言い換えラティスを用いたテキスト生成の性能改善

西原 大貴
大阪大学大学院情報科学研究科
nishihara.daiki@ist.osaka-u.ac.jp

梶原 智之
愛媛大学大学院理工学研究科
kajiwara@cs.ehime-u.ac.jp

荒瀬 由紀
大阪大学大学院情報科学研究科
arase@ist.osaka-u.ac.jp

藤田 篤
情報通信研究機構
atsushi.fujita@nict.go.jp

1 はじめに

言語構造を機械学習に導入するために，機械翻訳 [1, 2] や文書要約 [3, 4]，対話における感情認識 [5]，抽象的意味表現 [6–10] など多くの自然言語処理タスクにおいてグラフに基づく手法が提案されている．例えば，有向非巡回グラフの一種であるラティスは，テキストを単純な単語の系列ではなく単語間の構造情報を持たせて表現でき，テキストの分類 [11, 12] や生成 [13, 14] に広く活用されている．

テキスト生成タスクでは，単語分割の曖昧性 [13, 14] や音声認識の曖昧性 [15–19] をラティスとして表現する Lattice2Seq モデルが，最も尤度の高い 1 つの系列を用いる Seq2Seq モデルよりも高い性能を達成することが知られている．本研究では，新たに語彙選択の曖昧性に着目し，入力テキストとその言い換えを用いてラティスを構築する．つまり，人間が書いたテキストの中の個々の表現は，必ずしも「それでなくてはならない」表現とは限らないという仮説に基づき，入力テキストの語句の言い換えを考慮し，入力テキストの情報を増やすことによってテキスト生成の性能改善を試みる．英日翻訳および英語のスタイル変換における実験の結果，この手法による BLEU [20] 値の顕著な改善を確認した．

2 提案手法

図 1 に示すように，本研究では言い換え辞書を用いて入力テキストを言い換えラティスに変換する．各ノードが 1 つの単語を表すラティスを構築し，入力テキストとその複数の言い換えをコンパクトに表現する．エッジは接続する 2 つのノード（単語）の繋がりを表しており，エッジの重みは言い換えとその前後の単語の繋がりの良さを表現する．このように，ラティス構造を用いることで複数の言い換えを表現でき，訓練時と評価時の両方で入力テキストの情報を増やすことが可能となる．

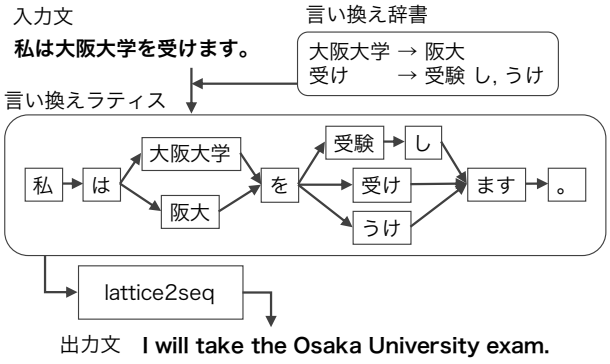

図 1 提案手法の全体図

2.1 言い換えラティスの構築

入力の単語列 $s_1\ s_2\ \cdots\ s_N$ （N は単語数）に対し，次の手順でラティス $G = (V, E)$ を構築する．

1. **ラティスの初期化** Source ノードを文頭記号 s_0，Sink ノードを文末記号 s_{N+1} とし，$s_0 \rightarrow s_1 \rightarrow \cdots \rightarrow s_N \rightarrow s_{N+1}$ をパスとするグラフを作る．すなわち，$V = \bigcup_{i=0}^{N+1}\{s_i\}$，$E = \bigcup_{i=0}^{N}\{(s_i, s_{i+1})\}$ とする．エッジ (a, b) があるとき，a を b の親ノード，b を a の子ノードと呼ぶ．
2. **言い換えパスの追加** 入力テキストに含まれるあらゆる単語 n-gram ($n \in \{1, 2, \cdots, N\}$) $s_i\ s_{i+1}\ \cdots\ s_{i+n-1}$ のそれぞれに対し，言い換え辞書を用いて言い換えを取得し，ラティスに加える．すなわち，個々の句の言い換え $p_1\ p_2\ \cdots\ p_M$ が得られた時，$V \cup \bigcup_{k=1}^{M}\{p_k\}$ を新たな V とし，$E \cup \{(s_{i-1}, p_1), (p_M, s_{i+n})\} \cup \bigcup_{k=1}^{M-1}\{(p_k, p_{k+1})\}$ を新たな E とする．

