低頻度語彙埋め込みの縮約による事前学習済みモデルの圧縮

田村 鴻希 [1]　吉永 直樹 [2]　根石 将人 [1]
[1] 東京大学大学院情報理工学系研究科　[2] 東京大学 生産技術研究所
[1]{tamura-k, neishi}@tkl.iis.u-tokyo.ac.jp　[2]ynaga@iis.u-tokyo.ac.jp

概要

既存の事前学習済みモデルの軽量化手法は中間層の圧縮を行うもので，さらなる圧縮には埋め込み層の圧縮が必要となる．本研究では，低頻度語彙の埋め込みを高頻度語彙を用いて縮約表現することで微調整後の事前学習済みモデルの埋め込み層を圧縮する手法を提案する．JNLI，JSTS，JCoLA の JGLUE の各タスクで微調整した事前学習済みモデル DistillBERT に提案手法を適用した結果，それぞれ 3 割程度のパラメタの削減を達成した．

1　はじめに

自然言語処理で標準的に用いられる事前学習済みモデルは，モデルサイズ（パラメタ数）およびその学習（データサイズ）を大規模化することで単調に性能が改善する [1] ことから，近年，肥大化の一途を辿っている．これらのモデルをスマートフォンのような計算資源の限られる環境でも扱うためには，計算量，メモリ使用量の両面でモデルを軽量化することが重要になる．

これに対して，量子化，枝刈り，知識蒸留など，様々なモデル圧縮手法 [2] が研究されているが，これらの多くはモデルの中間層を軽量化するものであり，圧縮されたモデルをさらに小さくするためには入出力層（語彙埋め込み）の軽量化が重要となる．事前学習済みモデルの語彙 [3, 4, 5] は，事前学習時の学習データに対して最適化されているため，単語分布の異なる下流タスクのドメインには出現しない語彙や低頻度となる語彙が含まれる．これらの低頻度語彙の下流タスクの推論への影響は少ないと予想され，単純に削除することも可能ではあるが，一定以上語彙を圧縮しようとすると，単純な削除では性能低下を避けることが難しい．

本研究では，下流タスクの学習データで微調整済みの事前学習済みモデルを対象に，推論への影響が少ないと期待できる，下流タスクの学習データで低頻度となる語彙の埋め込みを高頻度語彙の埋め込で縮約表現することで，推論への影響を抑えて語彙埋め込みを圧縮する．具体的にはまず，下流タスクの学習データを圧縮対象の事前学習済みモデルのトークナイザでトークン化し，各語彙の頻度を数える．次に，圧縮の目標とする語彙サイズを超える頻度順位を縮約対象とする低頻度語彙とし，事前学習済みモデルの埋め込み空間内で k 近傍となる高頻度語彙を検索し，その埋め込みの線形和により，低頻度語彙の埋め込みを縮約表現する（図 1）．

実験では，DistilBERT[6] を対象に，日本語言語理解データセット JGLUE[7] に含まれる下流タスクのデータセット（JNLI, JSTS, JCoLA[8]）を用いて提案手法の有効性を評価した．具体的には，各下流タスクで微調整した DistilBERT に対し，下流タスクの学習データに出現する語彙サイズを超える頻度順位の語彙を縮約表現したときの性能低下を評価した．

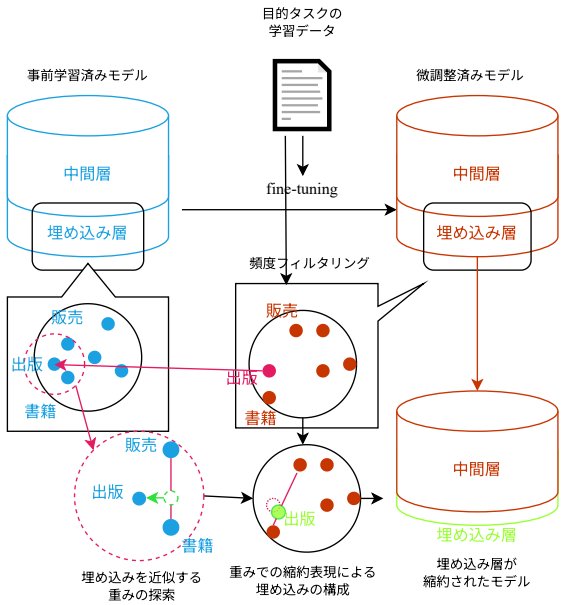

図 1　事前学習済みモデルの低頻度語彙を対象とした縮約表現 (k=2)．モデルは縮約表現された語彙自体は保存せず，埋め込みを再構成するための重みと対応する語彙の ID を保存する．

