同期注意制約を与えた依存構造に基づく Transformer NMT

出口 祥之 [1]　田村 晃裕 [2]　二宮 崇 [1]
[1] 愛媛大学　[2] 同志社大学
[1] {deguchi@ai., ninomiya@}cs.ehime-u.ac.jp　[2] aktamura@mail.doshisha.ac.jp

1 はじめに

近年，機械翻訳の分野において，Transformer ニューラル機械翻訳（Neural Machine Translation; NMT）モデル（以下 Transformer NMT）[1] が従来の回帰型ニューラルネットワークや畳み込みニューラルネットワークベースモデルの翻訳性能を上回り，注目を浴びている．特に，Transformer NMT の特徴の一つである自己注意は注意重みと呼ばれる確率分布行列を計算することで文内における単語間の関連の強さを捉えることができ，ここに依存構造を組み込むことで翻訳性能をさらに改善するモデルも提案されている [2, 3, 4]．

これまで，統計的機械翻訳では原言語文と目的言語文間の同期文法や同期依存文法を考慮することで翻訳性能を改善してきた [5, 6]．同期文法及び同期依存文法は，2 言語で定義される文法であり，2 言語の文構造を同時に生成する文法である．NMT では，この同期依存文法から着想を得た同期注意制約 [7] が提案されている．同期注意制約では，Transformer NMT の原言語側の自己注意と目的言語側の自己注意の間で言語間注意を通した注意の整合性を保つように各注意が学習される．しかし，Transformer NMT の自己注意の一部では依存構造を捉える場合があるという実験結果が報告されているものの [8]，同期注意制約を用いる従来手法は，自己注意と言語間注意の整合性を保証するだけであり，必ずしも依存構造を捉える自己注意が学習されるとは限らない．

そこで本稿では，原言語文と目的言語文の依存構造をそれぞれエンコーダとデコーダの自己注意で捉える Transformer NMT [2] に同期注意制約を組み込むことで，言語間で対応をもたせた依存構造に基づく NMT を提案する．また，従来の同期注意制約では制約を付与する際に別途注意の重みを計算する必要があったが，本稿では，モデルの順伝播時に得られる計算結果を利用することで注意の重みを別途求める必要のない手法を提案する．図 1 は対訳文におけるアライメント及び各文の依存構造の例を表している．提案モデルでは，図 1 において，“you” の親が “speak” であり，“you” が “du” に，“speak” が “Kannst” に対応している時，“du” の親が “Kannst” となるように学習する．

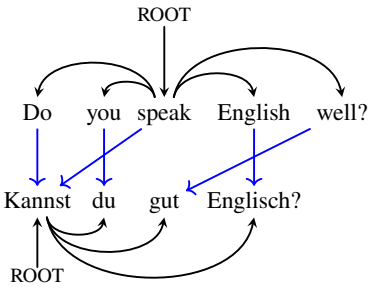

図 1: 依存構造とアライメント関係の例

提案モデルと従来モデルの翻訳性能を BLEU [9] を用いた評価により比較したところ，WAT Asian Scientific Paper Excerpt Corpus（ASPEC）日英翻訳タスクで最大 +0.36 ポイントの性能改善を確認した．

2 Transformer NMT

Transformer NMT [1] は，入力された原言語文 $X = (x_1, x_2, \ldots, x_I)$ を符号化する Transformer エンコーダ（以下エンコーダ）と，エンコーダの出力を受け取り目的言語文 $Y = (y_1, y_2, \ldots, y_J), \forall y_j \in \mathcal{V}$ に復号する Transformer デコーダ（以下デコーダ）を組み合わせたエンコーダ・デコーダモデルである．ただし，$\mathcal{V}$ は出力語彙集合である．

エンコーダ及びデコーダはそれぞれ N_{enc} 層のエンコーダ層，N_{dec} 層のデコーダ層から成る．各エンコーダ層及びデコーダ層は複数のサブレイヤから成り，全てのサブレイヤの出力には残差接続と層正規化が適用される．エンコーダ層は下層より順に自己注意層，順伝播層から構成され，デコーダ層は下層より順に自己注意層，言語間注意層，順伝播層から構成される．

