Vol.2019-NL-243 No.17
2019/12/5

階層的な注意機構に基づき統語的な先読みを行う文圧縮

上垣外 英剛[1,a)]　奥村 学[1,b)]

概要：Sequecen-to-Sequence（Seq2Seq）に基づく文圧縮では，原文中の単語を圧縮文に採用するか否かをデコーダが単方向に逐次決定していくため，すでに圧縮文に採用された単語列とこれから圧縮文に採用しようとする単語との間の文法的な依存関係を明示的に捉えることが難しい．このため，Seq2Seq では，重要語の親が既に誤って削除されている場合に，本来圧縮文に採用すべきこのような重要語を文法性の観点から過剰に削除してしまう．この問題を解決するため，本研究ではデコード時に階層的な注意機構に基づき統語的な先読みを行うことが可能な Seq2Seq を提案する．実験では Google sentence compression データセットを使用し，提案手法が F1，ROUGE-1，ROUGE-2，および ROUGE-L スコアにおいてそれぞれ 85.5，79.3，71.3，および 79.1 のスコアを達成し，既存手法に対する改善を確認した．特に長文に対しての改善が顕著であった．さらに，人手評価によって，提案手法が可読性を損なうことなく重要情報を保持した圧縮が可能であることを確認した．

1. はじめに

文圧縮とは，重要な情報を保持しつつ，冗長な単語を削除することにより長い文を短い文へと圧縮するタスクである．今まで多くの研究において文法的に正しい圧縮文を得るために，構文解析木の刈り込みを行う手法 [1], [2], [3], [4] が使用されてきた．だが，これらの手法は，構文解析器の誤りの影響を受けやすくなるという問題がある．Fillippova ら [5] は，構文解析器の誤りの影響を回避するために，構文解析結果に依存することなく，流暢な圧縮文を生成することが可能な Seqeunce-to-Sequence（Seq2Seq）モデルに基づく手法を提案した．しかし，通常の Seq2Seq モデルには，長い文を圧縮する際に精度が低下してしまうという問題が存在する．

この問題を解決するために，Kamigaito ら [6] は Seq2Seq モデルを拡張し，ある単語の依存構造木上の親を再帰的に辿ることにより，遠く離れた単語間の関係を捉えることが可能となる，再帰的な注意機構を提案している．この手法では，構文解析誤りの影響を回避するために，文圧縮と依存構造の情報は同時に学習される．これらの特徴により，再帰的な注意機構を用いた Seq2Seq は，流暢性を損なうことなく，重要な単語を維持した圧縮文を出力可能である．

ただし，この方法では依存構造木中で親となる単語のみを辿るため，これからデコードされる重要語が既にデコードされた単語の子である場合，その存在を事前に捉えることが難しい．Seq2Seq では，デコーダが一方向に文を圧縮していくため，通常，デコードされた単語と将来デコードされる単語との関係を明示的に捉えることができない．その結果，非文法的な文の生成を避けるために，デコーダは文の圧縮過程で重要な単語を過剰に削除してしまうことがある．この問題を解決するためには，依存構造木中の親と子の両方の単語を辿ることにより，将来の時刻においてデコードされる重要な単語をデコーダが事前に捉える必要がある．

図 1 は，圧縮の際に依存構造木上の親と子の両方を辿ることが重要な文の例を示している[*1] この文は 2 か国間の飛行機の輸出入についての内容を記述しているため，圧縮文中には重要な情報である飛行機，輸入国，および輸出国の名称が含まれていることが望ましい．

図 1 の上段の例において，デコーダーは文中の単語 “Japan” をデコードする際に，再帰的に “Japan” の親と子の依存関係を辿ることにより，“Japan” の親であり，かつこの文を代表する主辞である “hold” を圧縮文に含めることを事前に決定できる．文中の単語 “hold” を圧縮文に含めることにより，デコーダはさらに “hold” の子と孫である “Japan”，“and”，“India” を可読性を損なうことなく圧

[1] 東京工業大学　科学技術創成研究院　未来産業技術研究所
[a)] kamigaito@lr.pi.titech.ac.jp
[b)] oku@pi.titech.ac.jp

[*1] この文は実際に Google sentence compression データセットのテストデータに含まれているものを掲載している．(https://github.com/google-research-datasets/)

