報告する。これまでの日本語向けの事前学習済み言語モデルは殆んど Wikipedia という記述文のみを含む言語資源を基に作成されてきたが、本研究においては口語的な文章の処理に焦点を当て事前学習にブログコーパスを用いる。評価実験では、一般的な自然言語理解の課題の他に口語の処理が必要となるネットいじめ検出の課題にも応用し、YACIS-ELECTRA モデルの応用性について考察する。

1 はじめに

近年,ELMo[1], BERT[2], ELECTRA[3], RoBERTa[4] 等,事前学習済み言語モデル,そして特にTransformer型ニューラルネットワーク [5] のアーキテクチャに基づいた大規模の事前学習済み言語モデルが、様々な研究課題や実世界問題の解決に応用され実用性が実証されている。例えば、BERT モデルを応用することによって FAQ の適切な質問回答のペアを取得するため QA の関連性の計算が性能向上したことが確認されている [6]。また、BERT の応用によって日本語の構文解析の精度が向上されたことが確認されており [7] さらに、執筆期間が数十年離れた小説の同著者の判別にも応用されていることや [8], 知識ベースの自動拡張への成果も挙げられている [9]。

事前学習済み言語モデルの有用性が複数の課題で確認されている一方、日本語用のモデルはまだ少数にあり、多様性に欠けている。数多くの事前学習済み言語モデルを共有しているライブラリである HuggingFace¹⁾ 内で公開されている英語用モデルの数は執筆時点現在 5 千個近くある一方、日本語用のモデルは 182 個のみと少なく、音声や画像処理以外のテキストベースのモデルのほとんどが同様のテキストデータをもとに作成されており、日本語における事前学習済み言語モデル作成における多様性が沈滞していることが課題となっている。

具体的には、多くのモデルは、異なるアーキテクチャ (BERT, RoBERTa, ELECTRA, GPT-2 等) で作成

1) <https://huggingface.co/>

されているが、事前学習に用いられたデータにはほとんど Wikipedia とニュース記事と CommonCrawl コーパスのみのテキストデータをもとに作成されている。そのようなデータは、入手しやすい一方、多様性がないためモデルの性能へ影響が及ぼされている可能性がある。例えば、Wikipedia とニュース記事は日常言語の使用から離れたデータであり、すなわち記述文や説明文のみからなるデータが用いられるため、日常日本語を取り扱う課題に向かない可能性がある。

そのため、これらのモデルは典型的な名詞や動詞の間の係わりをつかむことは可能であるが、口語やスラング、砕けた言い方などのような日常的な言語を分析する場合は不向きである可能性が考えられる。これに起因して、モデル間の性能が似通っており明確に優れているモデルは提案されていない。さらに、実世界の様々な課題に対して専門性・多様性のないモデルのみが提案されており、自然言語処理分野における様々な課題解決や全般的な発展を妨げている。

そこで、それらの課題を解決するために本研究では、日本語の日常的表現を豊富に含む最大規模の日本語のブログコーパスである YACIS コーパス [10, 11] を元に新たに事前学習済み言語モデルを作成し、その性能評価を行う。また、性能評価では、一般的な自然言語推論の課題の他に、日常言語やスラングの適切な処理が必要となるネットいじめ検出の課題も用いる。

2 YACIS 大規模日本語 blog コーパス

事前学習用のデータとしては、YACIS コーパスを用いた [10, 11]。YACIS コーパスとは、56 億語を含む現在最大規模の日本語ブログコーパスである。2010 年に収集した YACIS コーパスは、アメバブログ (略: アメブロ)²⁾ から収集され、作成した際アメブロの 1/3 を含んでいた。さらに、YACIS コーパスは多面的アノテーションされており [12, 13], 数多くの研究に応用されてきている。YACIS コーパスの元データ及びアノテーションは現在以下のページ³⁾にて 2022.4 以降に公開予定である。