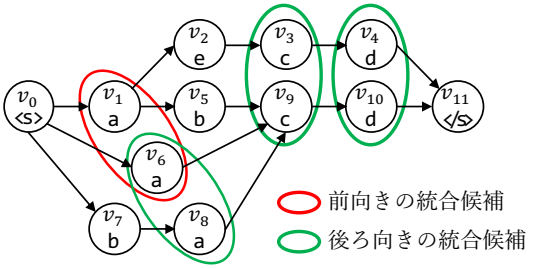

図 2　統合前のラティス（例 1）

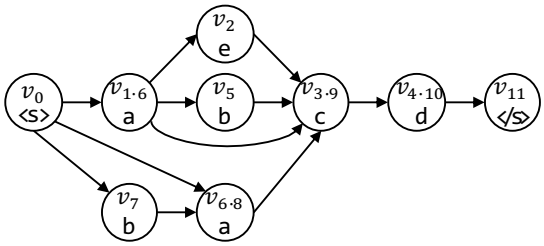

図 3　統合後のラティス（例 1）

3. **重複ノードの統合**　ラティスに含まれるパスの集合が変化しないように，ノードの重複を統合する．共通の単語を含む句の言い換えを適用した場合に，ノードの重複が起こる．重複するノード集合は実際には同一の単語を参照するため，これを統合する．ここで，ノード集合 $V'(\subset V)$ の統合とは，$\forall v \in V'$ を削除し，v に接続していた全てのエッジと接続する新たなノードを 1 つ作ることを指す．統合の大まかな手順としては，まずラティスを前向きおよび後ろ向きに探索し，統合するノード集合の候補を列挙する．次に，候補の中から実際に統合する集合を選び，ノードを統合する．
   (a) **統合候補の列挙**：ノードを Source ノードからトポロジカル順[1]に探索し，各ノードの子ノードのうち，単語ラベルが同じでかつ親ノードの集合が同じであるノードの集合を列挙する．ここで列挙されたノード集合の集合 $M_f = \{\{v, w\} \mid \forall u \in V,\ \forall v, \forall w \in C(u),\ v$ の単語ラベル $= w$ の単語ラベル $\}$ が，統合候補となる．なお，$C(u)$ は u の子ノードの集合である．同様に逆順にも探索し，統合候補 M_b を得る．ただし探索の際には，先に見つかったノード集合から貪欲に統合されると仮定して残りの探索を進めるが，ここではまだ実際には統合しない．
   (b) **ノードの統合**：M_f 中のある候補 A の要素が M_b で網羅されている場合，M_b を統合すれば A の統合は必要ない．そこで統合が必要な，M_f または M_b の片方のみに含まれるノード $D = \{v \mid \forall M \in M_f,\ \forall v \in M\} \triangle \{v \mid \forall M \in M_b,\ \forall v \in M\}$[2] を求める．

1)　有向非巡回グラフにおけるノードのソート手法にトポロジカルソートがある．これは，全ての異なる 2 つのノード a, b に対し $a \neq b$ かつノード a からノード b へ到達可能ならば a が b より先にくるようにノードを並べる手法である．ここではトポロジカルソートによる並べ方をトポロジカル順と呼ぶ．
2)　記号 $\triangle$ は，対称差集合を表す．

そして，統合候補 M_f および M_b の各要素に対して，D を含むノード集合 $M'_f = \{M \mid \forall M \in M_f,\ \exists v \in M,\ v \in D\}$ および $M'_b = \{M \mid \forall M \in M_b,\ \exists v \in M,\ v \in D\}$ を選び，これらのノード集合を統合する．

以下では統合処理の具体例を示す．

例 1　単語列 a b c d と言い換え {a b → a, a b → b a, b c d → e c d} が与えられた場合を考える．手順 1 および 2 により，これらの入力から図 2 のラティスを得る．ここで，文頭記号を<s>，文末記号を</s>としている．

手順 3.(a) では，v_0 の子ノード v_1, v_6, v_7 のうち v_1, v_6 はどちらも単語ラベルが a であり，かつ親ノードが v_0 であるため，$M_f = \{\{v_1, v_6\}\}$ が前向きの統合候補（赤囲み）と同定される．逆順でも同様に，$M_b = \{\{v_4, v_{10}\}, \{v_3, v_9\}, \{v_6, v_8\}\}$ が後ろ向きの統合候補（緑囲み）と同定される．

