テキストマイニングツールのログからの実験設定の説明文生成

森田 康介 * 西村 太一 * 亀甲 博貴 † 森 信介 †

概要

実験設定を適切に記述することは，科学技術論文において重要である．本研究では，テキストマイニングツールのログから実験設定の説明文を生成することを目的とする．人文科学分野において広く使用されている KH Coder を用いている論文を対象に収集し，論文中の実験設定の記述と実際のツールの実行ログを再現したもののペアからなるデータセットを構築した．また，このデータセットを用いて論文中の記述から実行ログを推定するモデルを構築し，アノテーションしていない論文に適用することにより自動的にデータセットを拡充した．これらを用いて，実験ログから説明文を生成するモデルを構築した．

1 はじめに

科学技術論文の実験において，その実験設定を正しく記述し残すことは重要である．なぜなら，実験設定を残すことによって論文を書いた本人だけでなく，論文を読んだ人間も行った実験の再現が容易になるからである．近年では，あらゆる分野で論文の再現性が問題となる事例が発生しており，ますますその重要性が高まっている．

テキストマイニングとは，文章を解析して情報を取り出す分析方法のことである．例えば，語の出現頻度や同時に出現（共起）しやすい単語などの情報から共起ネットワークや対応分析といった分析を施すことにより，文のテーマやトピックを探ることができる．現在，テキストマイニングツールが用いられている論文は社会学や文学をはじめとした幅広い分野で執筆されている．

本研究では，それらの論文の執筆に焦点を当てる．一般的に，ツールの利用にあたってはどのような処理が行われたかを示すログが生成される．テキストマイニングツールにおいてもログは生成される．テキストマイニングツールのログから行なった操作を説明する文を生成することができれば，論文の書き手に対する執筆の補助になり，読み手の実行内容の理解度と再現性が高まる．反対に，論文に記されている操作の説明文からテキストマイニングツールのログを生成することができれば，実行内容の再現がより容易になる．

元来は人間が作成した規則やテンプレートをもとにしたコンピュータによる文の自動生成が考えられてきた [1, 2]．しかし，自然な文を自動で生成するためには，膨大な規則の構築が必要となるという欠点がある．この問題を解決するために，近年では深層学習を利用したアプローチが注目されている．深層学習のアプローチが十分な性能を発揮するためには規模の大きいデータセットを用意しなければならず，その作成にはコストがかかるという問題がある．これを解決するために，データセットを自動で拡張する手法がとられてきた [3, 4]．

本研究ではテキストマイニングツールを利用することを想定し，仮想的なログを定義する．そして，固有表現認識とテキスト分類を用いて半自動的にログと説明文のペアからなるデータセットを構築する．構築したデータセットで T5 [5] を学習させ，ログから操作の説明文の生成と，その反対の操作である説明文からログの生成を行う．生成した説明文について，文の自動評価尺度を用いて評価を行う．

* 京都大学大学院情報学研究科
† 京都大学学術情報メディアセンター

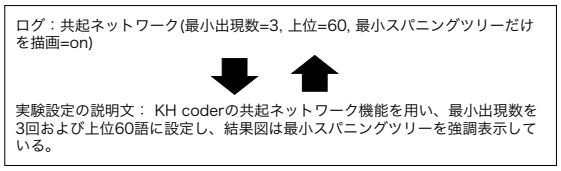

図 1　入出力の例

ここで，データセットの拡張の有無による評価結果の違いについて考察する．その結果，データセットを拡張することで，性能が向上したことが明らかになった．

2　関連研究

He らは実際のログを処理する一連の流れを説明し，各工程における手法を調査しまとめている [6]．ログの処理の工程の中の Log Parsing とは，ログを表形式に加工する処理のことである．

記号の組み合わせや表などデータとして記述されたものをテキストで説明することは Data-to-Text と呼ばれるタスクとして扱われている [7, 8]．近年の深層学習による手法として，Liu らは単語の情報だけでなく表における位置を組み込んだ Dual-attention のモジュールを考案した [9]．Puduppully らは入力の表のどの部分を出力するかをデコーダーに示すための Content Selection と Content Planning というモジュールを考案した [10]．これらの先行研究ではパイプライン式に RNN や Attention を使用したモジュールを組み合わせたエンコーダー・デコーダーモデルを提案している．一方で，Kale らは，そのような従来のパイプライン式のモデルだけでなく BERT や GPT-2 と比較しても，T5 [5] を用いたモデルは Data-to-Text のタスクにおいてより高いスコアを獲得できることを示した [11]．