2 関連研究

本節では，言語モデルを一から学習する際の語彙獲得手法，および学習済みモデルの語彙を目的ドメインに適応させる手法について説明する．

2.1 語彙獲得

Byte-Pair Encoding (BPE) [4] やユニグラム言語モデル [3] など既存のサブワード語彙の獲得手法では，予め定めた語彙サイズ[1)]となるように大規模な学習データからトークナイザを学習し，学習データを分割することで語彙集合を得る．これに対して，Xu ら [9] は決定的に定めた語彙サイズが必ずしも最適とはならないことを指摘し，最適輸送を用いることで，BPE で生成された語彙候補からモデルに最適な語彙サイズと，サイズに合った語彙を選択する手法を提案した．これらの手法はモデルの学習前に語彙集合を獲得するための手法であり，事前学習済みモデルの語彙を変更する用途には使えない．Hiraoka ら [10] は最適な語彙が個別のタスクに依存しうることを指摘し，下流タスクでの学習 (微調整) 時にタスクの損失を考慮して語彙を最適化する手法を提案しているが，これは語彙の圧縮を目的とするものではない．

一方で，サブワードより短い文字 [11, 12] やバイト [13, 14] を入力単位とするモデルも提案されており，バイトを入出力単位とする事前学習済みモデル [13] も存在する．このように極端に短いトークンに基づくモデルは入力のトークン長が長くなるため，推論速度の低下に繋がることから実用性を欠く．また，語彙が持つ情報量の低下を補うため，中間層を深くする必要があることが報告 [13] されており，中間層の軽量化が難しいと予想される．

2.2 語彙適応

また，本研究と関連する分野として，学習済みモデルを他の言語やドメインに適応するために，語彙に注目してその埋め込みを再構成する研究 [15, 16, 17, 18] が存在する．Sakuma ら [15] は，多言語モデルにおいて，原言語の下流タスクのデータを用いて学習したモデルを目的言語で活用するため，多言語埋め込み空間で近接する原言語の単語埋め込みとの位置関係に基づき，原言語のタスク特化単語埋め込み空間における目的言語の単語埋め込みを計算する手法を提案している．Sato ら [16] は，Sakuma らの手法を機械翻訳のドメイン適応に応用し，目的ドメインでの微調整前に学習済みモデルの語彙を目的ドメインの語彙に切り替える手法を提案している．Kajiura ら [17] は，事前学習済みモデルの語彙の一部を，下流タスクでの微調整前にそのタスクに頻出する語彙と入れ替える手法を提案している．Dobler ら [18] は多言語モデルを単言語用に追加学習する際に，下流タスクの学習データから語彙埋め込みを再構成する手法を提案している．これらの研究もやはり，語彙の最適化に主眼があり，学習済みモデルの語彙埋め込みの圧縮は行なっていない．

3 提案手法

本研究では，下流タスクで微調整した事前学習済みモデルを圧縮するため，語彙埋め込みを縮約表現する手法を提案する．具体的には，事前学習済みモデルの語彙 V を縮約対象とする語彙 V_L とその他の語彙 $V_H = V - V_L$ に分け，前者の埋め込みを後者の埋め込みの重み付き線形和で縮約表現することで代替とし，V_L の埋め込みをモデルから破棄することでモデルのパラメタ数を削減する．

本研究では，埋め込みの近似にあたって k 語の語彙の重み付き線形和を用いるため，モデルが直接埋め込みを保持する語彙は V_H のみである．V_L については線形和を構成する語彙の ID と重みのペア k 対を保存する．そのため，モデルの語彙埋め込みに必要となるパラメタ数は，V_L の埋め込みの次元数を d として，$d > 2k$ のとき，$(d - 2k)|V_L|$ 削減される．