2) <https://ameblo.jp>

3) <https://github.com/ptaszynski/yacis-corpus>

事前学習用のデータを選択する際は、データのサイズと言語の多様性という2点に注目を置いた。まず、コーパスのサイズとしては、現在 Wikipedia 全体（全言語版）のサイズは、290 億語を含み⁴⁾、その中日本語のみの記事は、25 億語を含む⁵⁾。YACIS は日本語 Wikipedia 全体より 2 倍以上大きいためそのまま事前学習に用いても十分と考えられる。さらに、言語の多様性に関しては、ほぼ説明文と記述文のみを含む Wikipedia と異なり、一般ユーザが毎日のように意見を表すブログ記事からなる YACIS は、日常的表現を豊富に含むため、事前学習に用いた場合、日常的な表現やスラングなどの処理に強みがあることが考えられる。

具体的には YACIS には感情文:非感情文の比率は 2:1 である [11] ため、感情に満ちた日常表現が豊富に含まれる。さらに Twitter にあるような文字制限のない言語資源（アメブロ）を基に収集されているためユーザが無理矢理に文字制限に合わせる中でメッセージが不自然かつ挑発的 [14] になる傾向はなく、思い通りに表現するため Twitter のようなマイクロブログよりも自然な日本語が含まれると考えられる。

Wikipedia に対して YACIS を使う可能な不利点としては、前者は数多くの概念を含むため、自然言語理解及び推論の課題に向く可能性があるが、本研究ではそのことを実験的に確認する。

3 事前学習方法について

3.1 ELECTRA アーキテクチャについて

事前学習には ELECTRA のアーキテクチャを使用する。ELECTRA [3] は、2018 年に提案された BERT [2] の事前学習アーキテクチャに基づき、事前学習プロセスを最適化し改良した手法であり、従来の改善手法である RoBERTa [4] の 1/4 の計算量で、近い性能を出すことが可能である [3]。

BERT をはじめこれまでの事前学習モデルでは、学習データにおけるトークンの一部が [MASK] という特別なトークン（隠しラベル）に置き換えられ、元のトークンは何のトークンであったかを予測させることによってモデルの事前学習を行う。しかし、[MASK] に置き換えられるのはトークンの一部（一般的には 15%）だけであり、モデルはその一部のトークンからしか学習できないため、学習プロセスの効率が低い。それと違って ELECTRA では、BERT アーキテクチャに基づく小規模の生成モデル（Generator）と識別モデル（Discriminator）という二つのモデルが利用される。具体的には、入力文のトークンの一部を隠し、生成モデルが予測したトークンに置き換える。それから、各トークンに対して元のデータにあったか、それとも生成モデルによって生成されたか、Discriminator に識別させることで事前学習を行う。[MASK] に置き換えられた部分だけでなく、すべてのトークンに対して予測が行われるため、より効率的な学習が可能になる。その上、事前学習にかかる時間を短くしても、複数のデータセッ

4) https://en.wikipedia.org/wiki/Wikipedia:Size_comparisons
5) <http://ja.wikicount.net>

表 1 事前学習モデルのサイズ。

Model	Layers	Hidden Size
Small	12	256
Base	12	768
Large	24	1024

トにて性能評価を行うための GLUE ベンチマーク [15] (言語モデルにおける自然言語理解能力を評価するためのデータセット) を用いた評価によると性能が低下せず、むしろ向上したことが明らかになった。

これらの理由から、本研究では事前学習モデルには、ELECTRA を使用することにした。

3.2 事前学習用の設定について

ELECTRA 事前学習済みモデルを訓練する場合、従来研究で提案されているモデルのサイズが 3 種類あり⁶⁾ (表 1 参照)、今回は日本語用の ELECTRA を公開した Cinnamon AI Blog にて掲載された先行研究⁷⁾ と比較するため、最初は Small で学習を行うことを決定した。まずは学習時間を考慮し Small で事前学習を行うことで、小規模のモデルを作成し、評価を実施した後、精度の向上が見込まれる Large モデルを用いた学習を予定している。

また、Small と Large の間となる Base サイズもあるが、Base を学習しないことにした。その理由は、YACIS ブログコーパス内の文は、500 単語（トークン）以上の文はノイズ扱いで削除され、256 単語より長い文も僅かである。上記を理由に、YACIS を使った場合、Small と Base の間は微量な精度向上しか期待できないため、Base モデルの学習は見送り、最終的に Small と Large のみを学習することにした。