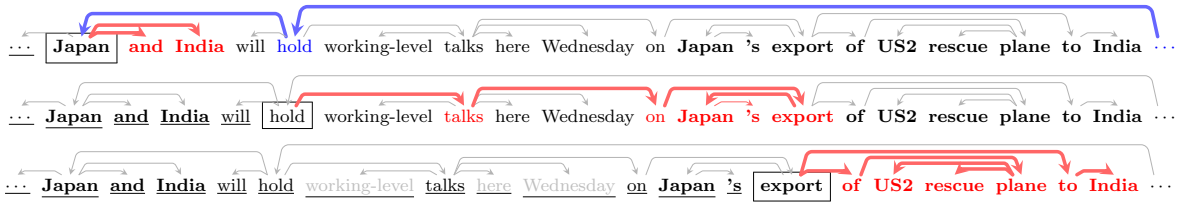

図 1: 図は圧縮中の例文とその依存関係関係を示す．灰色の単語は削除された単語を，黒枠内の単語は現在デコードされている単語をそれぞれ表している．下線が引かれた単語は既にデコードされていることを表す．親ノードを辿る経路は青い矢印として表され，子ノードを辿る経路は赤い矢印として表されている．太字の単語はこの文において重要な単語を表している．

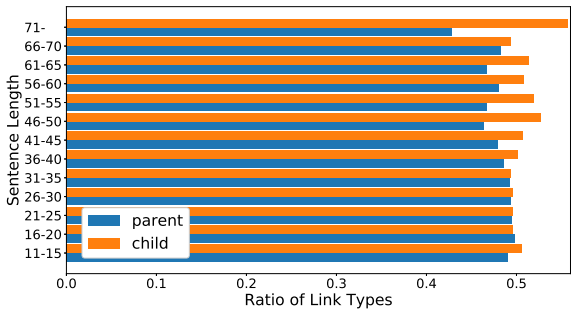

図 2: Seq2Seq において，文頭から文末へ，左から右へのデコードを行う際に，圧縮文中に含まれる単語がデコード中の圧縮文に含まれる単語の親として存在しているか，または子として存在しているかの割合．

縮文中に含めることができる．

そして，図 1 の中段の例において，デコーダは文中の単語 “hold” をデコードする際に，再帰的に “hold” の子の依存関係を辿ることで，文法性を維持しつつ重要なフレーズ “Japan’s export” を圧縮文に含めることが可能になる．

さらに，図 1 の下段の例のように、デコーダが文中の単語 “export” をデコードする際には，“export” の子の依存関係を辿ることで，重要なフレーズ “US2 rescue plane” と “of” を見つけ出し、これらの重要な単語を含めつつ文法的に正しい圧縮文を生成することができる．

なお，上記の例において，デコーダがある単語の依存構造木上の子を辿ることができない場合に，デコーダが明示的に重要な単語やフレーズを捉えることは難しい．さらに，図 2 が示す[*2]ように，Seq2Seq が将来デコードされる重要語の存在を捉えるために，現在デコードしている単語の依存構造上の親のみを辿ることは，特に長い文において不十分であると考えられる．

このように，依存構造木上の親と子を再帰的に辿ることにより，文法性を損なうことなく重要な単語を維持する，という考えを Seq2Seq に反映するために，我々はデコード時に階層的な注意機構に基づき統語的な先読みを行うことが可能な Seq2Seq，*syntactically look-ahead attention network*（SLAHAN）を提案する．SLAHAN は，依存構造上の親と子の両方の単語を明示的に辿ることで，将来の時刻においてデコードされる重要語を考慮することが可能となり，情報性の高い要約を生成することができる．SLAHAN において，単語間の依存関係は注意として表され，依存構造解析の誤りによる影響を軽減するために，依存構造と圧縮文は同時に学習される．さらに，文脈に応じて適切に親と子の情報を使用するために，再帰的に追跡された親と子からの情報の重要度は，ゲート機構により自動的に決定される．

Google sentence compression データセットを使用した評価実験において，SLAHAN は F1，ROUGE-1，ROUGE-2，および ROUGE-L スコアにおいてそれぞれ 85.5，79.3，71.3，および 79.1 のスコアを達成し，既存手法に対する改善を確認した．特に長文における精度の改善が顕著であった．また，人手評価によって，提案手法が可読性を損なうことなく重要な情報を保持した文の圧縮が可能であることについても確認した．