3.1 縮約対象とする語彙の選別

縮約表現は近似であるため，縮約による影響を小さくする[2)]ためには，縮約対象とする語彙を適切に選択する必要がある．本研究では，下流タスクの学習データにおける出現頻度に基づき縮約対象とする語彙を決定する．具体的には，V の全ての語彙 t_i について，トークン化された下流タスクの学習データでの出現頻度 f_i を記録する．その後，頻度降順で並べ替えたのち，モデルが定める特殊トークンの集合 $V_{sp} = \{\texttt{<mask>}, \texttt{<sep>}, \texttt{<unk>}, \ldots\}$ を除いたものを V' から，頻度順位で上位 r 語を，縮約に用いる語彙 $V_H = \{t'_i \in V' | i < r\} (|V_H| = r)$ とし，その他の低頻度語彙 $V_L = \{t'_i \in V' | i \geq r\}$ を縮約対象とする．

1) BPE では厳密にはトークンのマージ回数

2) 単純に，全語彙を縮約表現により再構成した場合，下流タスクでの性能低下が大きいことが報告 [19] されている．

3.2 低頻度語彙埋め込みの縮約

埋め込みの縮約によるモデルの圧縮効果を最大限発揮するには，埋め込みの次元数 d に対して，縮約に用いるパラメタ数 $2k$ を十分小さく取ることが望ましい．また，縮約対象となる低頻度語の多くは，下流タスクの学習データで一度も観測されないゼロ頻度の語彙である．そこで，これらの語彙に対して，よりよい埋め込みを縮約表現で与えることを狙って，本研究では Sakuma ら [15] が提案する局所線型写像 (LLM) を用いて低頻度語彙 V_L の各埋め込みの構成を行う．

V_L，V_H 中の各語彙の事前学習済みモデルでの埋め込み集合を Y^{pre}，X^{pre}，微調整済みモデルでの埋め込み集合を Y^{tuned}，X^{tuned} と定義する．微調整タスクの訓練データに直接分割として出現する語彙を V_{train} として，V' 中の語彙の頻度順位が $r(t'_i) > |V_{\text{train}}|$ を満たすゼロ頻度語彙では，定義した Y^{pre} の各埋め込み Y_i^{pre} について，X^{pre} のうちコサイン類似度での上位 k 個の最近傍語彙の集合 $\mathcal{N}_i^{\text{pre}}$，および X_i^{pre} から Y_i^{pre} を近似するための重み $\boldsymbol{\alpha}_i$ を求める．近似構成によって得られる埋め込み集合 $\hat{Y}^{\text{tuned}}$ の各埋め込み $\hat{Y}_i^{\text{tuned}}$ は，微調整済みモデルでの埋め込み集合 X^{tuned} の $\boldsymbol{\alpha}_i$ による重み付け計算 $\sum_{j\in\mathcal{N}_i^{\text{pre}}} \boldsymbol{\alpha}_{ij} X_j^{\text{tuned}}$ によって構成される．また，頻度順位が $r(t'_i) \le |V_{\text{train}}|$ を満たす低頻度語彙では，X^{pre} ではなく X^{tuned} から同様の手順で Y_i^{tuned} を近似する．

重みの計算　埋め込みの構成に用いる重み $\boldsymbol{\alpha}_i$ の計算手法を述べる．LLM では，X_i^{pre} との重み付け平均で Y_i^{pre} を最もよく近似するときの

$$\hat{\alpha}_i = \arg\min_{\alpha_i} \left\| Y_i^{\text{pre}} - \sum_{j\in\mathcal{N}_i^{\text{pre}}} \alpha_{ij} X_j^{\text{pre}} \right\|^2$$

を，Y_i^{pre} と X_i^{pre} を用いたラグランジュの未定乗数法によって

$$\hat{\alpha}_{ij} = \frac{\sum_l (C_i^{-1})_{jl}}{\sum_j \sum_l (C_i^{-1})_{jl}}$$