3.3 事前学習モデル訓練の環境設定

初めに事前学習開始までの流れを把握するため、OpenWebTextCorpus⁸⁾ を用いて、ローカル環境 (CPU: Intel Core i9-7920X, RAM: 132GB, GPU: GeForce GTX1080Ti) で事前学習の試運転を行った。

ELECTRA とのバージョン違いによるエラーを回避するため、TensorFlow⁹⁾ の GPU 版バージョン 1.15 をダウンロードして利用した。実行環境は最新版の場合、同じくバージョンによるエラーが発生するため、Python のバージョンを 3.6 とし、学習用データ作成から事前学習の開始まで確認した¹⁰⁾。

学習用データ (YACIS コーパス) は 32GB あり、Small で事前学習を開始した場合約 7 日間かかるという想定が得られた。

3.4 語彙リスト作成

事前学習を行うためには、YACIS ブログコーパスから新たに語彙リストを作成する必要になった。

語彙リストの作成には韓国語版 ELECTRA モデ

6) <https://github.com/google-research/electra>
7) https://cinnamon.is/ideas/2020/06/22/20200619_research_001/
8) <https://skylion007.github.io/OpenWebTextCorpus/>
9) <https://www.tensorflow.org>
10) 環境準備にて起こった問題のまとめと解決方法については以下の GoogleDocs にて公開している <https://t.ly/Vtff>

ルを公開している先行研究¹¹⁾を参考に,Huggingface Tokenizer の機能を用いた語彙リスト作成用のプログラムを作成した。

先行研究⁷⁾との比較をより公平にするために語彙数を同じく 32,000 語とし,特殊トークン(事前学習済み言語モデルが文を見分けるために用いるセパレータなど)を含む語彙リストを作成した。

3.5 学習用データの準備

上記で作成した語彙リストを利用し,YACIS ブログコーパスから学習用データを作成した.作成には ELECTRA 本体にある学習用データ作成プログラム¹²⁾をハイパーパラメータをデフォルト⁶⁾のまま利用した.元データとなる YACIS ブログコーパスのテキストデータは,既に単語分割化及び文分割化などの前処理が行われているため,そのまま学習用データとして用いることができた。

3.6 事前学習プロセスについて

上記で作成した語彙リストと学習用データを用いて事前学習を開始した。

先行研究⁷⁾と比較するためモデルサイズを Small,埋め込み層を 128,語彙数を 32,000 に設定した.その他のハイパーパラメータについてはデフォルトのままである.¹³⁾また事前学習にはローカル環境で単一 GPU を使用した。

学習用データが 32GB あるため,試運転と同様に一週間強かかり,正解値とモデルの予測値のズレの大きさ(英語: "Loss" 「損失」)を算出する関数である損失関数(Loss function)のスコアのログを取りながら経過を観察した.損失関数は,

$$\min_{\theta_G, \theta_D} \sum_{x \in X} L_{MLM}(x, \theta_G) + \lambda L_{Disc}(x, \theta_D) \quad (1)$$

で表され, x は入力される単語を表し, L_{MLM} が Masked LM による [G]enerator 側の学習, L_{Disc} がトークンの真偽判定による [D]iscriminator 側の学習, λ はハイパーパラメータとなっている.この関数の結果が Loss 関数スコアとして表示され,0 に近づくほど良い結果を表す。

7日と6時間かけて,単一 GPU による Small サイズの事前学習が終了した.100万ステップ学習し,事前学習の目的関数(loss関数)は数日間10と9を前後しながら徐々に下がり,9と8を前後するようになり,最終的に学習が終了した時点で8.7319となった.Loss関数値の動きを1000ステップごとにプロットし可視化したグラフを図1に表した.事前学習済みの YACIS-ELECTRA モデルは HuggingFace にて公開されており¹⁴⁾,今後の研究へ利用可能である。

11) https://github.com/monologg/KoELECTRA/blob/master/docs/wordpiece_vocab_EN.md

12) https://github.com/google-research/electra/blob/master/build_pretraining_dataset.py

13) https://github.com/google-research/electra/blob/master/configure_pretraining.py