2. ベースとなる Seq2Seq

文圧縮はテキスト生成の一種であるが，一方で，系列中の単語に対して削除するか否かの決定を行う系列ラベリング問題として考えることも可能である．具体的には，入力文 $\mathbf{x} = (x_0, ..., x_n)$ が与えられた際に，入力文中のそれぞれの単語 x_t $(1 \leq t \leq n)$ に対して，(“維持”, “削除”, “文の終了”) を選択することにより，圧縮文を生成できる．なお，この定式化において，x_0 は文の開始記号を表している．

文法的に正しい圧縮文を生成するために，我々は Seq2Seq をベースとなるモデルとして選択した．頑健なベースモデルを作成するために，我々は ELMo[7] や BERT[8] などの文脈に基づく単語ベクトルを適用した．これにより，後ほど実験の節で説明するように，我々のベースモデルは Zhao ら [9] によって報告されている現在最高の F_1 値よりも高い値を達成している．

我々のベースモデルは単語埋め込み層，エンコーダ，デコーダ，出力層により構成される．単語埋め込み層では，

[*2] この統計は，Google sentence compression データセットの訓練データに含まれている正解の圧縮と，その依存構造解析結果に基づいて計算されている．

入力単語 x_i は下記のように埋め込みベクトル e_i へと変換される．

$$e_i = \|_{j=1}^{|\mathbf{F}|} F_{i,j}, \tag{1}$$

$\|$ はベクトルの結合を，$F_{i,j}$ は単語 x_i の j 番目の特徴量ベクトルを，$|\mathbf{F}|$ は単語埋め込み層で考慮される特徴量ベクトルの種類をそれぞれ表している．我々は単語の特徴量として，GloVe [10]，ELMo，BERT を選択して使用した．ELMo と BERT は複数の層により構成されるため，我々は各層の重み付き和を $F_{i,j}$ として次のように使用した．

$$\begin{aligned} F_{i,j} =& \textstyle\sum_{k=1}^{|\mathbf{L}|} \psi_{j,k} \cdot L_{i,j,k}, \\ \psi_{j,k} =& exp(\phi_{j,k} \cdot L_{i,j,k}) / \textstyle\sum_{l=1}^{|\mathbf{L}|} exp(\phi_{j,l} \cdot L_{i,j,l}), \end{aligned} \tag{2}$$

$L_{i,j,k}$ は j 番目の特徴量の k 番目の層を，$\phi_{j,k}$ は j 番目の特徴量の k 番目の重みをそれぞれ表している．BERT では入力トークンと出力ラベルの対応を取るために，単語中に含まれるサブワードのベクトルの平均値を単語のベクトルとして使用している．

エンコーダはまず順方向の LSTM を用いて e_i を隠れ状態 $\overrightarrow{h}_i = LSTM_{\overrightarrow{\theta}}(\overrightarrow{h}_{i-1}, e_i)$ へと変換する．$\overleftarrow{h}_i$ についても同様に逆方向の LSTM を用いて計算される．次に，$\overrightarrow{h}_i$ と $\overleftarrow{h}_i$ はベクトル $h_i = [\overrightarrow{h}_i, \overleftarrow{h}_i]$ として結合される．これらの過程により，エンコーダは $\mathbf{e}$ を隠れ状態 $\mathbf{h}$ へと次のように変換する．

$$\mathbf{h} = (h_0, ..., h_n). \tag{3}$$

なお，逆方向 LSTM の最終状態 $\overleftarrow{h}_0$ はデコーダの初期状態として引き継がれる．

デコーダは時刻 t において，前回の時刻において予測されたラベルに基づき決定される 3 ビットの one-hot ベクトル，前回の時刻の隠れ状態ベクトル d_{t-1} (後述)，単語埋め込み e_t を結合し，順方向 LSTM を用いてデコーダの隠れ状態 $\overrightarrow{s}_t$ へと変換する．

出力層は出力ラベルの確率を次のように計算する．

$$\begin{aligned} P(y_t \mid y_{<t}, \mathbf{x}) =& softmax(W_o d_t) \cdot \delta_{y_t}, \\ d_t =& tanh(W_d[h_t, \overrightarrow{s}_t] + b_d), \end{aligned} \tag{4}$$