を

$$C_{ijl} = (Y_i^{\text{pre}} - X_j^{\text{pre}}) \cdot (Y_i^{\text{pre}} - X_l^{\text{pre}})$$

の下で求めることで解く（$l \in \mathcal{N}_i^{\text{pre}}$）．その他の $r(t'_i) \le |V_{\text{train}}|$ の語彙では，X^{pre} の代わりに X^{tuned} を用いて近似の重みを得る．

推論への適応　推論時には微調整済みモデルの V_L の各語彙の埋め込みは，k 近傍語彙の微調整済みモデルでの埋め込み X_i^{tuned} と求めた重み α_i との重み付け平均[3] を用いて算出された $\hat{Y}_i^{\text{tuned}}$ として再構成される．また，元の埋め込み空間 Y^{tuned} は不要となるため，これの破棄によりパラメタを削減する．

4 実験

提案手法による微調整済みモデルの推論性能への影響を確かめるため，提案手法を適用したモデルと，語彙圧縮前のモデルとの性能差を比較する．語彙圧縮のベースラインとしては，縮約対象の語彙を<unk>に置換する手法を用いる．また，提案手法は低頻度語彙を対象とすることから，評価データの中には縮約対象の語彙が出現しない事例も多いため，全評価データを用いた評価に加えて，縮約対象の語彙が出現する事例のみでも実験を行う．

4.1 実験設定

データセット　提案手法によって縮約された語彙埋め込みがモデルの推論性能に与える影響を確かめるため，日本語言語理解データセット JGLUE から，JNLI，JSTS，JCoLA の 3 タスクのデータセットを用いる．それぞれ，JNLI は 2 文の間の含意関係を，JSTS は 2 つの文の意味的な類似度を，JCoLA は文の容認性の可否を推定するタスクである．

事前学習済みモデル　実験では，事前学習済みモデルには LINE ヤフー社により提供されている LINE-distilbert[4] を用い，これを前述の 3 タスクで微調整したモデルを対象として提案手法，およびベースライン手法で語彙埋め込みの縮約を行い，元の語彙埋め込みを持つモデルと各タスクでの推論性能を比較する．モデルのパラメタ数は 68M であり，また語彙サイズは $|V| = 32768$ で，語彙埋め込みのパラメタは 25M であり全体の 36.6%を占める．モデルのトークナイザは，モデルに入力された文章を unidic-lite 辞書を用いた MeCab で分割後，SentencePiece を用いてユニグラム言語モデルによるサブワード単位のトークンに分割する．なお，各タスクでのモデルの微調整では，学習率を 5e-5 として 4 エポックの訓練を行った．

ハイパーパラメタ　低頻度語彙の基準順位 r には，各タスクの訓練データに分割として直接出現する語彙の大きさ $|V_{\text{train}}|$，および $\frac{|V_{\text{train}}|}{2}$ の 2 通りを設定した．前者では訓練データに分割として出現

3)　$\hat{\alpha}_i$ は，$\sum_j \alpha_i = 1$ を満たす
4)　https://huggingface.co/line-corporation/line-distilbert-base-japanese

表 1 LINE-distilbert での提案手法の実験結果 ($r = |V_{train}|$).

	全データセット			置換トークンあり		
	JNLI (4427)	JSTS (4687)	JCoLA (3558)	JNLI (4427)	JSTS (4687)	JCoLA (3558)
<unk>	87.55	83.62	76.50	87.58	85.58	75.05
$k = 1$	87.63	83.60	76.88	88.39	85.48	75.97
$k = 2$	**87.66**	83.61	77.34	**88.75**	85.39	77.10
$k = 3$	87.63	**83.64**	**77.67**	88.39	**85.75**	**77.88**
original ($\|V\| = 32768$)	87.67	83.70	77.90	88.93	86.41	78.49

表 2 提案手法適用時のモデルのパラメタ数.

r	JNLI	JSTS	JCoLA
$\|V_{\text{train}}\|$	47M (69%)	47M (70%)	47M (68%)
$\frac{\|V_{\text{train}}\|}{2}$	46M (67%)	46M (67%)	45M (66%)

表 3 $r = \frac{|V_{\text{train}}|}{2}$ での，全データセットでの実験結果.