手順 3.(b) では，M_f と M_b のいずれか片方のみに含まれるノードとして $D = \{v_1, v_4, v_{10}, v_3, v_9, v_8\}$ が挙げられる．M_f や M_b の要素は，全て 1 つ以上の要素が D に含まれるため，$M'_f = \{\{v_1, v_6\}\}$, $M'_b = \{\{v_4, v_{10}\}, \{v_3, v_9\}, \{v_6, v_8\}\}$ となる．

これらの M'_f および M'_b の各要素を統合すると，図 3 のラティスが得られる．なお，図 2 に含まれる <s> から </s> までの全てのパスの集合は，P = {<s> → a → e → c → d → </s>, <s> → a → b → c → d → </s>, <s> → a → c → d → </s>, <s> → b → a → c → d → </s>} であるが，図 3 のように統合しても，P に変化はない．

例 2　手順 3.(b) が必要となる例として，図 4 のラティスからは，$M_f = \{\{v_3, v_5\}\}$, $M_b = \{\{v_2, v_3\}, \{v_5, v_7\}\}$, $D = \{v_2, v_7\}$ が得られる．$\{v_3, v_5\} \in M_f$ の要素 v_3, v_5 はどちらも D に含まれないため，$M'_f = \{\}$ となる．$\{v_2, v_3\}, \{v_5, v_7\} \in M_b$ はそれぞれ v_2, v_7 が D に含まれるため，$M'_b = \{\{v_2, v_3\}, \{v_5, v_7\}\}$ となる．M'_f および M'_b の統合を行うと，図 5 が得られる．

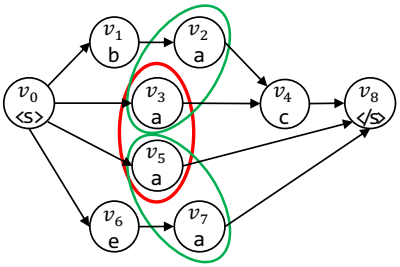

図 4　統合前のラティス（例 2）

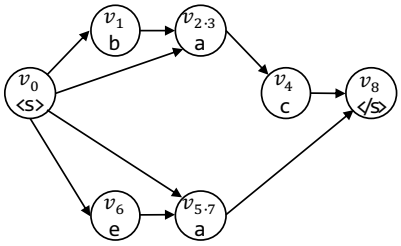

図 5　統合後のラティス（例 2）

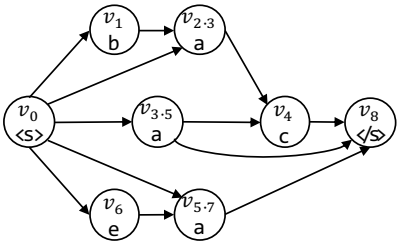

図 6　全ての候補を統合する失敗例

この例において，手順 3.(b) を省略，つまり $M'_f = M_f$ として $\{v_3, v_5\}$ も統合した場合，図 6 のラティスが得られる．このとき，パス $v_0 \to v_{2\cdot3} \to v_4$ とパス $v_0 \to v_{3\cdot5} \to v_4$ で表現される単語ラベルの系列は同じであり，またパス $v_0 \to v_{5\cdot7} \to v_8$ とパス $v_0 \to v_{3\cdot5} \to v_8$ で表現される単語ラベルの系列も同じであるため，ノード $v_{3\cdot5}$ およびこれに接続するエッジを削除してもラティス全体のパス集合は図 5 の場合と変わらない．手順 3.(b) で M'_f および M'_b を考慮することによって，重複ノードを持たない図 5 のラティスが得られる．なお，重複ノードを避ける理由については 2.3 節で説明する．