W_d は重み行列を，b_d はバイアス項を，W_o はソフトマックス層の重み行列を，δ_{y_t} は y_t 番目の要素が 1 となり，それ以外の要素が 0 となるクロネッカーのデルタをそれぞれ表している．

3. 提案手法

この節では，まず提案手法 SLAHAN で使用している依存関係のグラフ表現について説明する．次に，SLAHAN のネットワーク構造と各モジュールの詳細についての説明を行う．なお，3.3 節で説明するように，SLAHAN において依存構造のグラフ表現とニューラルネットワークのパラメータは同時に学習される．

3.1 依存関係のグラフ表現

本節において，提案手法で使用している，単語の親と子を再帰的に辿るための依存関係のグラフ表現についての説明を行う．Hashimoto ら [11] において説明されているように，依存関係は重み付きグラフとして表現することができる．この表現では，文 $\mathbf{x} = (x_0, ..., x_n)$ が与えられた際に，それぞれの単語 x_j の親は $\mathbf{x}$ から選択される．なお，我々は x_0 をルートノードとして扱う．我々は文 $\mathbf{x}$ において，x_j が単語 x_t の親となる確率を $P_{head}(x_j|x_t, \mathbf{x})$ と表す．Kamigaito ら [6] は $P_{head}(x_j|x_t, \mathbf{x})$ を再帰的に用いることにより，x_j が x_t の d 次の依存関係上の親となる確率を次式のように計算している．

$$\alpha_{d,t,j} = \begin{cases} \sum_{k=1}^{n} \alpha_{d-1,t,k} \cdot \alpha_{1,k,j} & (d>1) \\ P_{head}(x_j|x_t, \mathbf{x}) & (d=1) \end{cases}. \tag{5}$$

上式 (5) の 1 行目は行列積の定義と同様であるため，$A^d_{j,t} = \alpha_{d,t,j}$ となる行列 $A^d_{j,t}$ を用いて，式 (5) は次式のように変形できる．

$$A^d = A^{d-1} A^1. \tag{6}$$

なお，本論文では今後 A^d を d 次の親グラフと呼ぶ．

我々は式 (6) を d 次の依存関係にある子を表すグラフ（以下，d 次の子グラフと呼ぶ）を表現できるように拡張する．まず，文 $\mathbf{x}$ において，x_t が x_j の依存構造上の子となる確率を $P_{child}(x_t|x_j, \mathbf{x})$，$x_j$ が親となる確率を $P_{\mathbf{x}}(x_j = p)$，xt が子となる確率を $P_{\mathbf{x}}(x_t = c)$，x_j が x_t と依存関係を持つ確率を $P_{\mathbf{x}}(x_j, x_t)$ とそれぞれ定義する．互いの確率が依存関係を持つ確率が他の関係に対して独立であると仮定した場合，次式の関係が導かれる．

$$\begin{aligned} P_{\mathbf{x}}(x_j, x_t) &= P_{child}(x_t|x_j, \mathbf{x}) \cdot P_{\mathbf{x}}(x_j = p), \\ P_{\mathbf{x}}(x_j, x_t) &= P_{head}(x_j|x_t, \mathbf{x}) \cdot P_{\mathbf{x}}(x_t = c). \end{aligned} \tag{7}$$

上式は次式のように変換できる．

$$\begin{aligned} & P_{child}(x_t|x_j, \mathbf{x}) \\ =& P_{head}(x_j|x_t, \mathbf{x}) \cdot P_{\mathbf{x}}(x_t = c) / P_{\mathbf{x}}(x_j = p). \end{aligned}$$

ここで，$P_{\mathbf{x}}(x_t = c)$ は依存構造木の定義より常に 1 となる．また，我々の定式化において，x_j は常に親として与えられているため，$P_{\mathbf{x}}(x_j = p)$ については定値として考えることができる．これらより，次式の関係を得ることができる．

$$P_{child}(x_t|x_j, \mathbf{x}) \propto P_{head}(x_j|x_t, \mathbf{x}). \tag{8}$$

式 (8) に基づき，x_j が x_t の d 次の子となる強さ $\beta_{d,t,j}$ を次式で定義する．

$$\beta_{d,t,j} = \begin{cases} \sum_{k=1}^{n} \beta_{d-1,t,k} \cdot \beta_{1,k,j} & (d>1) \\ P_{head}(x_t|x_j, \mathbf{x}) & (d=1) \end{cases}. \tag{9}$$

