雑音のある通信路モデルを用いた句構造解析

原田慎太朗 [1]　渡辺太郎 [1]　大内啓樹 [1,2]
[1] 奈良先端科学技術大学院大学　[2] 理化学研究所
{harada.shintaro.hk4,taro,hiroki.ouchi}@is.naist.jp

概要

構文解析モデルである分類/識別モデルおよび結合モデルは多くの学習データを必要する．しかし、学習データとして利用される Treebank の構築には膨大な人日を必要とするため、低資源下で機能する構文解析モデルが求めらる．これに対して、雑音のある通信路モデルを用いた句構造解析モデルを提案する．雑音のある通信路モデルは問題を分割しており、学習に必要なデータ数を抑えることができる．

本研究では、構造が比較的単純な sequence-to-sequence モデルで句構造解析を定式化する．実験結果として、英語および中国語データセットにおいて、本モデルが有効であることを確認した．また、低資源設定においても有効であることを確認した．

1　はじめに

構文解析は、単語関係または階層的構造を計算する問題である．その結果、入力文の解釈性が高まるため、言語学や自然言語処理における基本的なタスクである．それゆえに研究は盛んであり、多数の高精度かつドメイン特有の句構造解析モデルが提案されている．構文解析モデルである分類/識別モデルあるいは結合モデルには非常に多くの学習データが必要である．しかし、構文解析の学習データとして使用される Treebank の訓練データは、他のデータと比較すると非常に少ない．さらに、Treebank の構築および拡張には莫大な時間と言語学的な専門性が必要である．

この課題に対処するために、雑音のある通信路モデルを用いた句構造解析モデルを提案する．雑音のある通信路モデルは、通信路モデルと言語モデルに分割できる．通信路モデルは入力をうまく説明する出力を選択ように学習する．これにより、直接モデルで起こりうる、入力の一部から出力が予測されるという問題を回避でき [1]、その結果、データシフトに頑健なモデルになる．言語モデルは、対ではないデータから学習可能であり、出力言語における良さをモデル化できる．雑音のある通信路モデルは誤り訂正や音声認識の分野で使用されてきたモデルであり、近年では深層学習に拡張され [2]、機械翻訳 [3, 4] や対話生成 [5] などで優れた性能を発揮している．特に、低資源設定では通常のモデルより大きな効果を発揮できる．

本研究では、雑音のある通信路モデル用いるにあたり構文解析を Seq2Seq（sequence-to-sequence）問題として定式化する．英語（PTB: Penn Treebank） および 中国語（CTB: Chinese Treebank）における実験、および、低資源設定における結果より、本モデルが有効であることを確認した．

2　関連研究

構文解析アルゴリズム　句構造解析において、Chart 型、Transition 型、Seq2Seq 型が代表的なアルゴリズムとして挙げられる．Chart 型は、各スパンのスコアを計算し、動的計画法を用いて生成可能な木構造を求める．そのため、高性能であるが文長 n に対して $\mathcal{O}(n^3)$ の計算量を持つ．Transition 型は、履歴を考慮した語彙的特徴を持つ Shift-Reduce 操作を繰り返し選択することにより、木構造を求める．最先端の性能は持たないが文長 n に対して $\mathcal{O}(n)$ の計算量で解析可能である．Seq2Seq 型は Chart 型および Transition 型と比較して、モデル設計が比較的簡単である．また、ビームサーチデコーディングを採用することで、出力長 m に対してビームサイズ k で性能と計算量 $\mathcal{O}(km)$ を調整できる．

Seq2Seq 型は線形化した構文木（S 式）を構成する開/閉括弧およびラベルを単語と見なすことで定式化される [6]．ここにおける線形化とは、構文木の終端ノードを記号に置き換える操作である．線形化手法には様々な方法 [6, 7, 8] が提案されているが、本研究では代表的な S 式の線形化 [6, 7] を用いる．その例を表 1 に示す．ここで、表 1 における LBD（Linearized by Dummy）、LBP（Linearized by

Pre-Terminal)、LBT（Linearized by Treminal）はそれぞれ線形化手法の名称であり、それぞれ終端ノードをダミー記号 “XX“、前終端記号、終端記号で置き換える．さらに、開/閉括弧にもラベルを付ける．

