言語モデルと解析戦略の観点からの修辞構造解析器の比較

小林 尚輝 † 平尾 努 § 上垣外 英剛 † 奥村 学 † 永田 昌明 §
† 東京工業大学 §NTT コミュニケーション科学基礎研究所
{kobayasi, kamigaito, oku}@lr.pi.titech.ac.jp
{tsutomu.hirao.kp, masaaki.nagata.et}@hco.ntt.co.jp

概要

修辞構造解析の研究分野では，ニューラルモデルの発展により多くの手法が提案され，ベンチマークデータでの最高スコアが日々更新されている．しかし，提案された手法は解析戦略，EDU 系列のベクトル表現のための事前学習済み言語モデル，探索アルゴリズムなどが異なっており，どの要素が良い結果をもたらしたのかが分かりにくい．そこで，本稿では技術の進歩を明確にできるよう，既存の上向き，下向きの解析戦略と最新の事前学習済み言語モデルを組み合わせることで強いベースライン解析器を構築する．この解析器を RST-DT で評価した結果，解析戦略には大きな差がなく，トークンではなくスパンのマスキングを採用した事前学習済み言語モデルが有効であることが分かった．特に，DeBERTa を用いると世界最高性能を達成した．今後，このベースラインを基準とすることで新しい技術の有効性をより明確にできると考える．

1 はじめに

修辞構造理論 [1] は，文書の背後にある構造を，EDU と呼ばれる節相当のユニットを終端ノード，単一もしくは連続した EDU からなるスパンを非終端ノードとする構成素木として表現する．非終端ノードには核 (Nucleus) または衛星 (Satellite) の核性ラベルが与えられ，同じ親を持つ兄弟の非終端ノード間のエッジにはその関係を表す修辞ラベルが与えられる．

修辞構造解析では，上向き，下向きの解析戦略のどちらかが利用される．上向き解析には主にシフト還元法を用いた手法 [2, 3]，下向き解析にはスパン分割法を用いた手法 [4, 5] やエンコーダデコーダモデルを用いた手法 [6] がある．これら 2 つの解析戦略において，近年ニューラルネットワークが活用され，さらなる性能向上のため，ビーム探索 [7]，動的オラクルを用いた探索 [5]，モデルアンサンブル [4]，敵対学習 [6] など様々な技術が導入されている．

さらに，スパンのベクトル表現を得るために事前学習済み言語モデルが利用されることからその性能向上が修辞構造解析の性能改善にもつながる．たとえば，XLNet[8] を用いた下向き解析法 [6] と SpanBERT[9] を用いた上向き解析法 [3] は従来法よりも大幅に性能が向上している．

このようにニューラルモデルを用いた修辞構造解析手法の性能は大きく向上しているものの，各手法が採用した解析戦略，事前学習済み言語モデル，探索アルゴリズムなどが異なるため，どの要素が解析器の性能向上にどの程度寄与したかが分かりにくく，技術の進歩を正しく評価することが難しい．

本稿は，既存の上向き，下向き解析戦略に対し，5 種の事前学習済み言語モデル[1)]を組み合わせるだけでどの程度の性能が得られるかを検証し，技術の進歩を明らかにするためのベースラインを構築することを目的とする．RST-DT を用いた実験結果から，(1) 2 つの解析戦略の間に大きな差はないこと，(2) トークンではなくスパンのマスキングを利用した目的関数で学習された事前学習済み言語モデルが有効であること，(3) DeBERTa を利用すると現在の世界最高性能を達成すること，が分かった．さらに，チェックポイント重み平均の導入や事前学習済み言語モデルのパラメタを増やすことでも性能の改善が得られることが分かった．

2 解析手法

本稿では，上向き解析戦略を用いた解析器としてシフト還元法を用いた手法 [3]，下向き解析戦略を用いた解析器として再帰的なスパン分割による手法 [4] を採用した．その理由はどちらもシンプルな方法であり，かつ実装が公開されているからである．