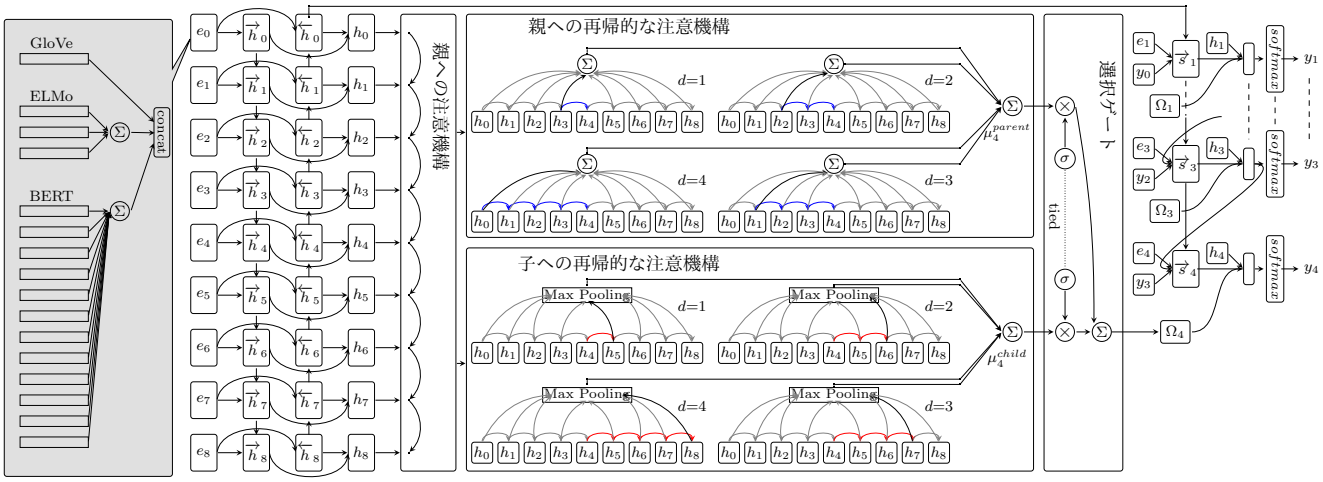

図 3: 提案手法のネットワーク構成図.

式 (5) と同様に，$B^d_{j,t} = \beta_{d,t,j}$ となる行列 $B^d_{j,t}$ を用いて，式 (9) は次式のように変形できる．

$$B^d = B^{d-1}B^1. \tag{10}$$

以降，我々は B^d を d 次の子グラフと呼ぶ．なお，式 (5) の定義と式 (9) より，A^1 と B^1 は常に $B^1_{tj} = A^1_{jt}$ の関係を満たす．この関係は $B^1 = (A^1)^T$ と変形することができる．さらに，行列の転置の定義より，我々は以下の式を得る．

$$\begin{aligned} B^d &= B^1B^1\cdots B^1 \\ &= (A^1)^T(A^1)^T\cdots(A^1)^T = (A^d)^T. \end{aligned} \tag{11}$$

すなわち，上式 (6) の計算結果を利用することにより，我々は式 (10) を直接計算することなく d 次の子グラフ B^d を利用することができる．これらより，提案手法 SLAHAN の計算量は隠れ状態ベクトルの次元数 d を用いて，従来手法 HiSAN [6] と同様に $O(n^2d^2)$ となる．これは次元数 d が多くの場合で入力文長 n よりも大きいという仮定に基づく．なお我々のベースとなる Seq2Seq の計算量は $O(nd^2)$ である．

3.2 ネットワーク構造

図 3 は SLAHAN の全体のネットワーク構成を示している．SLAHAN は前節で説明したベースとなる Seq2Seq の上に構築されている．それぞれの構成要素は次のような機能を持つ．