構文解析モデル　構文解析モデルは大きく分けて分類/識別モデルおよび結合モデルに分けられる．分類/識別のモデルは、Transition-based モデル [9]、Chart-based モデル [10]、Seq2Seq モデル [6, 7, 11] が挙げられる．Transition-based モデルと Chart-based モデルはそれぞれ木構造を構成する操作とラベル付きスパンを識別する．それに対して、Seq2Seq モデルは木構造を構成する線形化された記号を識別する．結合モデルは、木構造を構成するルールの生成確率をモデル化する手法 [12, 13]、および、木構造を構成する単語および操作の生成確率をモデル化する手法 [14] などが挙げられる．最尤推定、あるいは、ニューラルネットワークによる学習のために ha 、多くのデータを必要とする．

3　雑音のある通信路句構造解析器

本研究では、構文解析を Seq2Seq モデルを用いて以下のようにモデル化する．

$$P(\boldsymbol{y}|\boldsymbol{x}) = \sum_{i=1}^{|\boldsymbol{y}|} P(y_i|\boldsymbol{y}_{<i}, \boldsymbol{x}) \tag{1}$$

ここで，$\boldsymbol{x}$ は単語列、$\boldsymbol{y}$ を線形化された記号列、$|\boldsymbol{y}|$ は出力する記号列の長さであり，$\boldsymbol{y}_{<i}$ は記号 y_i より前の記号列 $x_1, ..., y_{i-1}$ を表す．ここで、ベイズの定理 $P(\boldsymbol{y}|\boldsymbol{x}) \propto P(\boldsymbol{x}|\boldsymbol{y})P(\boldsymbol{y})$ より、雑音のある通信路モデルは、線形化された記号列 $\boldsymbol{y}$ から単語列 $\boldsymbol{x}$ を生成する通信路モデル $P(\boldsymbol{x}|\boldsymbol{y})$ と線形化された記号 $\boldsymbol{y}$ の言語モデル $P(\boldsymbol{y})$ から以下のようにモデル化される．

$$P(\boldsymbol{x}|\boldsymbol{y}) = \sum_{i=1}^{|\boldsymbol{x}|} P(x_i|\boldsymbol{x}_{<i}, \boldsymbol{y}) \tag{2}$$

ここで，$|\boldsymbol{x}|$ は入力文の長さであり，$\boldsymbol{x}_{<i}$ は単語 x_i より前の単語列 $x_1, ..., x_{i-1}$ を表す．

雑音のある通信路モデルは、出力から入力を予測するため低資源設定でも効果を発揮できることが報告されている [2]．そのため、比較的学習データ数の少ない Treebank でも有効であることが期待できる．さらには、言語モデルで木構造の良さをモデル化することが可能である．そのため、既存のモデルとは異なり、木構造を意識しながらデコーディングすることで学習時と推論時のズレを抑えることが期待できる．事実、先行研究 [15] は直接モデルに木構造の保証するための制約付きデコーディングを導入しているが、制約無しと比べて精度が低下している．

雑音のある通信路モデルを用いて $\boldsymbol{x}$ から $\boldsymbol{y}$ を生成するためには、$\mathrm{argmax}_{\boldsymbol{y}} \log P(\boldsymbol{x}|\boldsymbol{y}) + \log P(\boldsymbol{y})$ を計算する必要がある．しかし、通信路モデル $P(\boldsymbol{x}|\boldsymbol{y})$ は各候補 $\boldsymbol{y}_{<i}$ に対して条件付きであり、各語彙に対して別々の順伝搬が必要なため計算量が大きくなる．この問題を軽減するに、先行研究 [2, 3, 4] ではビームサイズ $k1$ と $k2$ を用いた 2 段階のビームサーチを採用しており、直接モデル、通信路モデル、言語モデルの線形結合でデコーディングする．

$$\frac{\lambda_{dir}}{n} \log P(y|x) + \frac{\lambda_{ch}}{m} \log P(x|y) + \frac{\lambda_{lm}}{m} \log P(y) \tag{3}$$

ここで、m は入力文の長さであり、n はこれまで出力した記号の長さである．また、λ_{dir}、λ_{ch}、λ_{lm} はそれぞれ、直接モデル、通信路モデル、言語モデルに対する重みを示すハイパーパラメータである．