自己注意や言語間注意は，複数ヘッド注意 $\mathrm{Attn}(Q,K,V)$ により，d_{emb} 次元の単語埋め込み空間から H 個の $d_k(=\frac{d_{emb}}{H})$ 次元のヘッド（部分空間）に射影し，各ヘッドごとに次式のように注意を計算する．

$$\mathrm{Attn}(Q,K,V) = [M_1; M_2; \ldots; M_H]W^M, \quad (1)$$

$$M_h = A_h V_h,\ A_h = \mathrm{softmax_r}\left(\frac{Q_h K_h^\top}{\sqrt{d_k}}\right), \quad (2)$$

$$Q_h = QW_h^Q, K_h = KW_h^K, V_h = VW_h^V. \quad (3)$$

なお，$\mathrm{softmax_r}()$ は行列に対して行毎に softmax 関数を適用する関数である．複数ヘッド注意の入力 Q，K，V は，それぞれパラメータ行列 $W_h^Q, W_h^K, W_h^V \in \mathbb{R}^{d_{emb}\times d_k}$ によって，Q_h, K_h, V_h（$1 \le h \le H$）に射影される．ここで，注意重み行列 A_h の各要素の値は Q の各単語と K の各単語の間の関連の強さを示している．続いて，注意重み行列 A_h に V_h を掛けることで Q_h に対応する表現を V_h から荷重和の形で得られる．その後，全ヘッドの M_h（すなわち，$M_1, M_2, \ldots, M_H$）を結合し，パラメータ行列 $W^M \in \mathbb{R}^{d_{emb}\times d_{emb}}$ によって d_{emb} 次元の単語埋め込み空間に射影する．

なお，自己注意では，入力の Q，K，V はいずれも前のサブレイヤの出力であり，原言語文，目的言語文それぞれにおいて文中の全ての単語との関連の強さを計算する．デコーダ側のみで用いられる言語間注意では，入力 Q は前のサブレイヤの出力，K，V はどちらもエンコーダの最終層の出力であり，目的言語文の各単語と原言語文との関連の強さを計算する．ただし，翻訳時に目的言語文の各単語は自己回帰的に生成されるため，デコーダ側の自己注意は各単語の後方の単語を指さないようマスクをかけて訓練する．

最後に，デコーダの最終層の出力を $|\mathcal{V}|$ 次元の空間に線形変換し，各単語毎に softmax 関数を適用することで，目的言語文の生成確率 $\Pr(Y|X) = \prod_{j=1}^{J} p(y_j \in \mathcal{V} | y_{<j}, X)$ を計算する．

3 提案手法

本稿で提案するモデルは，原言語側と目的言語側の自己注意でそれぞれの文の依存構造を捉え，それらを同期注意制約によって対応付ける．

また，従来の同期注意制約の手法の問題点として，制約を付与する際に逆向きの言語間注意（原言語文から目的言語文への注意）を追加で計算する必要があった．本稿では Transformer NMT の順伝播計算時の注意重みを保持することで，追加の計算なしで逆向きの言語間注意を得る手法を提案する．

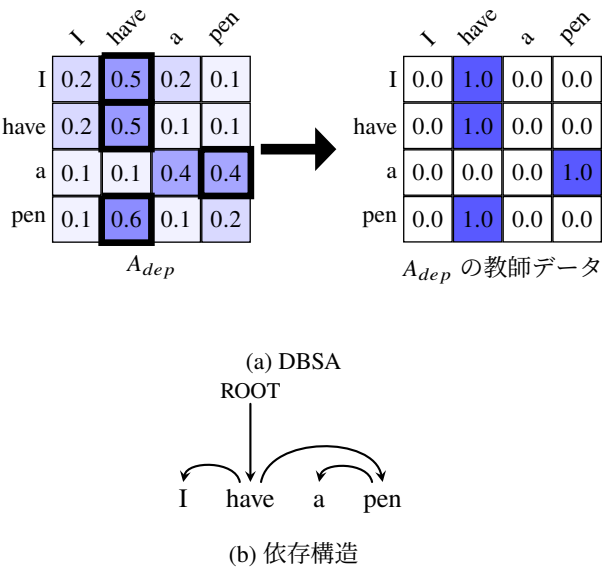

図 2: 依存構造と DBSA の例

依存構造に基づいた自己注意　はじめに，依存構造に基づいた自己注意（dependency-based self-attention; DBSA）[2] について説明する．DBSA を用いた Transformer NMT（以下 Transformer+DBSA）は，モデルの翻訳の損失に加え，以下のような依存構造解析の損失 $\mathcal{L}_{dep}$ を制約として付与する．