- **親への注意機構**はそれぞれの x_t に対し，x_j が x_t の親になる確率を式 (3) の隠れ状態 x_j と x_t を計算し，依存関係グラフをして出力する．
- **親への再帰的な注意機構**は親への注意機構で依存関係グラフに基づいて d 次の親グラフ A^d を計算する．そして $\alpha_{d,t,j}$ $(= A^d_{j,t})$ に基づいて，式 (3) で説明したエンコーダの隠れ状態ベクトルの集合 $\mathbf{h}$ から時刻 t におけるそれぞれの重み付き和 μ^{parent}_t を抽出する．
- **子への再帰的な注意機構**は d 次の子グラフ $\beta_{d,t,j}$ $(= B^d_{j,t})$ を用いて $\mathbf{h}$ から時刻 t におけるそれぞれの重み付き和 μ^{child}_t を抽出する．
- **選択ゲート**はデコーダが将来の時刻においてデコードをすることになる重要語を捉えるために，現在の文脈に基づいて重み付けをし，μ^{parent}_t と μ^{child}_t を足し合わせる．足し合わされた結果である Ω_t はデコーダにおいて出力ラベル y_t を決定するために使用される．

それぞれの構成要素の詳細な説明は次節以降において行う．

3.2.1 親への注意機構

Zhang ら [12] の研究と同様に，我々は $P_{head}(x_j|x_t,\mathbf{x})$ を次式に基づき計算する．

$$\begin{aligned} P_{head}(x_j|x_t,\mathbf{x}) &= softmax(g(h_{j'},h_t))\cdot\delta_{x_j}, \\ g(h_{j'},h_t) &= v_a^T\cdot tanh(U_a\cdot h_{j'} + W_a\cdot h_t), \end{aligned} \tag{12}$$

v_a，U_a，W_a はそれぞれ g の重み行列である．依存構造木において，ルートノードは親を持たず，各ノードは自分自身を親にはしない．これらの定義を満たすため，我々は $P_{head}(x_j|x_t,\mathbf{x})$ に対して以下の制約を加えた．

$$P_{head}(x_j|x_t,\mathbf{x}) = \begin{cases} 1 & (t=0\wedge j=0) \\ 0 & (t=0\wedge j>0) \\ 0 & (t\neq 0\wedge t=j). \end{cases} \tag{13}$$

上式 (13) の 1 行目と 2 行目はルートノードの親がルートノードとなる場合を制限している．また，3 行目は各ノードの親が自分自身を親とする場合を制限している．これらの制限のもとで，後に目的関数の節で説明するように，$P_{head}(x_j|x_t,\mathbf{x})$ はラベル出力確率 $P(\mathbf{y}\mid\mathbf{x})$ と同時に学習される．

3.2.2 親への再帰的な注意機構

親への再帰的な注意機構では式 (5) に基づき，$P_{head}(x_j|x_t,\mathbf{x})$ を再帰的に使用することにより $\alpha_{d,t,j}$ を

計算する．計算された $\alpha_{d,t,j}$ はエンコーダの隠れ状態 $\mathbf{h}$ に重み付けを行うために次のように使用される．

$$\gamma_{d,t} = \sum_{k=j}^{n} \alpha_{d,t,j} \cdot h_j. \tag{14}$$

入力文に対する適切な再起の回数 d を計算するために，$\gamma_{d,t}$ はさらに次元数に応じて，重みベクトル $\beta_{d,t}$ による重み付けが行われた上で次式のように μ_t^{parent} として足し合わされる．

$$\begin{aligned} c_t &= [\overleftarrow{h}_0, \overrightarrow{h}_n, h_t, \overrightarrow{s}_t], \\ \eta_{d,t} &= softmax(\gamma_{d,t} W_d^{parent} c_t) \cdot \delta_d, \\ \mu_t^{parent} &= \sum_{d \in \mathbf{d}} \eta_{d,t} \cdot \gamma_{d,t}, \end{aligned} \tag{15}$$

なお，W_d^{parent} は重み行列を，$\mathbf{d}$ は依存構造の再起回数の集合を，c_t は現在の文脈情報を含むベクトルを，それぞれ表している．