2.2　エッジの重み付け

言い換えとその前後の単語の繋がりの良さを表現するため，2.1 節で得られたラティス $G = (V, E)$ の各エッジに，次の手順で重み付けを行う．

1. エッジ $(v_j, v_k) \in E$ で接続されている 2 つのノードの単語ラベルの 2-gram 言語モデルスコアを，当該エッジの重み w_{jk} とする．
2. 各ノード v_j の出力エッジの重みの和が 1 になるよう，エッジの重みを正規化する．具体的には，ノード v_j の子ノード集合 $C(v_j)$ を考え，

$$w'_{jk} = \frac{w_{jk}}{\sum_{v_l \in C(v_j)} w_{jl}}$$

をエッジ (v_j, v_k) の新たな重みとする．

2.3　Lattice2Seq モデル

言い換えラティスを入力として，Sperber ら [19] の Lattice2Seq モデルでテキストを生成する．Sperber らは，Transformer [21] の符号化器における Positional Encoding (PE) と自己注意機構を改変し，ラティス構造の入力を可能にした．PE では，各ノードの位置を Source ノードからの最長距離として定義した．また，単語間の接続関係およびエッジの重みを表現するために，自己注意機構にマスクを導入した．

表 1　実験に使用したデータセット

データセット	訓練	検証	評価
英日翻訳：small parallel enja	50,000	500	500
スタイル変換：GYAFC (E&M)	52,595	11,508	1,416
スタイル変換：GYAFC (F&R)	51,967	11,152	1,332

ラティスに重複ノードが含まれる場合，PE の位置が異なる場合がある．例えば図 6 のラティスにおいて，Source ノード v_0 の位置を 0 とすると，同一の単語を参照する $v_{2\cdot3}$ と $v_{3\cdot5}$ の位置はそれぞれ 2 と 1 であり，異なる．このため位置を表す分散表現も異なるが，同一の単語を参照するノード同士には同一の分散表現を割り当てたい．そこで本研究では，重複ノードを含まない言い換えラティスを設計する．

3　実験

提案手法の有効性を確認するため，表 1 に示す英日翻訳および英語のスタイル変換の実験を行った．英日翻訳には，田中コーパス[3]の一部である small parallel enja[4]を使用した．スタイル変換には，カジュアルな英文とフォーマルな英文からなるパラレルコーパスである GYAFC[5] [22] を使用した．本実験では，Entertainment & Music (E&M) と Family & Relationships (F&R) の両方のドメインにおいて，カジュアルからフォーマルへの変換を行う．評価には sacreBLEU[6] [20] で求めた BLEU 値を用いた．

3.1　実験設定

前処理として，Moses ツールキット[7] [23] の `normalize-punctuation` および `tokenizer` を使用した．また，本実験で使用する言い換え辞書が小文字化されているため，Moses ツールキットの `lowercase`

3）http://www.edrdg.org/wiki/index.php/Tanaka_Corpus
4）https://github.com/odashi/small_parallel_enja
5）評価用データは，4 つの参照文を持つマルチリファレンス．
6）https://github.com/mjpost/sacrebleu
7）https://github.com/moses-smt/mosesdecoder

表 2　英日翻訳とスタイル変換（E&M・F&R）の BLEU

	翻訳	E&M	F&R
比較手法：Seq2Seq	37.89	64.27	69.46
比較手法：MultiSource	37.88	63.18	71.98
提案手法：Lattice2Seq	**40.01**	**67.30**	**73.36**

を用いて，GYAFC の入力側も小文字化した．small parallel enja は，事前に単語分割および英語の小文字化が行われているため，それに従った．両タスクで，さらに Byte Pair Encoding [24] を用いて語彙サイズ 32,000 のサブワード化を行った．

言い換え辞書には，PPDB 2.0[8] [25] のうち，英語の Lexical および Phrasal（S サイズ）を用いた．なお，収録されている語句は，事前に小文字化されている．ノイズの影響を抑えるため，適用する言い換えは PPDB 2.0 におけるスコアが θ 以上のものに限定した．このスコアの中央値は 5.3 のため，$\theta \in \{5.1, 5.2, 5.3, 5.4, 5.5\}$ の中から，検証用データにおける性能が最も高い値を選択した．

2-gram 言語モデルの訓練には，CC-100[9] の en を用いた．CC-100 (en) は約 556 億トークンからなる英語の大規模な単一言語コーパスである．前処理として Moses ツールキットの `normalize-punctuation`, `tokenizer` および `lowercase` を適用し，KenLM[10] [26] によって 2-gram 言語モデルを訓練した．