$$\mathcal{L}_{dep} = -\sum_{i=1}^{I} \log \Pr(\mathrm{head}(x_i) \mid X) - \sum_{j=1}^{J} \log \Pr(\mathrm{head}(y_j) \mid Y). \quad (4)$$

ただし，head() は引数に与えた単語の親を返す関数である．DBSA は，エンコーダまたはデコーダの第 l_{dep} 層目の自己注意において，複数ヘッドのうち 1 つのヘッドで依存構造を捉える．依存構造を捉えるヘッドの Q_h，K_h，V_h をそれぞれ Q_{dep}，K_{dep}，V_{dep} とすると，2 単語間の依存関係を示す注意重み行列 A_{dep} は以下のようなパラメータ行列 $U \in \mathbb{R}^{d_k\times d_k}$ を用いた bi-affine 変換 [10] によって得られる．

$$A_{dep} = \mathrm{softmax_r}\left(\frac{Q_{dep} U K_{dep}^\top}{\sqrt{d_k}}\right). \quad (5)$$

A_{dep} において，$A_{dep}[t,q]$ は単語 q が単語 t の親である確率を示す．

$$A_{dep}[t,q] = \Pr(q = \mathrm{head}(t) \mid S). \quad (6)$$

ここで，S は原言語文または目的言語文である．ただし，親が ROOT となる単語 t_{ROOT} は自身を指すように設定する（$\text{head}(t_{\text{ROOT}}) = t_{\text{ROOT}}$）．次に，$M_{dep}$ は依存関係に基づいた注意重み A_{dep} と V_{dep} を掛け合わせることで得られる．

$$M_{dep} = A_{dep} V_{dep}. \tag{7}$$

最後に，M_{dep} を他のヘッドと結合し，通常の自己注意と同様にして d_{emb} 次元の埋め込み空間に射影する．

$$\text{Attn}^{(l_{dep})} = [M_{dep}; M_1; \ldots; M_{H-1}] W^M. \tag{8}$$

ここで，$\text{Attn}^{(l_{dep})}$ は第 l_{dep} 層目の自己注意を表す．なお，デコーダ側の DBSA については，通常の自己注意と同様に後方の単語を指さないようマスクをかけて訓練し，親が後方に存在する単語に対しては制約を与えない．図 2 に依存構造と注意重み行列 A_{dep}，A_{dep} の教師データの例を示す．図 2a では，色の濃い要素ほど高い値を示している．

DBSA は，現在 NMT で広く用いられているサブワード単位 [11, 12] にも拡張されている．ある単語が複数のサブワードに分割された場合，その単語を構成する各サブワードの親は隣接する後方のサブワードとし，単語の終端となるサブワードの親はその単語の親に設定する．また，親となる単語が複数のサブワードに分割されている場合，親は先頭のサブワードとする．

同期注意制約　続いて，Transformer NMT で原言語文の文構造と目的言語文の文構造の対応を捉える同期注意制約 [7] について説明する．同期注意制約を与えた Transformer NMT は翻訳の損失に加え，原言語側の自己注意で捉えた文構造と目的言語側の自己注意で捉えた文構造の整合性を保つための損失 $\mathcal{L}_{sync}$ を制約として付与する．$\mathcal{L}_{sync}$ は，言語間注意によって，原言語側のエンコーダ自己注意を目的言語側の確率空間に変換し，デコーダ自己注意との差を計算することで得られる．目的言語側の空間に変換されたエンコーダ自己注意を D^{pseudo} とすると，以下の式によって D^{pseudo} が求められる．