深層学習のモデルが高いパフォーマンスを発揮するためには大きいサイズのデータセットが必要である．そのため，近年では Data-to-Text のタスクにおいても，データセットの拡張が行われてきた．Chang らはデータのスロットの値を置き換えることに加えて，言語モデルを使用し多様なテキストを出力することでデータセットを拡張した [3]．Qader らはデータの値しかないサンプルに対して一度テキストに変換した後，再びデータに変換し損失を計算することによりデータセットを拡張する半教師ありの手法を考案した [4]．

これらの結果を用いて，van der Lee らはデータ拡張と疑似ラベリングの 2 つの半教師あり学習の手法を 3 種の Data-to-Text のデータセットに適用することで拡張したのち，それぞれの組み合わせにおける自動評価・人手での評価での違いを T5 を用いて検証し定量的にまとめた [12]．

3　提案手法

3.1　データセットの手動作成

最初に，小規模なデータセットを人手で作成する．本研究では，テキストマイニングツールとして KH Coder [13] を用いる．まず，KH Coder を用いるにあたって，使用者が変更可能な設定を予めリストアップした．

次に，KH Coder を用いた論文のリスト[*1]から 2021 年の論文をダウンロードした．そして，それらの論文から KH Coder でテキストマイニングを行った際の操作の説明にあたる文を抽出した．例を**図 1** の実験設定の説明文に示した．共起ネットワークか対応分析を使用している論文がほとんどであったため，どちらかが行われている論文のみを対象にした．

その後，リストアップした設定をもとに説明文に対応するログを擬似的に人手で作成した．ログの先頭部分には分析方法を記述し，続く括弧の中には各設定項目に対する値が「(設定項目)=(値)」という形式でコンマによって区切って記述されるようにログの記述方法を定めた．ログの例を**図 1** に示した．ここで作成したデータセットの概要を**表 1** に記載する．

[*1] https://khcoder.net/bib.html

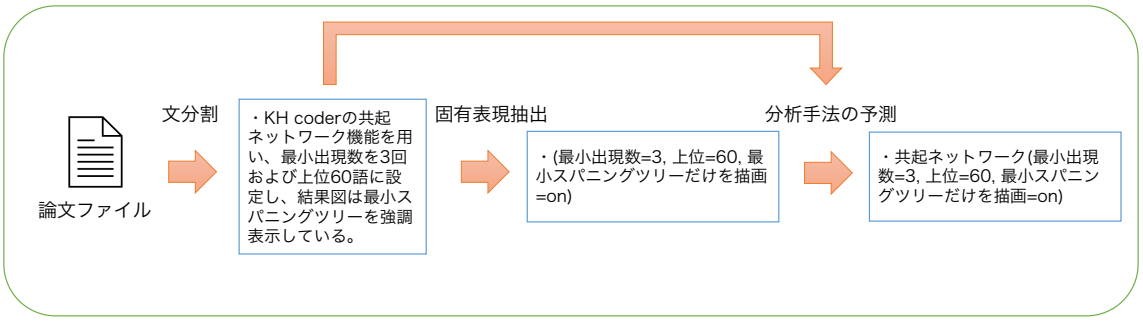

図 2　自動でデータセットを構築する一連の流れ

	共起ネットワーク	対応分析	合計
論文数	192	38	230
操作数	213	40	253

表 1　手動作成したデータセットの概要

3.2　データセットの自動作成

3.2.1　固有表現抽出を用いた設定部分の自動作成

次に，手動作成のデータセットをもとに，データセットの自動拡張を行う．擬似的に定義したログには実験設定の情報が含まれている。そのようなログのデータセットを自動で作成するために，どのような設定で分析を行ったのかを表す表現を説明文から抽出する固有表現認識（NER）を行う．まず，手動作成のデータセットの説明文を MeCab [14] を用いて単語分割した．そして，説明文に含まれる実験設定を固有表現とし，単語に IOB 形式でアノテーションを施した．**表 2** にアノテーションした固有表現の一覧を示す．