時間的複雑さは、各部分候補に対して入力全体のスコアリングを繰り返し行うため $\mathcal{O}(k_1 k_2 mn)$ になるが、GPU では並列処理できるため、ほぼ無視できる [3]．ただし、通信路モデルの出力確率の計算的複雑さは $k_1 \times k_2 \times S \times V$ であり、メモリを多く確保するためにデコーディング時のバッチサイズを大幅に小さくする必要がある．ここで、S は入力の最大文長、V は出力の語彙数である．そのため、GPU で並列化できる計算量が少なくなり推論速度が低下する．そこで、先行研究 [4] では語彙数 V を語彙の中で最も頻度の高い部分集合の数を示すハイパーパラメータ V' に置き換えることで計算的複雑さを抑えている．

4　実験

実験データは PTB と CTB を用いる．データの分割および前処理は先行研究 [10] に従う．ただし、頻度が 5 未満の単語を未知語として処理した．データの内訳は表 2 に示す．先行研究 [7, 11] では LSTM モデルを用いているが、本研究では Seq2Seq モデルと言語モデルに数多くのタスクで高い性能を発揮する Transformer モデル [16] を用いる．学習および推論時の設定については付録 A に記載する．

木構造の評価は EVALB[1)] を用いる．今回の構文解析モデルは木構造を構成するためのルールに従わないため、有効な木構造を保証しない．また、EVALB は無効な木構造を評価対象の母数に含めないため、

1）　https://nlp.cs.nyu.edu/evalb/

表 1　木構造（S 式）の線形化の例

文	he had an idea .
S 式	(S (NP (PRP he)) (VP (VBD had) (NP (DT an) (NN idea))) (. .))
LBD	(S (NP XX NP) (VP XX (NP XX XX NP) VP) XX S)
LBP	(S (NP PRP NP) (VP VBD (NP DT NN NP) VP) . S)
LBT	(S (NP he NP) (VP had (NP an idea NP) VP) . S)

表 2　データの内訳

	Train	Valid	Test
PTB	39,832	1,700	2,416
CTB	17,544	352	348

F 値による比較が難しい．ここで、無効な木構造は次の通りである:（1）開括弧と閉括弧の個数が一致しない．（2）終端記号の個数が入力文に含まれる単語数または字面が一致しない．そこで、無効な木構造を評価対象の母数に含め計算した比較可能な F 値（CF: Comparable F-measure）も報告する．

木構造の表現による性能の違いを確認するために、左側に分解した二分木（LLBD、LLBP、LLBT）と右側に分解した二分木（RLBD、RLBP、RLBT）も用いた．さらに二分木変換後に単鎖を結合することで、系列長を $3n$ で統一する．ここで、n は木構造に含まれる単語数である．

5　結果

簡易化のために、直接モデル、通信路モデル、言語モデルをそれぞれ DIR、CH、LM で表す．本研究では、DIR、DIR+LM、DIR+CH+LM という三つのモデルで実験を行う．また、CH の必要性を示すために、DIR+LM の実験も示す．

5.1　定量評価

表 3 と表 4 にそれぞれ PTB と CTB における構文解析の結果を示す．ここで、表 3 と表 4 における CP と CR は比較可能な適合率と再現率を示す．PTB と CTB において、DIR+CH+LM が他のモデルより数多くの有効かつ質の良い木構造を出力することが確認できる．線形化手法について、PTB ではダミー記号による線形化手法（LBD）が、CTB では前終端記号による線形化手法（LBP）が最も良い．二分木については、PTB と CTB において二分木の有効性にはばらつきがあり、言語または Treebank の特徴に依存すると考えられる．