$$D^{pseudo} = \overrightarrow{C} E \overleftarrow{C}. \tag{9}$$

ただし，E と D はそれぞれエンコーダ及びデコーダの l_n 層目の自己注意の注意重み行列 A_h であり，$\overrightarrow{C}$，$\overleftarrow{C}$ はそれぞれ，以下のような式で求められる順方向（目的言語側から原言語側へ）の言語間注意と逆方向（原言語側から目的言語側へ）の言語間注意である．

$$\overrightarrow{C} = \text{softmax}_{\text{r}}\left(\frac{Q_h K_h^\top}{\sqrt{d_k}}\right), \tag{10}$$

$$\overleftarrow{C} = \text{softmax}_{\text{r}}\left(\frac{(Q_h K_h^\top)^\top}{\sqrt{d_k}}\right) = \text{softmax}_{\text{r}}\left(\frac{K_h Q_h^\top}{\sqrt{d_k}}\right). \tag{11}$$

なお，Q_h と K_h はデコーダの第 l_{sync} 層の言語間注意の入力である．次に，デコーダ自己注意は各単語における後方の単語をマスクして訓練するため，同様に D^{pseudo} に対しても後方の単語を指さないようマスクをかけて softmax 関数を適用する．

$$D' = \text{softmax}_{\text{r}}\left(\text{mask}\left(D^{pseudo}\right)\right). \tag{12}$$

ただし，mask は各単語における後方の単語をマスクする関数である．最後に，D' と D の最小自乗誤差を計算することで $\mathcal{L}_{sync}$ を得る．

$$\mathcal{L}_{sync} = \sum_{t,q}\left(D'_{t,q} - D_{t,q}\right)^2. \tag{13}$$

図 3 に同期注意制約の例を示す．図 3 に示すように，エンコーダ自己注意 E は言語間注意 $\overrightarrow{C}$，$\overleftarrow{C}$ によって目的言語側の確率空間に変換される．制約の損失はデコーダ自己注意 D と求めた D' との差で表される．

提案モデル　本稿で提案するモデルは Transformer+DBSA に同期注意制約を付与することで，原言語文の依存構造と目的言語文の依存構造の対応をもたせる．提案モデルは次式のような目的関数 $\mathcal{L}$ を最小化することにより学習する．

$$\mathcal{L} = \mathcal{L}_t + \lambda_{dep}\mathcal{L}_{dep} + \lambda_{sync}\mathcal{L}_{sync}. \tag{14}$$

なお，$\lambda_{dep}, \lambda_{sync} > 0$ はそれぞれ損失関数において $\mathcal{L}_{dep}, \mathcal{L}_{sync}$ が影響する度合いを調整するハイパーパラメータである．ここで，同期注意制約に用いる原言語側と目的言語側の自己注意は第 l_{dep} 層目，すなわち，$l_n = l_{dep}$ とする．

また，従来の同期注意制約では，式 11 のように，$\mathcal{L}_{sync}$ を求める際に逆向き言語間注意 $\overleftarrow{C}$ を計算する必要があった．本稿では，逆方向言語間注意を順方向言語間注意の転置，すなわち，$\overleftarrow{C} = \overrightarrow{C}^\top$ とすることで，Transformer のモデル順伝播計算時の注意重みを保持するだけで制約を算出できるようにする．