1) 本稿では，トランスフォーマに基づく事前学習済み言語モデルを対象とする．

双方とも単一もしくは連続した EDU からなるスパンをベクトルにより表現する必要がある．以下，2.1 節でスパンのベクトル表現の獲得方法について説明する．次に 2.2 および 2.3 節においてそれぞれの解析手法を説明する．

2.1 スパンのベクトル表現の獲得方法

N 個の EDU からなる文書が与えられた時，文書をトークン系列とみなし，事前学習済み言語モデルを通すことでトークンのベクトル表現 $\{\boldsymbol{w}_1, \ldots, \boldsymbol{w}_M\}$ を得る．ここで M は文書に含まれる全トークン数とする．M が 512 を超える場合には，スライド窓を利用してすべてのトークンのベクトル表現を得る．

あるスパンが i 番目から j 番目の EDU を内包する時（ただし，$1 \leqq i < j \leqq N$），このスパンを表現するベクトル表現 $\boldsymbol{u}_{i:j}$ はスパンの両端のトークンのベクトル表現から $\boldsymbol{u}_{i:j} = (\boldsymbol{w}_{\mathrm{b}(i)} + \boldsymbol{w}_{\mathrm{e}(j)})/2$ として求める．ここで $\mathrm{b}(i)$ および $\mathrm{e}(j)$ はそれぞれ引数で指定された EDU の先頭および末尾のトークンのインデックスを返す関数である．

2.2 上向き解析法

上向き解析器として Guz ら [3] のシフト還元法を用いた解析器を採用する．スタック S に解析済みの部分木を格納，キュー Q にこれから解析対象となる EDU を格納し，以下のシフト，還元操作を適用することで，左から右に向かって EDU 系列を読み込みながら上向きに修辞構造木を構築する．

シフト キュー Q の先頭の EDU を取り出し，スタックに積む，

還元 スタック S の上 2 つの部分木を取り出しそれらを併合することで 1 つの木を構築し，再度スタック S に積む．

なお，還元操作を行った後，2 つの部分木に対し核性ラベル，修辞関係ラベルを，異なる分類器を用いて推定する．つまり，木の構築，核性ラベル推定，修辞関係ラベル推定は独立に行う[2]．核性ラベルは N-S，S-N，N-N の 3 種のいずれか，修辞関係ラベルは Elaboration，List など 18 種のいずれかである．それぞれの推定は以下の順伝播型ニューラルネットワーク $\mathrm{FFN}_{\mathrm{act}}$，$\mathrm{FFN}_{\mathrm{nuc}}$，$\mathrm{FFN}_{\mathrm{rel}}$ を用いて行う．

$$s_* = \mathrm{FFN}_*(\mathrm{Concat}(\mathbf{u}_{\mathrm{s}_0}, \mathbf{u}_{\mathrm{s}_1}, \mathbf{u}_{\mathrm{q}_0}, \mathbf{u}_{\mathrm{org}})), \tag{1}$$

2） Guz ら [3]，Wang ら [2] は，木の構築と核性ラベル推定は同時に行ったほうが良いと報告しているが，我々の実験では独立に行ったほうが性能が良かった．

ここで，$\mathbf{u}_{\mathrm{s}_0}$，$\mathbf{u}_{\mathrm{s}_1}$ はそれぞれ S の上 2 つの部分木が支配するスパンのベクトル表現，$\mathbf{u}_{\mathrm{q}_0}$ は Q の先頭の EDU のベクトル表現，$\mathbf{u}_{\mathrm{org}}$ は，S の上 2 つのスパンが同じ段落/文に存在するか，隣接する段落/文に存在するか，S の一番上のスパンと Q の先頭の EDU が同じ段落/文に存在するか，それぞれのスパンが文書/段落/文の先頭/末尾であるかを 2 値ベクトルとして表現したものである．