3.2.3 子への再帰的な注意機構

子への再帰的な注意機構では，d 次の子グラフ B^d に基づき，エンコーダの隠れ状態の集合 $\mathbf{h}$ への重み付けを行う．親への再帰的な注意機構とは異なり，B^d は確率ではないために，要素の総和は 1 にはならず，また依存構造中で 1 つの単語は 2 つ以上の子を持つこともある．これらの特徴は注意のみでは表現できないため，我々は式 (14) のような注意機構だけではなく，最大プーリング (max-pooling) についても使用した．最大プーリングは次式のように，$\beta_{d,t,j}$ を用いてエンコーダの隠れ状態の集合 $\mathbf{h}$ への重み付けを行った後に適用される．

$$\rho_{d,t} = MaxPool(\|_{k=j}^{n} (\beta_{d,t,j} \cdot h_j)^T). \tag{16}$$

入力文に対して適切な再起回数 d を選択するために，$\rho_{d,t}$ は現在の文脈に基づき計算された $\eta_{d,t}$ により次式のように μ_t^{child} として重み付けをされる．

$$\begin{aligned} \eta_{d,t} &= softmax(\rho_{d,t} W_d^{child} c_t) \cdot \delta_d, \\ \mu_t^{child} &= \sum_{d \in \mathbf{d}} \eta_{d,t} \cdot \rho_{d,t}, \end{aligned} \tag{17}$$

W_d^{child} は重み行列を表している．

3.2.4 選択ゲート

選択ゲートは依存構造上の親の情報を含む μ_t^{parent} と，子の情報を含む μ_t^{child} の重み付き和 Ω_t を，文脈に応じて決定されるゲート出力 z_t を用いて次式のように計算する．

$$\begin{aligned} \Omega_t &= z_t \circ \mu_t^{parent} + (1 - z_t) \circ \mu_t^{child}, \\ z_t &= \sigma(W_z [\mu_t^{parent}, \mu_t^{child}, c_t]), \end{aligned} \tag{18}$$

なお，$\circ$ は要素積を，σ はシグモイド関数を，W_z は重み行列をそれぞれ表す．そして，提案手法である SLAHAN では，式 (4) 中の d_t は $d'_t = [h_t, \Omega_t, \overrightarrow{s}_t]$ に置き換えられ，d'_t は時刻 $t+1$ のデコーダの入力としても使用される．

3.3 目的関数

構文解析結果の誤りの影響を低減させるために，提案手法である SLAHAN では依存構造木の親となる確率 $P_{head}(x_j|x_t)$ と出力ラベルの確率 $P(\mathbf{y}|\mathbf{x})$ は同時に学習される．親となる単語 w_j と子となる単語 w_t が依存関係をもつ場合を $a_{t,j} = 1$ と表記し，そうでない場合を $a_{t,j} = 0$ と表記する．この表記を用いて，SLAHAN の目的関数は次式のように定義される．

$$-logP(\mathbf{y}|\mathbf{x}) - \lambda \cdot \sum_{j=1}^{n} \sum_{t=1}^{n} a_{t,j} \cdot log\alpha_{1,t,j}, \tag{19}$$

λ は出力ラベルと依存関係の重要度を調整するためのハイパーパラメータである．依存関係の重要度を調査するために，我々は依存構造を考慮する設定 *with syntax (w/ syn)* では $\lambda = 1.0$ とし，依存構造を考慮しない設定 *with out syntax (w/o syn)* では $\lambda = 0.0$ とした．

4. 実験

ベースラインと提案手法を比較するために，我々は自動評価と人手評価を実施した．以下の節では実験の詳細についての説明を行う．

4.1 設定

4.1.1 データセット

評価には，Google sentence compression データセット (Google データセット) [4] を使用した．また，ドメイン外のデータセットにおける文圧縮性能を評価するために，Broadcast News Compression Corpus（BNC コーパス）[*3]についても使用した．これらのデータセットにおける実験設定は以下の通りである．**Google データセット:** 従来研究 [5], [6], [9], [13], [14] と同様に *comp-data.eval.json* の最初の 1000 文をテストデータとして使用した．また，*comp-data.eval.json* の最後の 1000 文を開発データとして使用した．近年行われた研究 [6], [9] を参考に，我々は *sent-comp.train*.json* に含まれる全ての 200,000 文を訓練データとして使用した．各文の依存構造木としてはデータセットに含まれているものを使用した．

さらに，長文における文圧縮の性能を調査するために，我々はテストデータの中で平均長 (= 27.04) よりも長い 417 文を対象とした評価についても実施した．

BNC コーパス: このデータセットは 3 人のアノテータによって作成された話し言葉に対する圧縮文への正解が含まれている．ドメイン外の長文に対する性能評価を実施するために，平均文長 19.83 よりも長い，595 文をテストデータとして扱った．訓練データについては Google データセットにおける設定と同様のものを用いた．また，このデータセットには依存構造木が含まれていないため，我々はこのデータセット中の全ての文に対して Stanford dependency