14) <https://huggingface.co/ptaszynski/yacis-electra-small-japanese>

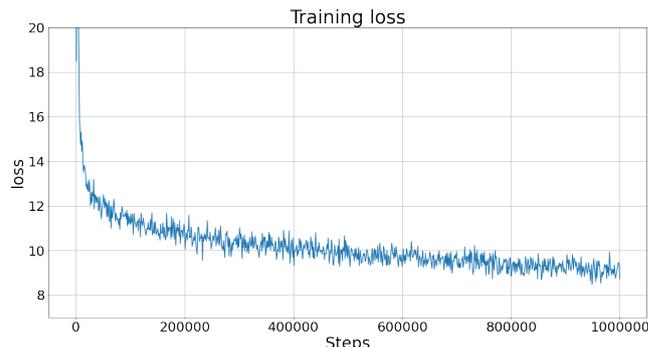


図1 YACIS-ELECTRA-Small 事前学習中の損失関数変動

4 評価実験

事前学習が終了した後,完成した事前学習済み言語モデルの性能評価を行なった.性能評価には2種類のデータセットを用いて,それぞれのデータセットの訓練セット部分をモデルの微調整に使い,テストセットを評価に使用し,2段階(微調整→評価)で評価を行なった。

4.1 評価用データセット

JSNLI データセット

まずは,モデルの総合的な自然言語推論能力を測るために,JSNLI データセット¹⁵⁾[16]を用いた.JSNLI データセットとは,英語の SNLI[17] データセットを日本語に翻訳したデータセットであり,標準として事前学習済み言語モデルの性能評価に用いられる。

データセットには,533,005 文ペアを含む訓練セットと,3,916 文ペアのテストセットが含まれる.データセットが表す課題では,前提を表す文,仮説を表す文とその間の関連性を表すラベルがあり,モデルの評価には前提文・仮説文の関連性を当てるタスクが与えられ,モデルの性能は正確率(Accuracy)によって算出される。

ネットいじめ検出データセット

総合的な自然言語推論能力の他,より要求性のある口語の処理への YACIS-ELECTRA の有用性を図るために,スラングや口語が豊富に含まれるネットいじめ検出用のデータセットを用いた.このデータセットは,(1)松葉[18]が学校非公式サイトから収集し,これまでに数多くのネットいじめ検出の研究[19]に使われたデータセットと,(2)荒田[20]が新しく Twitter から収集した有害ツイートデータセットを統合して作成した.データセット内の有害・非有害の事例はバランスをとった比率であり,訓練データには,有害1,650個,非有害1,624個,テストデータには,有害411個,非有害409個ある。

4.2 実験設定

上記の2つのデータセットを用いて,本研究で事前学習を行った日本語 ELECTRA モデルの評価を行った.そこで,総合的な自然言語推論能力を JSNLI データセットで評価した.また,事前学習

15) [https://nlp.ist.i.kyoto-u.ac.jp/index.php?日本語SNLI\(JSNLI\)データセット](https://nlp.ist.i.kyoto-u.ac.jp/index.php?日本語SNLI(JSNLI)データセット)

表2 JSNLI 課題の結果.

モデル種類	モデルサイズ	モデル名	正確率 (%)	損失値
BERT	Base	L-12_H-768_A-12_E-30_BPE	70.52	0.71
	Base	L-12_H-768_A-12_E-30_BPE_WWM	70.19	0.71
	Large	L-24_H-1024_A-16_E-30_BPE_WWM	70.27	0.71
ELECTRA	Small	YACIS-ELECTRA-Small	88.43	0.32

表3 ネットいじめ検出課題の結果.

モデル種類	モデルサイズ	モデル名	Acc	Prec	Rec	F1	F1 非有害	F1_ma cro_avg
BERT	Large Base	cl-tohoku-bert-large-japanese	0.899	0.898	0.900	0.899	0.899	0.899
		cl-tohoku-bert-base-japanese-v2	0.877	0.832	0.914	0.871	0.882	0.877
ELECTRA	Base	megagonlabs-transformers-ud-japanese-electra-base-discriminator	0.915	0.888	0.938	0.913	0.917	0.915
	Base	izumi-lab-electra-base-japanese-discriminator	0.830	0.779	0.870	0.822	0.839	0.830
	Small	izumi-lab-electra-small-japanese-discriminator	0.800	0.749	0.835	0.790	0.809	0.800
	Small	Cinnamon-electra-small-japanese-discriminator	0.820	0.830	0.814	0.822	0.817	0.819
	Small	YACIS-ELECTRA-Small (worst)	0.796	0.747	0.830	0.786	0.806	0.796
	Small	YACIS-ELECTRA-Small (best)	0.846	0.832	0.857	0.844	0.848	0.846