Lattice2Seq モデルの実装には，JoeyNMT[11] [27] を用いた．符号化器および復号化器には 4 層 4 ヘッドの Transformer [21] を使用した．埋込層および隠れ層は 512 次元とし，符号化器と復合化器で埋込層の重みを共有した．埋込層および隠れ層のドロップアウト率は 0.2 とした．最適化には Adam [28] を使用した．バッチサイズは 4096 で，英日翻訳タスクでは 200 回，スタイル変換タスクでは 400 回パラメータを更新するごとに検証用データにおけるパープレキシティを評価し，連続で 32 回改善しなくなったところで訓練を終了した．スケジューラには plateau を用い，初期の学習率を 0.0002 とし，連続で 8 回改善しなくなる毎に減衰率 0.7 を乗じた．

3.2　比較手法

Seq2Seq　言い換えを適用しないベースライン．

8）http://paraphrase.org/#/download
9）http://data.statmt.org/cc-100/
10）https://github.com/kpu/kenlm
11）https://github.com/joeynmt/joeynmt

表 3　PPDB 2.0 スコアの閾値 θ と検証用データの BLEU

θ	翻訳	E&M	F&R
5.1	40.00	33.83	39.13
5.2	**41.14**	34.10	39.15
5.3	40.12	**34.10**	39.36
5.4	39.44	33.61	**39.49**
5.5	40.16	32.97	39.35

MultiSource　入力テキストを言い換えるが，ラティスを構築しない比較手法．PPDB に基づく言い換え生成器[12] [29] によって $(r-1)$ $(r \in \{2,3,4,5\})$ 種類の言い換えを取得し，入力文を含めた r 文を連結して Seq2Seq モデルに入力する．r は，検証用データにおける性能が最も高い値を選択した．

3.3　実験結果

表 2 に，英日翻訳（翻訳）およびスタイル変換（E&M および F&R）の BLEU 値を示す．英日翻訳では，提案手法は Seq2Seq と比較して 2.12 ポイント，MultiSource と比較して 2.13 ポイント，それぞれ大幅に BLEU を改善した．スタイル変換でも，提案手法は Seq2Seq から 3.03 ポイント（E&M）および 3.90 ポイント（F&R），MultiSource から 4.12 ポイント（E&M）および 1.38 ポイント（F&R），それぞれ大幅に BLEU を改善した．これらの実験結果から，言い換えラティスの有効性を確認できた．

表 3 に，使用する言い換えの閾値 θ を変化させたときの検証用データにおける評価結果を示す．θ が小さいと低品質な言い換えを含んでしまい，大きいと有用な言い換えを使用できないことが分かる．

4　おわりに

本研究では，語彙選択の曖昧性をモデル化するために，入力テキストおよび言い換え辞書からラティスを構築する手法を提案した．英日翻訳およびスタイル変換の実験を通じて，提案手法によって性能を大幅に改善できることを示した．

今後は，語順の交替など語句の言い換え以外の言い換えへの拡張を検討する．また，テキスト分類など他のタスクにおける有効性を検証する．

謝辞

本研究は JST（ACT-X，課題番号: JPMJAX1907）の支援を受けたものです．

12）https://cwiki.apache.org/confluence/pages/viewpage.action?pageId=65142863

参考文献

[1] Liangyou Li, Andy Way, and Qun Liu. Graph-Based Translation Via Graph Segmentation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pp. 97–107, 2016.

[2] Jasmijn Bastings, Ivan Titov, Wilker Aziz, Diego Marcheggiani, and Khalil Sima'an. Graph Convolutional Encoders for Syntax-aware Neural Machine Translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 1957–1967, 2017.

[3] Danqing Wang, Pengfei Liu, Yining Zheng, Xipeng Qiu, and Xuanjing Huang. Heterogeneous Graph Neural Networks for Extractive Document Summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 6209–6219, 2020.

[4] Wei Li, Xinyan Xiao, Jiachen Liu, Hua Wu, Haifeng Wang, and Junping Du. Leveraging Graph to Improve Abstractive Multi-Document Summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 6232–6243, 2020.

[5] Taichi Ishiwatari, Yuki Yasuda, Taro Miyazaki, and Jun Goto. Relation-aware Graph Attention Networks with Relational Position Encodings for Emotion Recognition in Conversations. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pp. 7360–7370, 2020.

[6] Jie Zhu, Junhui Li, Muhua Zhu, Longhua Qian, Min Zhang, and Guodong Zhou. Modeling Graph Structure in Transformer for Better AMR-to-Text Generation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pp. 5459–5468, 2019.

[7] Shaowei Yao, Tianming Wang, and Xiaojun Wan. Heterogeneous Graph Transformer for Graph-to-Sequence Learning. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 7145–7154, 2020.