4　実験

実験設定　本実験では，提案手法の有効性を確認するため，提案手法を適用したモデルと従来モデルとの翻訳性能を比較する．Transformer モデルに

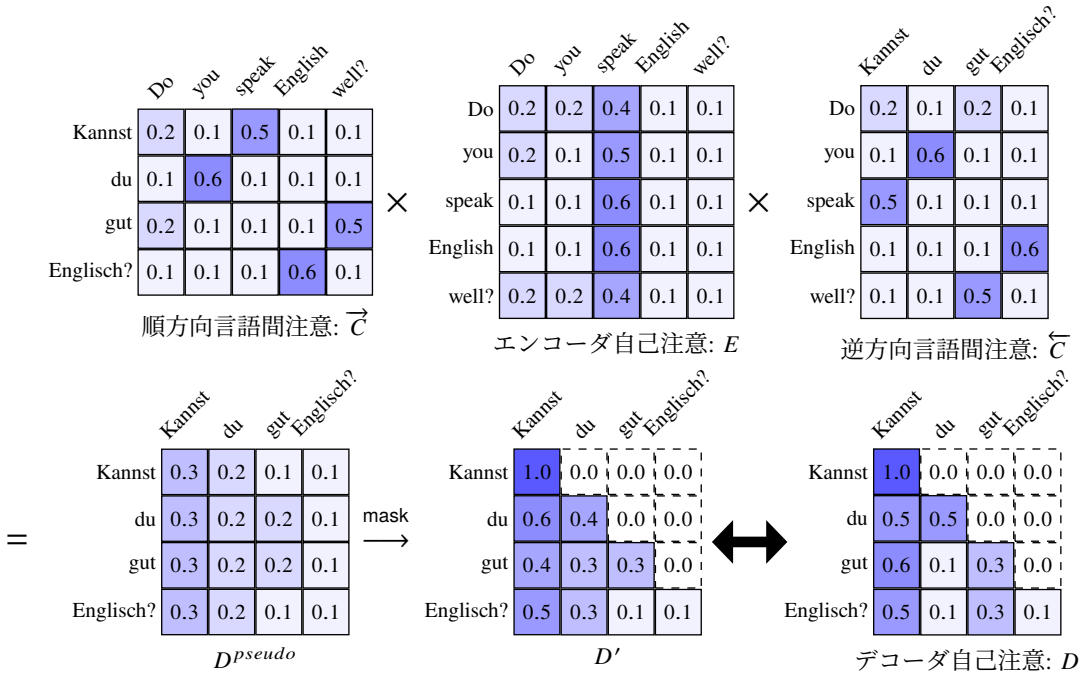

図 3: 同期注意制約の計算例

は base [1] モデルを使用した．モデルの詳細なハイパーパラメータは付録に記載した．

翻訳性能の評価実験は，ASPEC[13] 日英翻訳タスクを用いた．訓練データは上位 150 万を抽出して用いた．単語分割器は，日本語文には KyTea [14] を，英語文には Moses トークナイザを用いた．単語分割した後は訓練データから学習したバイトペア符号化（Byte Pair Encoding; BPE）[11] によりサブワード分割を行い，BPE の語彙数は原言語側と目的言語側で独立してそれぞれ 16,000 に設定した．DBSA の教師データは依存構造解析器の出力結果を用い，英語には Stanza [15] を，日本語には EDA [16] を使用した．ただし，解析器は訓練時の教師データ作成のためのみに使用し，翻訳時には使用していない．その他，詳細なデータ前処理は付録に記載した．

目的関数の $\mathcal{L}_t$ はラベル平滑化交差エントロピーを用いた．各損失を考慮する度合いを調整するハイパーパラメータは $\lambda_{dep} = 0.5$，$\lambda_{sync} = 10.0$ とした．同期注意制約を与える層 l_{dep} は Transformer base モデルの第 1 層目とし，D' を求める際に必要な言語間注意の層 l_{sync} は第 5 層目とした．モデルのパラメータ更新回数は 100,000 回とした．翻訳文の生成にはビーム探索を用い，ビーム幅は 4 とした．

実験結果 実験結果を表 1 に示す．表中の SyncAttn は同期注意制約を表す．表より，DBSA と SyncAttn を組み合わせることで，従来の DBSA，SyncAttn 単体で用いたモデルよりそれぞれ+0.27，+0.36 BLEU(%) の性能改善を確認した．

表 1: 実験結果 (BLEU(%))

Transformer	28.94
+DBSA	29.57
+SyncAttn	29.48
+DBSA+SyncAttn	**29.84**

5 おわりに

本稿では，Transformer NMT の自己注意において，原言語文と目的言語文の依存構造を捉えさせ，言語間でそれらの対応をもたせるモデルを提案した．また，従来の対応付けでは注意の重みを別途計算する必要があったが，提案手法では順伝播時の計算結果を保持しておくだけで制約が計算できる．提案モデルを用いることで，ASPEC 日英翻訳タスクにおいて，従来モデルよりも最大 +0.36 BLEU(%) の性能改善が確認された．今後は，他の言語対についても提案手法の有効性を確認したい．

謝辞

本研究成果は，国立研究開発法人情報通信研究機構の委託研究により得られたものである．また，本研究の一部は JSPS 科研費 20K19864 の助成を受けたものである．ここに謝意を表する．

参考文献

[1]Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems 30*, pp. 5998–6008. 2017.