に用いるデータの種類がどこまでモデルの専門性を強化するかを評価するには、ブログ記事を基に作成した YACIS-ELECTRA と Wikipedia で学習した Cinnamon-ELECTRA を比較した。

また、それぞれのタスクを具体的に以下の通り実施した。JSNLI データセットを用いた実験では、YACIS-ELECTRA の他に、柴田らが提供した BERT 日本語 Pretrained モデル [7]¹⁶⁾を用いた。JSNLI データセットを使った評価では、元の ELECTRA⁶⁾と提供されている正式な評価スクリプトを用いるためには、モデルのチェックポイント（事前学習終了前の状態）のファイルが必要なため、比較実験には、柴田ら [7] が提供した日本語 BERT モデル 3 種類のみが使用可能であった。

ネットいじめ検出データセットを用いた実験では、二項分類を行うための平凡のスクリプトを作成し、複数のモデルを比較した。具体的には、HuggingFace ライブラリに登録されている日本語 ELECTRA モデル 4 種類と日本語 BERT モデル 2 種類と比較を行った。また、我々以外のモデルの場合は、数回実験を繰り返した後のベストのスコアのみを報告し、YACIS-ELECTRA はベストおよびワーストスコア両方報告する。

4.3 実験結果と考察

JSNLI 課題の結果を表 2 に表す。BERT ベースのモデルには正確率の大きな違いは見られなかった。モデル作成者は全単語マスクモデル (Whole Word Masking, WWM) の方が精度が高い傾向があると報告しているがそれを確認することはできなかった。さらに、関連研究 [16] では精度が 93%を超えていることを報告しているが、その報告結果を再現することもできなかった。YACIS-ELECTRA-Small は 88%を超え、比較した中で最高結果となった。

さらに、ネットいじめ検出課題の結果¹⁷⁾を表 3 に示す。結果に見られる傾向としては、同種類のモデ

ルの中では、Base モデルは Small モデルより結果が高かった。また、同じく同種類の Large モデルが Base より結果が高かった。また、同サイズのモデルで、ELECTRA が BERT より高い結果を取得できた。Small 級のモデルの中では、提案の YACIS-ELECTRA が一番高い結果になった。ただし、微調整における出発時点のランダムサンプル及びパラメータによって結果には 5%ほどの低下も見られ、微調整の設定が結果に大きな影響をすることが分かった。また、特に Small 級のモデルでは YACIS-ELECTRA との違いは事前学習時のデータセット (YACIS vs Wikipedia) の 1 つだけであるため、口語処理が必要とする課題における口語を含むデータセットの有用性を確認することができた。最高結果を取得したのは megagonlabs の ELECTRA-Base であり、事前学習には Wikipedia を使わず mc4 データセット¹⁸⁾のみを使い、アーキテクチャとしての ELECTRA の有用性を確認している。さらに megagonlabs 及び提案モデル以外は Wikipedia を使ったことから事前学習データとして Wikipedia は最適ではないことが示唆されている。

5 おわりに

本研究では、大規模日本語ブログコーパスを事前学習データとして用いて ELECTRA 事前学習済み言語モデルを作成した。モデル作成にブログを用いることで、口語の事前学習への有用性を確認でき、特に砕けた言い方やネットスラングの処理が必要とするタスクでの精度向上を確認できた。また、これまでは Small 級のモデルのみを作成、評価してきたが今後は Large 級も作成する予定である。

今後の課題としては、微調整時のランダムサンプルの結果への影響を調査する必要がある。また、事前学習時のデータ種類や訓練データ内に使われる言語がどのように応用タスクで得られる結果に影響をするか、つまり事前学習済み言語モデルの普遍性よりも専門性に関するさらなる研究も必要である。

16) https://nlp.ist.i.kyoto-u.ac.jp/?ku_bert_japanese

17) <https://huggingface.co/ptaszynski/yacis-electra-small-japanese-cyberbullying>