	JNLI (2213)	JSTS (2343)	JCoLA (1779)
<unk>	73.87	76.21	69.38
$k = 1$	85.03	80.42	73.11
$k = 2$	84.84	79.92	73.81
$k = 3$	**85.07**	**80.43**	**73.95**

しない語彙が，後者では学習文書に出現する語彙のうち下位 50%が V_L として扱われる．本実験の設定では，JNLI，JSTS，JCoLA の $|V_{\text{train}}|$ は，それぞれ 4427，4687，3558 であった．また，$r = |V_{\text{train}}|$ とする設定では，トークンの構成に用いる埋め込みの数 k は {1,2,3} のそれぞれで実験を行った．

4.2 結果と考察

縮約表現の効果 表 1 に，$r = |V_{\text{train}}|$ の設定での推論精度を示す．提案手法は JSTS での $k = 1, 2$ を除き，V_L の全てのトークンを <unk> トークンに置き換え流手法を精度で上回った．このことから，提案手法によって構成される語彙埋め込みの有効性が確認できた．ただし，提案手法によって構成した語彙埋め込みを用いることで元のモデルを精度で上回ることはなく，提案手法は元の埋め込みに学習された表現を完全に代替するものとはなっていない．

語彙削減の効果 $k = 3$ で提案手法を適用した，各タスクのモデルのパラメタ数と元のモデルからの割合を表 2 に示す．表より，本実験の設定では，$r = |V_{\text{train}}|$ の条件下で，3 つのタスクのそれぞれで 3 割程度のパラメタを削減する．

近傍に用いる語彙の数による影響 $r = \frac{|V_{\text{train}}|}{2}$ で，$r = |V_{\text{train}}|$ と同様に推論を行った結果を表 3 に示す．表より，全ての設定において推論性能は $r = |V_{\text{train}}|$ の <unk> トークンによる置き換えを下回っており，学習時に直接出現したトークンの埋め込みが各タスクに与える影響はその他のトークンの表現が及ぼす影響に比べて強いことを確認した．また，近傍トークンを利用する場合と <unk> トークンに置き換える場合の性能差が $r = |V_{\text{train}}|$ に比べて大きいことから，提案手法は学習時に出現したトークンに対しても有効な近似埋め込みを構成しているといえる．

用いる埋め込みの数による影響 実験では，用いる近傍埋め込みの数 k を 1 から 3 まで変化させたが，k が大きいほど近傍トークンは置き換えるトークンをよく近似する．しかし，$r = |V_{\text{train}}|$ の JNLI では $k = 3$ ではなく $k = 2$ の方が性能が高いように，k が大きいほどよい埋め込みを構成するとは限らない．この原因としては，モデルが中間層での計算の過程で近い語彙埋め込みの間の意味の違いを判別し，出力埋め込みでは離れた表現とすることが考えられる．

5 まとめ

本研究では，微調整済み言語モデルの語彙圧縮を目的として，高頻度語彙の埋め込みを用いて低頻度語彙埋め込みを縮約表現する手法を提案した．提案手法では，低頻度語彙の埋め込みを，事前学習済みモデルの語彙空間における近傍の高頻度語彙の埋め込みの重み付け線形和によって近似する．実験では，本手法で埋め込みを置き換えたモデルを用いて JGLUE データセット上で推論を行い，事前学習済みモデル DistilBERT に対して 3 割程度のパラメタの削減，および <unk> トークンと比較して性能の高い埋め込みを近似した．

今後の課題として，下流タスクでの微調整時に同時に語彙を縮約する手法を検討する．

謝辞

本研究は東京大学生産技術研究所特別研究経費および JSPS 科研費 JP21H03494 の助成を受けています.

参考文献

[1] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models, 2020.

[2] Yu Cheng, Duo Wang, Pan Zhou, and Tao Zhang. Model compression and acceleration for deep neural networks: The principles, progress, and challenges. **IEEE Signal Processing Magazine**, Vol. 35, No. 1, pp. 126–136, 2018.

[3] Taku Kudo. Subword Regularization: Improving Neural Network Translation Models with Multiple Subword Candidates. In **Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**, pp. 66–75, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[4] 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.