Flair [15] を用いて固有表現認識器を学習した．学習にあたっては，Flair 内で提供されている日本語の文字単位の事前学習済みの埋め込みベクトルを使用した．

このようにして得られた学習済みモデルを利用し，2020 年の論文 509 件に対し，以下の手順で固有表現認識を行なった．

1. pdfminer[*2]を用いて論文の pdf ファイルを読み込む．
2. 読み込んだ論文に対して spaCy[*3]の文分割機能を用いて自動で文単位に分割する．
3. 分割した文それぞれに対して NER のモデルを適用し，固有表現を予測する．これによって固有表現が予測された文と固有表現のペアを得ることができる．
4. 予測された固有表現をはじめに定義したログの形へとまとめ，整える．

この手順により，509 件の論文から 5,291 個の固有表現が抽出された．同じ文に対する固有表現をまとめると，4,087 対の文とログの括弧部分の組みが得られた．このようにして，自動で説明文とそれに対応したログの括弧部分が得られる．

Flair の固有表現の予測において，確信度を示すスコアも出力される．学習するデータの質による生成されるテキストの違いを検証するために，スコアをしきい値として 0 から 0.9 まで 0.1 刻みで値を変化させ，それより小さい確信度の固有表現を除外することで異なるデータセットを作成する．これは，スコアを確信度としてみなし，確信度が低いものはデータとして適切ではないという仮定に基づいて行った．それぞれのデータセットにおける全体の固有表現の数と文とログの組の数を**表 4** に示す．

3.2.2　BERT を用いた分析方法の予測

次に，ログの分析方法部分のアノテーションを自動化するために，文からその分析方法を予測するモ

[*2] https://pdfminersix.readthedocs.io/en/latest/

[*3] https://spacy.io/

タグの種類	タグの説明	出現数
MINAPP	語の最小出現数を表す	79
JAC	Jaccard 係数を使用したことを表す	77
TOP	共起関係の強い語の上位何件まで表示するかを表す	69
JACNUM	Jaccard 係数の最小値を示す	33
EXT	外部変数を表す	33
BOLD	共起関係の強さによって線の太さを変えることを表す	25
SUBG	サブグラフ検出の種類を表す	24
BUBBLE	バブルプロットを使用したことを表す	22
PART	分析対象の品詞を表す	11
UNIT	分析単位を表す	10
TREE	最小スパニングツリーのみを描画したことを表す	8
MINDOC	最小文書数を表す	8
DIF	差異が顕著な語を分析に使用することを表す	4
ORI	原点から離れた語のみラベル表示することを表す	3
MAXAPP	語の最大出現数を表す	3
STAN	係数の標準化を表す	3
COS	コサイン類似度を使用したことを表す	2
TYPE	共起関係の種類を表す	1
BUBSIZE	バブルプロットの大きさを表す	1
MAXDOC	最大文書数を表す	1
合計		417

表 2　共起ネットワーク・対応分析における固有表現

デルを作成する．固有表現抽出によって自動で作成したデータセットのうち，18 件の論文にあたる 205 件のペアに対してその分析方法を人手でアノテーションした．さらに，このデータセットに，はじめに手動で作成したデータセットのペア 253 件も加え，合計で 458 件のデータセットを作成した．

固有表現抽出では分析とは関係のない文も固有表現として予測され，取得される場合がある．そのため，共起ネットワーク，対応分析，それ以外の 3 クラスの分類を行う問題として学習させる．

モデルは transfromers の BERT [16] を用いた．事前学習済みモデルには東北大学乾研究室のモデル cl-tohoku/bert-large-japanese[*4]を用い，ファインチューニングを行った．データセットを学習データ: 検証データ: テストデータで 8:1:1 に分割した後，学習データで 20 エポック学習させ，そのうち検証データにおける損失が最も小さいモデルを最終的な学習済みモデルとした．この時のテストデータにおける accuracy は 0.9574 であった．その他の結果を**表 3** に示す．

	precision	recall	macro-f1	support
いずれでもない	0.909	1.000	0.952	20
共起ネットワーク	1.000	0.957	0.978	23
対応分析	1.000	0.750	0.857	4

表 3　BERT による分析手法の分類の結果

このモデルを自動で作成した固有表現抽出により作成したデータセットに適用した．この分類により，文が共起ネットワークか対応分析であると分類されたデータのログの先頭に分析方法の記述を追加したものを最終的なデータセットに用いる．そのデータ数を**表 4** の分類後のデータ数に示す．