2.3 下向き解析法

下向き解析器として Kobayashi ら [4] の再帰的スパン分割法を用いた解析器を採用する．この解析器は，文書全体をあらわすスパンから始め貪欲に 2 分割していき，分割したスパン間のラベルを推定することで下向きに修辞構造木を構築する．なお，Kobayashi らの手法は文書-段落，段落-文，文-EDU という 3 つの階層で独立に解析を行った後それらを統合することで木を構築するが，上向き解析器と近い状態で比較するため，本稿では，階層化は行わず上向き解析器と同様にスパンがどの階層に存在するかを 2 値ベクトルとして表現して利用する．i 番目から j 番目の EDU からなるスパンを k 番目の EDU で分割するスコアは以下の式で定義される．

$$\begin{aligned} s_{\mathrm{split}}(i,j,k) = \mathbf{h}_{i:k}\mathbf{W}\mathbf{h}_{k+1:j} + \mathbf{v}_{\mathrm{left}}\mathbf{h}_{i:k} + \mathbf{v}_{\mathrm{right}}\mathbf{h}_{k+1:j} \\ + \mathbf{v}_{\mathrm{org}}\mathbf{h}_{\mathrm{org}}, \end{aligned} \tag{2}$$

ここで，$\mathbf{W}$ は重み行列，$\mathbf{v}_{\mathrm{left}}$, $\mathbf{v}_{\mathrm{right}}$ $\mathbf{v}_{\mathrm{org}}$ は重みベクトル，$\mathbf{h}_{i:k}$, $\mathbf{h}_{k+1:j}$, $\mathbf{h}_{\mathrm{org}}$ は以下の式で定義される．

$$\mathbf{h}_{i:k} = \mathrm{FFN}_{\mathrm{left}}(\mathbf{u}_{i:k}), \tag{3}$$

$$\mathbf{h}_{k+1:j} = \mathrm{FFN}_{\mathrm{right}}(\mathbf{u}_{k+1:j}), \tag{4}$$

$$\mathbf{h}_{\mathrm{org}} = \mathrm{FFN}_{\mathrm{org}}(\mathbf{u}_{\mathrm{org}}), \tag{5}$$

ここで，$\mathbf{h}_{\mathrm{org}}$ は上向き解析と同様，左右のスパンが同じ段落/文に存在するか，文/段落の開始か終了などを表す 2 値ベクトルである．そして，以下の式でスパンの分割を決定する．

$$\hat{k} = \underset{i \leq k < j}{\mathrm{argmax}}\, s_{\mathrm{split}}(i,j,k). \tag{6}$$

分割と同様に，スパンのラベルを推定するスコアは以下の式で定義される．

$$\begin{aligned} s_{\mathrm{label}}(i,j,\hat{k},\ell) = \mathbf{h}_{i:\hat{k}}\mathbf{W}^{\ell}\mathbf{h}_{\hat{k}+1:j} + \mathbf{v}^{\ell}_{\mathrm{left}}\mathbf{h}_{i:\hat{k}} + \mathbf{v}^{\ell}_{\mathrm{right}}\mathbf{h}_{\hat{k}+1:j} \\ + \mathbf{v}^{\ell}_{\mathrm{org}}\mathbf{h}_{\mathrm{org}}, \end{aligned} \tag{7}$$

ここで，$\mathbf{W}^{\ell}$ は特定のラベル ℓ に対応する重み行列，$\mathbf{v}^{\ell}_{\mathrm{left}}$, $\mathbf{v}^{\ell}_{\mathrm{right}}$, $\mathbf{v}^{\ell}_{\mathrm{org}}$ はそれぞれ ℓ に対応する重みベクト

ルである．そして，以下の式を最大にするラベル $\hat{\ell}$ を，$\hat{k}$ で分割した 2 つのスパンに対するラベルとする．上向き解析器と同様，核性推定時は，N-S，S-N，N-N のいずれか，修辞関係推定時には 18 種のラベルのいずれかを与える．

$$\hat{\ell} = \underset{\ell \in \mathcal{L}}{\operatorname{argmax}}\, s_{\text{label}}(i, j, \hat{k}, \ell). \tag{8}$$