表 3　PTB における実験結果

	モデル	CP	CR	CF1	F1
LBD	DIR	90.72	76.43	82.96	90.50
	DIR + LM	90.85	84.40	87.51	90.53
	DIR + CH + LM	**91.15**	85.62	**88.30**	90.85
LLBD	DIR	89.59	81.46	85.33	89.59
	DIR + LM	89.27	84.67	86.91	89.27
	DIR + CH + LM	90.00	84.64	87.24	90.00
RLBD	DIR	88.55	81.25	84.74	88.55
	DIR + LM	88.77	84.81	86.74	88.77
	DIR + CH + LM	89.25	**85.79**	87.48	89.25
LBP	DIR	90.59	75.47	82.34	90.59
	DIR + LM	90.89	83.54	87.06	91.13
	DIR + CH + LM	90.80	84.14	87.35	**91.16**
LLBP	DIR	90.13	69.96	78.78	90.34
	DIR + LM	90.54	76.96	83.20	90.77
	DIR + CH + LM	90.59	75.98	82.64	90.82
RLBP	DIR	88.71	78.47	83.27	89.30
	DIR + LM	89.75	83.95	86.75	90.24
	DIR + CH + LM	89.98	85.19	87.52	90.53
LBT	DIR	90.29	79.87	84.76	90.48
	DIR + LM	90.38	80.91	85.38	90.86
	DIR + CH + LM	90.20	82.57	86.21	90.79
LLBT	DIR	88.69	65.82	75.56	89.09
	DIR + LM	89.94	70.38	78.96	90.29
	DIR + CH + LM	90.26	70.22	78.99	90.60
RLBT	DIR	86.97	79.18	82.89	87.65
	DIR + LM	88.57	83.39	85.91	89.07
	DIR + CH + LM	88.81	83.48	86.06	89.40

5.2　定性評価

図 1 に LBP 形式の PTB テストセットにおける出力例を示す．DIR および DIR+LM のモデルでは、“all“の品詞または階層的位置の予測に失敗している．また、DIR+LM モデルは、“care of in the music man“を名詞句として予測してしまい、木構造を正しく出力することに失敗している．しかし、DIR+CH+LM は木構造を正しく出力できている．これより、CH が DIR および LM の失敗を修正していることが分かる．

5.3　低資源設定における定量評価

雑音のある通信路モデルが低資源設定の句構造解析において有効であるかを調べるために LBP 形式の PTB および CTB の訓練データを 10%刻みで分割

表 4　CTB における実験結果

	モデル	CP	CR	CF1	F1
LBD	DIR	77.34	40.94	53.54	76.79
	DIR + LM	75.52	45.22	56.56	74.95
	DIR + CH + LM	79.24	50.46	61.65	79.05
LLBD	DIR	35.25	11.29	17.10	35.25
	DIR + LM	36.51	13.86	20.09	36.51
	DIR + CH + LM	33.69	11.63	17.29	33.69
RLBD	DIR	63.34	35.29	45.33	61.64
	DIR + LM	61.09	37.21	46.25	61.09
	DIR + CH + LM	63.34	35.29	45.33	63.34
LBP	DIR	83.15	54.95	66.17	82.42
	DIR + LM	80.83	59.51	68.55	80.73
	DIR + CH + LM	80.32	**64.74**	**71.69**	80.35
LLBP	DIR	74.78	37.43	49.89	74.78
	DIR + LM	79.28	40.53	53.64	79.28
	DIR + CH + LM	77.16	42.85	55.10	77.16
RLBP	DIR	72.82	52.60	61.07	72.83
	DIR + LM	73.96	55.26	63.25	73.96
	DIR + CH + LM	77.00	61.24	68.22	77.00
LBT	DIR	88.06	55.73	68.26	87.20
	DIR + LM	86.67	55.07	67.35	86.91
	DIR + CH + LM	**88.17**	57.16	69.35	**88.10**
LLBT	DIR	83.85	33.03	47.39	83.83
	DIR + LM	85.75	42.42	56.77	85.74
	DIR + CH + LM	87.36	41.43	56.21	87.33
RLBT	DIR	80.48	51.62	62.89	80.46
	DIR + LM	83.33	58.02	68.41	83.31
	DIR + CH + LM	83.48	57.02	67.76	83.46

してモデル学習した．結果を図 2 に示す．低資源設定の PTB と CTB において、DIR+CH+LM の性能が全体的に高いことが確認できる．また、他のモデルでは性能が低下する箇所でも、DIR+CH+LM は頑健に動作しており、低資源設定においても有効であることが確認できる．

6　おわりに

本研究では、低資源かつ高性能な句構造解析器の構築を目的に雑音のある通信路を用いた句構造解析器を提案した．結果、雑音のある通信路を用いた句構造解析器が PTB および CTB、さらには低資源設定においても有効であることを確認した．また、Treebank によって有効な線形化手法が異なることを確認した．今後の課題として、常に正しい木構造を保証するために制約付きデコーディング [11, 15] の導入を試みる．また、NPCMJ（NINJAL Parsed Corpus of Modern Japanese）[2] や Arabic Treebank など実験データの種類を増やし提案手法の有効性を検証する．