3.3　テキストとログの生成

得られたデータセットを使用して，ログを入力としたテキスト生成とテキストを入力としたログ生成を行う．ここではモデルとして T5 [5] を用い

[*4] https://huggingface.co/cl-tohoku/bert-large-japanese

	$\theta = 0.0$	$\theta = 0.1$	$\theta = 0.2$	$\theta = 0.3$	$\theta = 0.4$	$\theta = 0.5$	$\theta = 0.6$	$\theta = 0.7$	$\theta = 0.8$	$\theta = 0.9$
固有表現の数	5291	5279	5249	5217	5152	5038	4833	4361	3263	1498
ペア数（文の数）	4087	4077	4058	4044	4016	3950	3819	3492	2665	1221
分類後のデータ数	718	718	718	717	714	708	691	657	557	374

表 4　自動作成したデータの概要

る．T5 とは”Text-to-Text Transfer Transformer”を略した名称である．入力と出力の両方をテキストに統一し，Transformer[17] をベースとして大規模な言語コーパスを用いて学習させる．その後，転移学習を行うことで質問応答や文書分類などのさまざまなタスクを解くことができるモデルである．Data-to-Text のタスクの中でも T5 の有用性は示されていることから [11]，今回はこれを用いることにした．

4　実験

4.1　T5 を用いたログからの説明文生成

作成したデータセットをもとに，ログからの説明文生成を行う．モデルには T5 を用いる．日本語の事前学習済みモデルとして，園部 勲氏により transformers に公開されている sonoisa/t5-base-japanese[*5]を使用した．この事前学習済みモデルを作成したデータセットでファインチューニングする．

本研究で人手で作成したデータセットのサイズは大きくない．そのため，以下の手続きに則って交差検定を実施した．まず，手動で作成したデータセットをランダムに 8:2 に訓練データおよびテストデータと検証データに分割した．その後，前者のデータを 9:1 に分割して交差検定を行った．この際，後者の検証データを用いて BLEU-4 の性能が最も高いモデルを評価に用いた．また，評価結果は交差検定の平均値を示している．

パラメータの更新には Adam [18] を用い，学習率は 3e-4，ϵ は 1e-8 とした．学習は 50 エポック行った．その他のハイパーパラメータの設定はデフォルトのものを用いた．評価指標には，BLEU [19] と METEOR [20] を用いる．交差検定におけるテストセットでのスコアの平均値を計測した．

比較のため，単純な規則でログから説明文を生成するシステムを作成した．具体的なアルゴリズムとしては，ログの形式が「分析方法 (設定項目=値, 設定項目=値...)」であることを用いて「(設定項目) を (値),」という文字列を繰り返し，最後の設定項目まで達した場合「とし (分析方法) を行った」と出力する．例えば，「共起ネットワーク (最小出現数=3, 上位=60, 最小スパニングツリーだけを描画=on)」というログが入力として与えられた場合，「最小出現数を 3，上位を 60，最小スパニングツリーだけを描画を on とし共起ネットワーク分析を行った.」となる．

表 5 に結果を示す．これらの結果から，BLEU-1 に関しては手動で作成したデータセットのままのスコアが高いことがわかる．一方で，BLEU-2, BLEU-3，BLEU-4，METEOR に関しては固有表現抽出のしきい値 θ を 0.3 にして自動作成したデータを追加した場合が最もスコアが高かった．また，テンプレートと比較するとどのしきい値であっても BLEU では大きい差ができていることがわかる．

特筆すべき点として，どのしきい値の場合でも何もデータを加えない場合と比較して METEOR スコアが高くなっている．

また, BLEU-2 と BLEU-3, BLEU-4, METEOR のスコアが固有表現抽出でのしきい値が 0.3 の場合に最も高いという結果からデータの量に加えて擬似データに対してもある程度以上の質が求められていると考えられる．しかし，しきい値が 0.0 から 0.5 の時に見られるように，ほとんど同じデータセットの大きさであるにもかかわらず，しきい値が上がる