[2]Hiroyuki Deguchi, Akihiro Tamura, and Takashi Ninomiya. Dependency-based self-attention for transformer NMT. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, pp. 239–246, Varna, Bulgaria, September 2019. INCOMA Ltd.

[3]Xing Wang, Zhaopeng Tu, Longyue Wang, and Shuming Shi. Self-attention with structural position representations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 1403–1409, Hong Kong, China, November 2019. Association for Computational Linguistics.

[4]Emanuele Bugliarello and Naoaki Okazaki. Enhancing machine translation with dependency-aware self-attention. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 1618–1627, Online, July 2020. Association for Computational Linguistics.

[5]Hongfei Jiang, Muyun Yang, Tiejun Zhao, Sheng Li, and Bo Wang. A statistical machine translation model based on a synthetic synchronous grammar. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, pp. 125–128, Suntec, Singapore, August 2009. Association for Computational Linguistics.

[6]Yuan Ding and Martha Palmer. Machine translation using probabilistic synchronous dependency insertion grammars. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pp. 541–548, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics.

[7]出口祥之, 田村晃裕, 二宮崇. 同期注意制約を与えた transformer によるニューラル機械翻訳. 言語処理学会第 26 回年次大会 発表論文集, pp. 1459–1462, 2020.

[8]Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 5797–5808, Florence, Italy, July 2019. Association for Computational Linguistics.

[9]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, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics.

[10]Timothy Dozat and Christopher D. Manning. Deep biaffine attention for neural dependency parsing. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017.

[11]Rico Sennrich, Barry Haddow, 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 (Volume 1: Long Papers)*, pp. 1715–1725, Berlin, Germany, August 2016. Association for Computational Linguistics.

[12]Taku Kudo and John Richardson. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 66–71, Brussels, Belgium, November 2018. Association for Computational Linguistics.

[13]Toshiaki Nakazawa, Manabu Yaguchi, Kiyotaka Uchimoto, Masao Utiyama, Eiichiro Sumita, Sadao Kurohashi, and Hitoshi Isahara. ASPEC: Asian scientific paper excerpt corpus. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pp. 2204–2208, Portorož, Slovenia, May 2016. European Language Resources Association (ELRA).

[14]Graham Neubig and Shinsuke Mori. Word-based partial annotation for efficient corpus construction. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta, May 2010. European Language Resources Association (ELRA).

[15]Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. Stanza: A python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pp. 101–108, Online, July 2020. Association for Computational Linguistics.

[16]Daniel Flannery, Yusuke Miayo, Graham Neubig, and Shinsuke Mori. Training dependency parsers from partially annotated corpora. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pp. 776–784, Chiang Mai, Thailand, November 2011. Asian Federation of Natural Language Processing.

[17]Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Łukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, Jeffrey Dean. Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, Vol. abs/1609.08144, , 2016.

A　データセット

表 2: 実験データ（ASPEC 日英）の対訳文対数

訓練	開発	評価
1,428,181	1,790	1,812

前処理　英語文の単語分割は，aggressive hyphen splitting オプションを付けた Moses トークナイザを使用した．また，訓練データに対しては `normalize-punctuation.perl`[1] を用いてデータの正規化を行った．訓練データには，サブワード分割する前の原言語文と目的言語文のトークン数がともに 100 以下の対訳文対を用い，トークン数の比が 2 を超える対訳文対を除外した．

B　ハイパーパラメータ

表 3: Transformer モデルのハイパーパラメータ

ミニバッチサイズ	約 12,000 トークン
ラベル平滑化	$\epsilon = 0.1$
ドロップアウト	$p = 0.1$
最適化器	Adam ($\beta_1 = 0.9, \beta_2 = 0.98$)
学習率調整	更新回数の逆平方根に比例して減衰
学習率調整（初期）	4000 回更新まで線形的に増加
最大学習率	7e-4
文長正則化	$\alpha = 0.6$ [17]

[1] https://github.com/moses-smt/mosesdecoder/blob/master/scripts/tokenizer/normalize-punctuation.perl