3 事前学習済み言語モデル

本稿ではスパンのベクトル表現を得るために，以下に説明するトランスフォーマに基づく 5 種の事前学習済み言語モデルを試し，性能を比較する．

BERT[10] はランダムにマスクされたトークンを前後の文脈から推定する Masked Language Model (MLM) と 2 つの文が連続しているかどうかを推定する Next Sentence Prediction (NSP) の 2 つの目的関数により学習される双方向型の言語モデルであり，学習には 13GB のテキストが用いられる．

RoBERTa[11] は BERT を MLM のみで，より大きなテキスト (160GB) を用いて長時間学習することでさらに性能を向上させた言語モデルである．

XLNet[8] は単方向言語モデルであるが，attention のマスクを改善し擬似的に語順を入れ替え，双方向言語モデルと同様に前後の文脈を考慮できる Permuted Language Model (PLM) を目的関数としている．

SpanBERT[9] は BERT と同じデータセットで学習されるが，MLM のみを学習の目的関数とし，マスクする単位を個々のトークンではなく連続するトークン (スパン) へと変更している．

DeBERTa[12] は各トークンの埋め込みとその位置情報の埋め込みの関係をより強く学習するために Distangled attention を導入した．さらに，SpanBERT と同様，マスクの単位は個々のトークンではなく連続したトークンである．DeBERTa の学習に用いるコーパスは RoBERTa よりも小さく 78GB であるが様々なベンチマークで RoBERTa を上回る性能を達成している．

4 実験設定

本稿では RST-DT[13] をベンチマークとして実験を行った．RST-DT は Wall Street Journal から収集されており，学習データ 347 文書，テストデータ 38 文書から構成される．開発データは Heilman ら [14] に従い学習データのうちの 40 文書とした．過去の研究にならい修辞構造ラベルには 18 種類からなる大分類を用い，修辞構造木は二分木へと事前に変換して解析器の学習，評価を行う．

本稿では正解の EDU 分割を用いるため，先行研究 [15] に従い Standard Parseval を用いて評価する．木の構造のみを評価する Span，核性または修辞関係ラベルも含めて評価する Nuc. および Rel.，双方のラベルも含めて評価する Full の計 4 通りの評価尺度により評価する．

上向き，下向きの両解析法に使用した FFN は，活性化関数に GeLU を用い，隠れ状態の次元数が 512 次元の 2 層の MLP とした．過学習を抑制するために，FFN の dropout 率を 0.2，勾配クリッピングの値を 1.0，L2 正則化の係数を 0.01 とした．ミニバッチは文書ではなくスパン（上向き解析であればシフトまたは還元の操作，下向き解析であればあるスパンの分割）を単位に構成し，サイズは 5 とした．パラメータは AdamW[16] で最適化し，言語モデルの学習率は 1e-5，それ以外に関しては 1e-5/2e-5 から開発データを用いて選択した．また，学習率スケジューラを用いて，1 エポック目は学習率を 0 から増加させ，それ以降は学習率を 0 へと線形に減少させた．最大のエポック数は 20 としたが，エポック毎の開発データによる Full の評価値が最高値を 5 回連続で下回った場合は早期に学習を打ち切った．

比較対象としては，RST-DT における最高性能を達成した Zhang らの方法 [6] を採用する．この手法は，エンコーダ・デコーダモデルに基づく下向き解析戦略，XLNet を利用したスパンのベクトル表現，敵対学習を用いたパラメタ最適化法を利用しており，本稿で提案する手法よりもパラメタが多く複雑な手法である．

5 実験結果と考察

実験結果を表 1 に示す．提案法を解析戦略で比較すると Span の場合，下向きが上向きよりも良い傾向にあるが，その差はわずかである．Rel の場合，上向きが下向きよりも良い傾向にあり，Full で比較するとやや上向きが良いがほぼ同等といって良い性能である．