[*5] https://huggingface.co/sonoisa/t5-base-japanese

	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR
$\theta = 0.0$	21.10	13.70	9.31	6.70	23.13
$\theta = 0.1$	18.70	12.70	9.16	6.94	24.40
$\theta = 0.2$	18.57	12.77	9.38	7.21	22.52
$\theta = 0.3$	21.23	**14.69**	**10.74**	**8.18**	**25.73**
$\theta = 0.4$	21.11	14.20	10.26	7.84	24.71
$\theta = 0.5$	20.74	13.23	8.97	6.45	22.07
$\theta = 0.6$	20.77	14.06	10.13	7.75	25.60
$\theta = 0.7$	17.97	12.08	8.45	6.24	23.65
$\theta = 0.8$	21.03	14.10	10.20	7.78	24.86
$\theta = 0.9$	21.73	14.49	10.14	7.47	23.45
オリジナル	**22.01**	13.60	8.98	6.16	21.27
ルールベース	10.00	6.10	3.91	2.75	20.49

表 5　ログから生成した説明文の評価

と必ずしもスコアが高くなるという結果が得られたわけではない．この原因については，固有表現抽出を行なっているのも深層学習のモデルであることから，しきい値が絶対的なものではないということが考えられる．

4.2　T5 を用いた説明文からのログ生成

次に，テキストからログを生成する．4.1 項の実験から入力と出力を入れ替えた学習を行う．ここでは，検証データにおける損失関数の値が最も小さいモデルを採用した．その他の設定については前項と同じにしたまま実験を行った．ログには分析の方法と設定を表す部分が存在するが，これを分けて評価を行う．

分析方法については，ログの先頭部分に記述するように定義している．生成されたログからこの部分を抜き取り，正解データに対する分類精度を一般的な分類タスクと同様に計算した．ここでは F1 の値を計算し，f1-method として表に掲載した．

分析の設定については，ログの先頭部分の後に続く括弧の中に記述するように定義している．括弧の中には各設定項目に対する値が「(設定項目)=(値)」という形式で区切り文字コンマにより記述されている．これら設定項目と値の組みの 1 ブロックを予測とみなし，スコアを計算する．具体的には，予測されたブロックのうち正解と設定項目と値が完全に一致している割合を precision とし，正解のうち設定項目と値が予測と完全に一致している割合を recall として計算した．例えば，生成されたログの設定項目を「(最小出現数=3, 上位=60)」，正解のログの設定項目を「(最小出現数=3, 上位=60, 最小スパニングツリーだけを描画=on)」とすると，precision は 2/2 となり，recall は 2/3 となる．precision と recall の調和平均を f1-options とし表に掲載した．

結果を**表 6** に掲載した．これらの結果から，全体的な傾向としてはログを生成する場合においても今回のデータ拡張の手法によってスコアが著しく低下することはないことがわかった．f1-method においては $\theta = 0.0$ の場合に最もスコアが高く，そのほかの場合ではオリジナルのデータセットのスコアを超えることはなかった．さらに，説明文の生成の場合と同様にしきい値が高くなるにつれてスコアが高くなる傾向は確認できなかった．

一方で，f1-options においてはすべてのしきい値でオリジナルのデータセットよりもスコアが高くなった．しかし，こちらもしきい値とスコアの関連性は確認できなかった．

なお，生成されたログの形式が定義したものと合致しない場合があった．ここでは「共起ネットワーク」「対応分析」のどちらかの文字列で始まり，それに続いて「(」で始まり「)」で終わるものでなかったものをエラーとして計測する．**表 6** にその数を記載する．なお，f1 の値はそのようなデータは除外

	$\theta = 0.0$	$\theta = 0.1$	$\theta = 0.2$	$\theta = 0.3$	$\theta = 0.4$	$\theta = 0.5$	$\theta = 0.6$	$\theta = 0.7$	$\theta = 0.8$	$\theta = 0.9$	オリジナル
f1-method	**0.989**	0.959	0.970	0.953	0.964	0.953	0.964	0.953	0.959	0.958	0.982
f1-options	0.750	0.735	0.757	0.760	0.759	**0.761**	0.758	0.735	0.745	0.744	0.690
エラー数	0	1	2	2	1	0	1	1	1	1	26

表 6　説明文から生成したログの評価

して計算した．エラーの割合はしきい値につき平均して全テストデータ 202 件中 3.2 件であった．しかし，すべてのエラーのうち約 74% である 26 件はオリジナルのデータセットのみを訓練データとした場合であった．この結果からデータ数の増加により正しい形式の学習がより可能になることがわかる．