2）　https://npcmj.ninjal.ac.jp/interfaces/

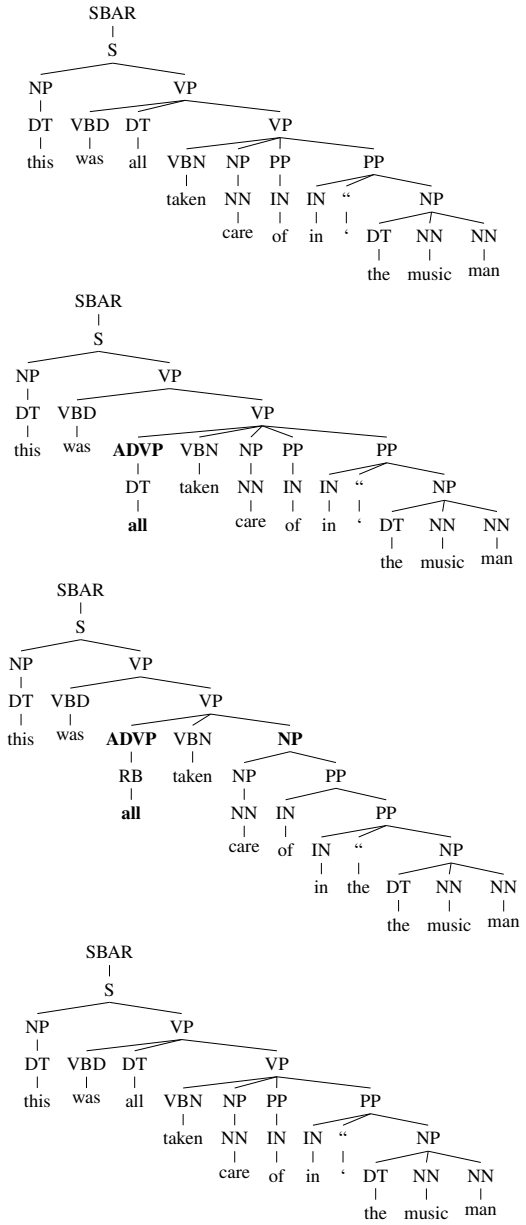

図 1　木構造の比較（一番目：GOLD、二番目：DIR、二番目：DIR+LM、四番目：DIR+CH+LM）

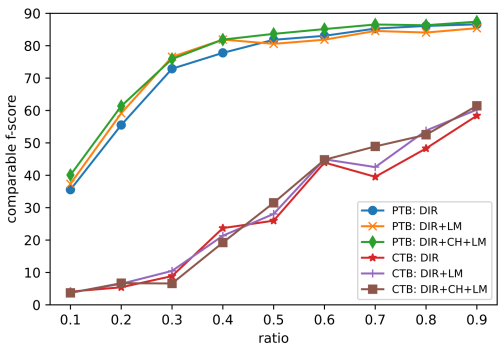

図 2　低資源設定の PTB と CTB における実験結果

謝辞

本研究は JSPS 科研費 JP19K20351 および JP20K23325 の助成を受けたものである．

参考文献

[1] Dan Klein and Christopher D. Manning. Conditional structure versus conditional estimation in nlp models. In **Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - Volume 10**, EMNLP '02, p. 9–16, USA, 2002. Association for Computational Linguistics.

[2] Lei Yu, Phil Blunsom, Chris Dyer, Edward Grefenstette, and Tomás Kociský. The neural noisy channel. **CoRR**, Vol. abs/1611.02554, , 2016.

[3] Kyra Yee, Yann Dauphin, and Michael Auli. Simple and effective noisy channel modeling for neural machine translation. 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. 5696–5701, Hong Kong, China, November 2019. Association for Computational Linguistics.

[4] Shruti Bhosale, Kyra Yee, Sergey Edunov, and Michael Auli. Language models not just for pre-training: Fast online neural noisy channel modeling. In **Proceedings of the Fifth Conference on Machine Translation**, pp. 584–593, Online, November 2020. Association for Computational Linguistics.