事前学習済み言語モデルによる性能差を比較すると双方の解析戦略とも BERT が最も悪く，DeBERTa が最も良い．特に BERT を用いた場合，他の事前学習済み言語モデルを利用したときより 5 ポイント以上の性能劣化がみられる．Koto ら [5] も BERT

		Span	Nuc.	Rel.	Full
Zhang ら [6]		76.3	65.5	55.6	53.8
上向き	BERT	72.2	60.1	49.5	47.4
	RoBERTa	77.2	67.0	56.6	54.5
	XLNet	77.2	67.0	56.9	54.9
	SpanBERT	77.2	66.6	55.2	53.1
	DeBERTa	**79.0**	**68.9**	**57.5**	**55.2**
下向き	BERT	72.1	60.0	49.1	46.9
	RoBERTa	78.5	67.2	56.0	53.0
	XLNet	78.3	67.0	56.9	54.4
	SpanBERT	77.1	65.7	54.5	52.2
	DeBERTa	**79.0**	68.4	**57.5**	55.1

表 1　解析戦略および言語モデルの比較．各指標で最も高い数値を太字は示す．スコアは 3 回の試行の平均値である．

を利用すると GloVe[17] よりも性能が劣化することを報告しており，この結果と合致する．BERT は 13G のデータで学習されており，今回用いた事前学習済み言語モデルのなかでは SpanBERT とともに最も訓練データのサイズが小さい．事前学習済み言語モデルの訓練に使ったデータのサイズと解析性能が相関するのであれば RoBERTa が最も良いはずである．しかし，それよりも訓練データサイズの小さい XLNet，DeBERTa の方が成績が良いこと，BERT と同じ学習データを用いているにもかかわらず SpanBERT が RoBERTa よりもやや劣る程度の性能を達成していることからデータサイズだけでなく，事前学習済み言語モデルの学習時の目的関数も重要であることが分かる．XLNet は PLM という語順を意識した特殊な目的関数を採用しておりこれが有効であったと考える．一方，SpanBEERT がデータサイズが小さいにもかかわらず RoBERTa に匹敵する性能を達成したこと，DeBERTa が最も良い性能を達成したことを勘案すると MLM において，トークンではなく連続したトークン，スパンをマスキングすることが有効であると考える．

Zhang らの手法と比較すると，BERT を除くほぼすべての手法が上回っており，DeBERTa を用いると Zhang らの手法よりも 2 ポイント近いゲインが得られている．つまり，事前学習済み言語モデルを適切に選択すれば現存する単純な手法であっても世界最高性能を達成できることを示している．Zhang らの手法は我々の手法よりも高度な技術を採用しているが，なぜこの程度の性能しかでていないのかは不明である．

我々の手法をベースラインとして，我々の手法の上にパラメタ最適化の新しい技術，新しい探索アルゴリズムなどを導入すると，高い解析性能を保ちながら新しい技術の貢献を評価できるようになると考える．

		Span	Nuc.	Rel.	Full
上向き	DeBERTa	79.0	68.9	57.5	55.2
	+CWA	79.3	69.6	58.3	56.1
	w/ large	79.3	69.1	58.4	56.0
	both	80.0	70.4	60.0	57.7
下向き	DeBERTa	79.0	68.4	57.5	55.1
	+CWA	80.2	69.9	58.3	56.2
	w/ large	80.1	69.7	58.5	56.3
	both	81.0	71.2	60.5	58.3

表 2　チェックポイント重み平均 (CWA) とパラメタを増やした事前学習済み言語モデル (large) を導入した場合の評価結果．スコアは 3 回の試行の平均値である．