[5] Mike Schuster and Kaisuke Nakajima. Japanese and Korean voice search. In **2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)**, pp. 5149–5152, March 2012.

[6] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter, February 2020.

[7] Kentaro Kurihara, Daisuke Kawahara, and Tomohide Shibata. JGLUE: Japanese General Language Understanding Evaluation. In **Proceedings of the Thirteenth Language Resources and Evaluation Conference**, pp. 2957–2966, Marseille, France, June 2022. European Language Resources Association.

[8] Taiga Someya, Yushi Sugimoto, and Yohei Oseki. Jcola: Japanese corpus of linguistic acceptability. 2023.

[9] Jingjing Xu, Hao Zhou, Chun Gan, Zaixiang Zheng, and Lei Li. Vocabulary Learning via Optimal Transport for Neural Machine Translation. In **Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)**, pp. 7361–7373, Online, August 2021. Association for Computational Linguistics.

[10] Tatsuya Hiraoka, Sho Takase, Kei Uchiumi, Atsushi Keyaki, and Naoaki Okazaki. Optimizing Word Segmentation for Downstream Task. In **Findings of the Association for Computational Linguistics: EMNLP 2020**, pp. 1341–1351, Online, January 2020. Association for Computational Linguistics.

[11] Jonathan H. Clark, Dan Garrette, Iulia Turc, and John Wieting. Canine : Pre-training an Efficient Tokenization-Free Encoder for Language Representation. **Transactions of the Association for Computational Linguistics**, Vol. 10, pp. 73–91, January 2022.

[12] Yi Tay, Vinh Q. Tran, Sebastian Ruder, Jai Gupta, Hyung Won Chung, Dara Bahri, Zhen Qin, Simon Baumgartner, Cong Yu, and Donald Metzler. Charformer: Fast Character Transformers via Gradient-based Subword Tokenization, February 2022.

[13] Linting Xue, Aditya Barua, Noah Constant, Rami Al-Rfou, Sharan Narang, Mihir Kale, Adam Roberts, and Colin Raffel. ByT5: Towards a Token-Free Future with Pre-trained Byte-to-Byte Models. **Transactions of the Association for Computational Linguistics**, Vol. 10, pp. 291–306, 2022.

[14] Changhan Wang, Kyunghyun Cho, and Jiatao Gu. Neural Machine Translation with Byte-Level Subwords. **Proceedings of the AAAI Conference on Artificial Intelligence**, Vol. 34, No. 05, pp. 9154–9160, April 2020.

[15] Jin Sakuma and Naoki Yoshinaga. Multilingual Model Using Cross-Task Embedding Projection. In Mohit Bansal and Aline Villavicencio, editors, **Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)**, pp. 22–32, Hong Kong, China, January 2019. Association for Computational Linguistics.

[16] Shoetsu Sato, Jin Sakuma, Naoki Yoshinaga, Masashi Toyoda, and Masaru Kitsuregawa. Vocabulary adaptation for domain adaptation in neural machine translation. In Trevor Cohn, Yulan He, and Yang Liu, editors, **Findings of the Association for Computational Linguistics: EMNLP 2020**, pp. 4269–4279, Online, November 2020. Association for Computational Linguistics.

[17] Teruno Kajiura, Shiho Takano, Tatsuya Hiraoka, and Kimio Kuramitsu. Vocabulary replacement in sentencepiece for domain adaptation. In **Proceedings of the 37th Pacific Asia Conference on Language, Information and Computation**, 2023.

[18] Konstantin Dobler and Gerard de Melo. FOCUS: Effective Embedding Initialization for Monolingual Specialization of Multilingual Models. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, **Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing**, pp. 13440–13454, Singapore, February 2023. Association for Computational Linguistics.

[19] Nobukazu Fukuda, Naoki Yoshinaga, and Masaru Kitsuregawa. Robust Backed-off Estimation of Out-of-Vocabulary Embeddings. In Trevor Cohn, Yulan He, and Yang Liu, editors, **Findings of the Association for Computational Linguistics: EMNLP 2020**, pp. 4827–4838, Online, November 2020. Association for Computational Linguistics.