[5] Qi Liu, Lei Yu, Laura Rimell, and Phil Blunsom. Pretraining the noisy channel model for task-oriented dialogue. **Trans. Assoc. Comput. Linguistics**, Vol. 9, pp. 657–674, 2021.

[6] Oriol Vinyals, Ł ukasz Kaiser, Terry Koo, Slav Petrov, Ilya Sutskever, and Geoffrey Hinton. Grammar as a foreign language. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, **Advances in Neural Information Processing Systems**, Vol. 28. Curran Associates, Inc., 2015.

[7] Mitchell Stern, Daniel Fried, and Dan Klein. Effective inference for generative neural parsing. In **Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing**, pp. 1695–1700, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.

[8] Daniel Fernández-González and Carlos Gómez-Rodríguez. Enriched in-order linearization for faster sequence-to-sequence constituent parsing. In **Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics**, pp. 4092–4099, Online, July 2020. Association for Computational Linguistics.

[9] Taro Watanabe and Eiichiro Sumita. Transition-based neural constituent parsing. In **Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)**, pp. 1169–1179, Beijing, China, July 2015. Association for Computational Linguistics.

[10] Nikita Kitaev and Dan Klein. Constituency parsing with a self-attentive encoder. In **Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**, pp. 2676–2686, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[11] Jun Suzuki, Sho Takase, Hidetaka Kamigaito, Makoto Morishita, and Masaaki Nagata. An empirical study of building a strong baseline for constituency parsing. In **Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)**, pp. 612–618, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[12] Michael Collins. Three generative, lexicalised models for statistical parsing. In **35th Annual Meeting of the Association for Computational Linguistics and 8th Conference of the European Chapter of the Association for Computational Linguistics**, pp. 16–23, Madrid, Spain, July 1997. Association for Computational Linguistics.

[13] Slav Petrov and Dan Klein. Improved inference for unlexicalized parsing. In **Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference**, pp. 404–411, Rochester, New York, April 2007. Association for Computational Linguistics.

[14] Chris Dyer, Adhiguna Kuncoro, Miguel Ballesteros, and Noah A. Smith. Recurrent neural network grammars. In **Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies**, pp. 199–209, San Diego, California, June 2016. Association for Computational Linguistics.

[15] Daniel Deutsch, Shyam Upadhyay, and Dan Roth. A general-purpose algorithm for constrained sequential inference. In **Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)**, pp. 482–492, Hong Kong, China, November 2019. Association for Computational Linguistics.

[16] 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.

[17] Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. fairseq: A fast, extensible toolkit for sequence modeling. In **Proceedings of NAACL-HLT 2019: Demonstrations**, 2019.

[18] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, **3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings**, 2015.

[19] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. **CoRR**, Vol. abs/1512.00567, , 2015.

表 5　雑音のある通信路モデルのハイパーパラメータ

ビームサイズ k_1	5
ビームサイズ k_2	3
直接モデルに対する重み λ_{dir}	1.0
通信路モデルに対する重み λ_{ch}	0.1
言語モデルに対する重み λ_{lm}	0.2
使用語彙数 V'	100

A　学習設定

A.1　モデルのハイパーパラメータ

Seq2Seq モデルと言語モデルには、それぞれ Fairseq[17] で提供されている `transformer_iwslt_de_en` と `transformer_lm` を用いた．

A.2　学習のハイパーパラメータ

モデルの学習における最適化アルゴリズムには Adam[18] を用いた．目的関数はラベル平滑化交差エントロピー [19] を用い，平滑化パラメータは 0.1 とした．全ての実験において、バッチサイズは 4096 単語とし、4GPU でモデルを学習させた．モデルのパラメータ更新回数は 100,000 回とした．また、学習率は 4000 回更新時で 5e-4 となるように線形的に増加させ、以降は更新回数の平方根の逆数に比例して減衰させた [16].

A.3　推論のハイパーパラメータ

デコーディングに用いたハイパーパラメータを表 5 に示す．ハイパーパラメータは開発用セットにおいて最も性能が良いものを選択した．なお、ビームサイズは 1 から 5 までを 1 刻みで、モデルに対する重みは 0.1 から 1.0 まで 0.1 刻みで検証した．