さらに，簡単な技術を用いてさらなる性能向上が可能かを，チェックポイント重み平均 [18, 19] の導入，事前学習済み言語モデルのパラメタを増加させることで検証した．DeBERTa を利用した上向き，下向き解析器に双方の技術を組み込んだ結果を表 2 に示す．それぞれの技術を導入すると Full で約 1 ポイント程度の性能向上が得られるが，双方を同時に導入すると 2.5 ポイント以上もの性能向上が得られる．最終的には上向きで Zhang らの手法より 4 ポイント，下向きで 4.5 ポイントの大きなゲインが得られている．つまり，我々の方法にはまだまだ改善の余地が残っており，新しい技術でさらなる性能向上が見込めることを示している．

6　まとめ

本稿では，技術の進歩を明らかにするためのベースライン修辞構造解析器を構築するため，既存の上向き，下向き解析戦略に対し，5 種の事前学習済み言語モデルを組み合わせることでどの程度の性能が得られるかを検証した．その結果，解析戦略とは関係なく，RoBERTa，XLNet，DeBERTa を用いると既存の解析器を上回る性能が得られた．特に，SpanBERT が訓練データ量が少ないにも関わらず良い性能を達成したこと，DeBERTa が最も良い成績であったことから，MLM においてトークンではなくスパンのマスキングが有効であることが分かった．チェックポイント重み平均の導入や事前学習済み言語モデルのパラメタを増やすことでさらに性能が向上することも分かった．

謝辞

本研究の一部は JSPS 科研費 JP21H03505 の助成を受けたものです.

参考文献

[1] W.C. Mann and S.A Thompson. Rhetorical structure theory: A theory of text organization. Technical Report ISI/RS-87-190, USC/ISI, 1987.

[2] Yizhong Wang, Sujian Li, and Houfeng Wang. A two-stage parsing method for text-level discourse analysis. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 184–188, Vancouver, Canada, July 2017. Association for Computational Linguistics.

[3] Grigorii Guz and Giuseppe Carenini. Coreference for discourse parsing: A neural approach. In *Proceedings of the First Workshop on Computational Approaches to Discourse*, pp. 160–167, Online, November 2020. Association for Computational Linguistics.

[4] Naoki Kobayashi, Tsutomu Hirao, Hidetaka Kamigaito, Manabu Okumura, and Masaaki Nagata. Top-down rst parsing utilizing granularity levels in documents. *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34, No. 05, pp. 8099–8106, Apr. 2020.

[5] Fajri Koto, Jey Han Lau, and Timothy Baldwin. Top-down discourse parsing via sequence labelling. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pp. 715–726, Online, April 2021. Association for Computational Linguistics.

[6] Longyin Zhang, Fang Kong, and Guodong Zhou. Adversarial learning for discourse rhetorical structure parsing. 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. 3946–3957, Online, August 2021. Association for Computational Linguistics.

[7] Ke Shi, Zhengyuan Liu, and Nancy F. Chen. An end-to-end document-level neural discourse parser exploiting multi-granularity representations. 2020.

[8] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, Vol. 32. Curran Associates, Inc., 2019.

[9] Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. SpanBERT: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, Vol. 8, pp. 64–77, 2020.

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[11] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *ArXiv*, Vol. abs/1907.11692, , 2019.

[12] Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*, 2021.

[13] Mary Ellen Okurowski Lynn Carlson, Daniel Marcu. *RST Discourse Treebank*. Philadelphia: Linguistic Data Consortium, 2002.

[14] Michael Heilman and Kenji Sagae. Fast rhetorical structure theory discourse parsing. *CoRR*, Vol. abs/1505.02425, , 2015.

[15] Mathieu Morey, Philippe Muller, and Nicholas Asher. How much progress have we made on RST discourse parsing? a replication study of recent results on the RST-DT. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 1319–1324, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.

[16] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

[17] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543, 2014.

[18] 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 I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, Vol. 30. Curran Associates, Inc., 2017.

[19] Hugh Chen, Scott Lundberg, and Su-In Lee. Checkpoint ensembles: Ensemble methods from a single training process. 2017.