18) <https://huggingface.co/datasets/mc4>

謝辞

本研究にあたって色々と相談に乗っていただいた北海道大学のジェブカラファウ助教に感謝いたします。

参考文献

- [1] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In **Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)**, pp. 2227–2237, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.
- [2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: pre-training of deep bidirectional transformers for language understanding. **arXiv preprint arXiv:1810.04805**, 2018.
- [3] Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. Electra: Pre-training text encoders as discriminators rather than generators. **arXiv preprint arXiv:2003.10555**, 2020.
- [4] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. **arXiv preprint arXiv:1907.11692**, 2019.
- [5] 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.
- [6] Wataru Sakata, Tomohide Shibata, Ribeka Tanaka, and Sadao Kurohashi. Faq retrieval using query-question similarity and bert-based query-answer relevance. In **Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval**, pp. 1113–1116, 2019.
- [7] 柴田知秀, 河原大輔, 黒橋禎夫. Bert による日本語構文解析の精度向上. 言語処理学会第 25 回年次大会, pp. 205–208, 2019.
- [8] Jagna Nieuwazny, Karol Nowakowski, Michal Ptaszynski, and Fumito Masui. Can you fool ai by doing a 180? — a case study on authorship analysis of texts by arata osada. **Information Processing & Management**, Vol. 58, No. 5, p. 102644, 2021.
- [9] Rzepka Rafal, Takishita Sho, and Araki Kenji. Bacteria lingualis on bertoids – concept expansion for cognitive architectures. 人工知能学会第二種研究会資料, No. AGI-014, p. 10, 2020.
- [10] Jacek Maciejewski, Michal Ptaszynski, and Pawel Dybala. Developing a large-scale corpus for natural language processing and emotion processing research in japanese. In **Proceedings of the International Workshop on Modern Science and Technology (IWMST)**, pp. 192–195, 2010.
- [11] Michal Ptaszynski, Pawel Dybala, Rafal Rzepka, Kenji Araki, and Yoshio Momouchi. Yacis: A five-billion-word corpus of japanese blogs fully annotated with syntactic and affective information. In **Proc. of the AISB/IACAP world congress**, pp. 40–49, 2012.
- [12] Michal Ptaszynski, Rafal Rzepka, Kenji Araki, and Yoshio Momouchi. Automatically annotating a five-billion-word corpus of japanese blogs for sentiment and affect analysis. **Computer Speech & Language**, Vol. 28, No. 1, pp. 38–55, 2014.
- [13] Michal Ptaszynski, Yoshio Momouchi, Jacek Maciejewski, Pawel Dybala, Rafal Rzepka, and Kenji Araki. Annotating japanese blogs with syntactic and affective information. In **Mining User Generated Content**, pp. 229–262. Chapman and Hall/CRC, 2014.
- [14] Sarita Yardi and Danah Boyd. Dynamic debates: An analysis of group polarization over time on twitter. **Bulletin of science, technology & society**, Vol. 30, No. 5, pp. 316–327, 2010.
- [15] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. **arXiv preprint arXiv:1804.07461**, 2018.
- [16] 吉越卓見, 河原大輔, 黒橋禎夫ほか. 機械翻訳を用いた自然言語推論データセットの多言語化. 研究報告自然言語処理 (NL), Vol. 2020, No. 6, pp. 1–8, 2020.
- [17] Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. A large annotated corpus for learning natural language inference. **arXiv preprint arXiv:1508.05326**, 2015.
- [18] 松葉達明, 里見尚宏, 榊井文人, 河合敦夫, 井須尚紀. 学校非公式サイトにおける有害情報検出. 電子情報通信学会技術研究報告. NLC, 言語理解とコミュニケーション, Vol. 109, No. 142, pp. 93–98, 2009.
- [19] Michal E Ptaszynski and Fumito Masui. **Automatic Cyberbullying Detection: Emerging Research and Opportunities: Emerging Research and Opportunities**. IGI Global, 2018.
- [20] Masaki Arata. Study on change of detection accuracy over time in cyberbullying detection. Master’s thesis, Kitami Institute of Technology, Dept. of Computer Science, 2019.