5　おわりに

本稿では，テキストマイニングの際に出力されるログを仮想的に定義し，そのログからの説明文生成とその反対の操作である説明文からのログの生成を行なった．これを深層学習的なアプローチで行うにあたって少量のデータセットを補完するための，データの構造性を利用した固有表現認識による半自動のアノテーション手法も提案した．半自動で構築したデータセットを用いた実験において，拡張前のデータセットとの自動評価尺度による比較も行ない，その影響を調査した．その結果，今回用いたほぼ全ての指標で最もスコアが高くなるのはデータセットを拡張した場合であることが確認できた．特に生成した説明文の BLEU-3，BLEU-4，生成したログのオプションで今回の手法の有用性が確認できた．また，ログの生成において，オリジナルのデータセットだけでは出力すべき形式を十分に学習できていなかったことからもデータ拡張の有用性が確認できた．

参考文献

[1] Eli Goldberg, Norbert Driedger, and Richard I Kittredge. Using natural-language processing to produce weather forecasts. *IEEE Expert*, Vol. 9, No. 2, pp. 45–53, 1994.

[2] Kees Van Deemter, Mariët Theune, and Emiel Krahmer. Real versus template-based natural language generation: A false opposition? *Computational linguistics*, Vol. 31, No. 1, pp. 15–24, 2005.

[3] Ernie Chang, Xiaoyu Shen, Dawei Zhu, Vera Demberg, and Hui Su. Neural data-to-text generation with lm-based text augmentation. *arXiv preprint arXiv:2102.03556*, 2021.

[4] Raheel Qader, François Portet, and Cyril Labbé. Semi-supervised neural text generation by joint learning of natural language generation and natural language understanding models. *arXiv preprint arXiv:1910.03484*, 2019.

[5] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, Vol. 21, No. 140, pp. 1–67, 2020.

[6] Shilin He, Pinjia He, Zhuangbin Chen, Tianyi Yang, Yuxin Su, and Michael R Lyu. A survey on automated log analysis for reliability engineering. *ACM Computing Surveys (CSUR)*, Vol. 54, No. 6, pp. 1–37, 2021.

[7] Ernie Chang, Jeriah Caplinger, Alex Marin, Xiaoyu Shen, and Vera Demberg. Dart: A lightweight quality-suggestive data-to-text annotation tool. In *Proceedings of the 28th International Conference on Computational Linguistics: System Demonstrations*, pp. 12–17, 2020.

[8] Thiago Castro Ferreira, Chris van der Lee, Emiel Van Miltenburg, and Emiel Krahmer.

Neural data-to-text generation: A comparison between pipeline and end-to-end architectures. *arXiv preprint arXiv:1908.09022*, 2019.

[9] Tianyu Liu, Kexiang Wang, Lei Sha, Baobao Chang, and Zhifang Sui. Table-to-text generation by structure-aware seq2seq learning. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[10] Ratish Puduppully, Li Dong, and Mirella Lapata. Data-to-text generation with content selection and planning. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 33, pp. 6908–6915, 2019.

[11] Mihir Kale and Abhinav Rastogi. Text-to-text pre-training for data-to-text tasks. *arXiv preprint arXiv:2005.10433*, 2020.

[12] Chris van der Lee, Thiago Castro Ferreira, Chris Emmery, Travis Wiltshire, and Emiel Krahmer. Neural data-to-text generation based on small datasets: Comparing the added value of two semi-supervised learning approaches on top of a large language model. *arXiv preprint arXiv:2207.06839*, 2022.

[13] 樋口耕一. テキスト型データの計量的分析—2つのアプローチの峻別と統合—. 理論と方法, Vol. 19, No. 1, pp. 101–115, 2004.

[14] Taku Kudo. Mecab: Yet another part-of-speech and morphological analyzer. `http://mecab.sourceforge.net/`, 2005.

[15] Alan Akbik, Tanja Bergmann, Duncan Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf. FLAIR: An easy-to-use framework for state-of-the-art NLP. In *NAACL 2019, 2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pp. 54–59, 2019.

[16] 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.

[17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, Vol. 30, , 2017.

[18] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[19] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pp. 311–318, 2002.

[20] Satanjeev Banerjee and Alon Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pp. 65–72, 2005.