[8] Linfeng Song, Ante Wang, Su Jinsong, Yue Zhang, Kun Xu, Yubin Ge, and Dong Yu. Structural Information Preserving for Graph-to-Text Generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 7987–7998, 2020.

[9] Yanbin Zhao, Lu Chen, Zhi Chen, Ruisheng Cao, Su Zhu, and Kai Yu. Line Graph Enhanced AMR-to-Text Generation with Mix-Order Graph Attention Networks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 732–741, 2020.

[10] Deng Cai and Wai Lam. AMR Parsing via GraphSequence Iterative Inference. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 1290–1301, 2020.

[11] Yue Zhang and Jie Yang. Chinese NER Using Lattice LSTM. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pp. 1554–1564, 2018.

[12] Xiaonan Li, Hang Yan, Xipeng Qiu, and Xuanjing Huang. FLAT: Chinese NER Using Flat-Lattice Transformer. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 6836–6842, 2020.

[13] Jinsong Su, Zhixing Tan, Deyi Xiong, Rongrong Ji, Xiaodong Shi, and Yang Liu. Lattice-Based Recurrent Neural Network Encoders for Neural Machine Translation. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, pp. 3302–3308, 2017.

[14] Fengshun Xiao, Jiangtong Li, Hai Zhao, Rui Wang, and Kehai Chen. Lattice-Based Transformer Encoder for Neural Machine Translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 3090–3097, 2019.

[15] Huda Khayrallah, Gaurav Kumar, Kevin Duh, Matt Post, and Philipp Koehn. Neural Lattice Search for Domain Adaptation in Machine Translation. In *Proceedings of the The 8th International Joint Conference on Natural Language Processing*, pp. 20–25, 2017.

[16] Matthias Sperber, Graham Neubig, Jan Niehues, and Alex Waibel. Neural Lattice-to-Sequence Models for Uncertain Inputs. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 1380–1389, 2017.

[17] Daniel Beck, Trevor Cohn, and Gholamreza Haffari. Neural Speech Translation using Lattice Transformations and Graph Networks. In *Proceedings of the Thirteenth Workshop on Graph-Based Methods for Natural Language Processing*, pp. 26–31, 2019.

[18] Pei Zhang, Boxing Chen, Niyu Ge, and Kai Fan. Lattice Transformer for Speech Translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 6475–6484, 2019.

[19] Matthias Sperber, Graham Neubig, Ngoc-Quan Pham, and Alex Waibel. Self-Attentional Models for Lattice Inputs. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 1185–1197, 2019.

[20] Matt Post. A Call for Clarity in Reporting BLEU Scores. In *Proceedings of the Third Conference on Machine Translation*, pp. 186–191, 2018.

[21] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention Is All You Need. In *Proceedings of the 31st Conference on Neural Information Processing Systems*, pp. 5998–6008, 2017.

[22] Sudha Rao and Joel Tetreault. Dear Sir or Madam, May I Introduce the GYAFC Dataset: Corpus, Benchmarks and Metrics for Formality Style Transfer. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 129–140, 2018.

[23] Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. Moses: Open Source Toolkit for Statistical Machine Translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, Demo and Poster session*, pp. 177–180, 2007.

[24] Rico Sennrich, Haddow Barry, and Alexandra Birch. Neural Machine Translation of Rare Words with Subword Units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pp. 1715–1725, 2016.

[25] Ellie Pavlick, Pushpendre Rastogi, Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. PPDB 2.0: Better paraphrase ranking, fine-grained entailment relations, word embeddings, and style classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, pp. 425–430, 2015.

[26] Kenneth Heafield. KenLM: Faster and Smaller Language Model Queries. In *Proceedings of the 6th Workshop on Statistical Machine Translation*, pp. 187–197, 2011.

[27] Julia Kreutzer, Jasmijn Bastings, and Stefan Riezler. Joey NMT: A Minimalist NMT Toolkit for Novices. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (System Demonstrations)*, pp. 109–114, 2019.

[28] Diederik P. Kingma and Jimmy Lei Ba. Adam: A Method for Stochastic Optimization. In *Proceedings of the 3rd International Conference on Learning Representations*, pp. 1–15, 2015.

[29] Courtney Napoles, Chris Callison-Burch, and Matt Post. Sentential Paraphrasing as Black-Box Machine Translation. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Demonstrations)*, pp. 62–66, 2016.