[*3] https://www.jamesclarke.net/research/resources

Glove	✓	✓		✓	✓		
ELMo	✓	✓	✓			✓	
BERT	✓		✓	✓			✓
$\mathbf{F_1}$	**86.2**	86.0	85.9	85.4	85.5	85.9	84.8

表 1: 開発データにおける **Base** の F_1 スコア．太字は表中で最も高い値であることを示す．

parser[*4]を用いた構文解析を行った．なお，BNC Corpus を用いた全ての評価において，我々は 3 人のアノテータとの比較における平均結果を報告している．

4.1.2 比較手法

我々のベースラインとなる比較手法は次の通りである．なお，全てのベースラインは入力に ELMo，BERT，GloVe ベクトルを使用している．

- **Tagger:** 両方向 LSTM を用いて入力文にラベル付けを行うモデルであり，多くの従来研究 [14], [15] で使用されている．
- **LSTM:** LSTM をエンコーダとデコーダに用いる Seq2Seq モデルであり，Fillipova ら [5] によって提案されたモデルと同じモデルである．
- **LSTM-Dep:** 上記の **LSTM** に依存構造木に関する特徴量を入力するモデルであり．Fillipova ら [5] の論文中で LSTM-Par-Pres と呼ばれているモデルである．
- **Base:** 第 2 節で説明したモデルである．
- **Attn:** Luong ら [16] が提案している Seq2Seq モデルにおいて，Fillipova ら [5] の研究を参考にエンコーダ側の単語埋め込み層の出力をデコーダ側の入力に用いたモデルである．
- **Parent:** 提案手法である SLAHAN から子への依存グラフを取り除いたモデルである．このモデルは従来研究のモデル HiSAN[6] と同様に，依存構造中の親のみしか辿ることができない．なお，公平な比較を行うために，式 (18) のゲート層に関しては取り除かずに残している．

比較対象となる我々の提案手法は次の通りである．

- **SLAHAN:** 第 3 節で説明した我々の提案手法
- **Child:** SLAHAN から親への再帰的な注意機構を取り除いたモデル．依存構造中の子のみを辿ることができる．**Parent** と同様に，公平な比較のためにゲート層は残している．

4.1.3 Model Parameters

GloVe を使用する際には *glove.840B.300d* を使用し，BERT を使用する際には *cased_L-12_H-768_A-12* を使用した．ELMo は全 3 層，BERT は全 12 層を使用した．最も良い特徴量の組み合わせを使用するために，我々は開発データを用いて調査を行った．それぞれの設定における F1 スコアの結果を表 1 に示す．この結果に従い，我々は実験において GloVe，ELMo，BERT を組み合わせて使用した．

LSTM と注意機構の隠れ層の次元数は 200 を，LSTM の層数には 2 を用いた．これらの設定は ELMo を用いた際の LSTM に基づく固有表現タグ付けの際の設定 [7] に従っている．全てのパラメータは Glorot らの手法 [18] に基づいて初期化されている．また，全てのモデルの LSTM の入力に，比率 0.3 でドロップアウト層を適用した．学習器としては Adam [19] を初期学習率 0.001 で使用した．全ての勾配はミニバッチ内で平均化されている．また勾配クリッピングの閾値には 5.0 を使用した．そして，最大の学習エポック数は 20 に設定した．なお，提案手法において，式 (15) と式 (17) の $\mathbf{d}$ には $\{1, 2, 3, 4\}$ を使用した．ミニバッチサイズの最大数は 16 を使用し，ミニバッチの順序は学習時にランダムにシャッフルされている．最終的に用いるモデルとしては，それぞれのエポックにおいて開発データで評価した際に，最も正解と完全一致する文の数が多いものを選択した．

圧縮文を出力する際には，先行研究 [6] に従い，貪欲法を用いた．実装には Dynet [20] を使用した．

4.2 自動評価

4.2.1 評価尺度

自動評価の尺度として，我々は従来研究と比較するために，圧縮文中に維持された維持されたトークンに対する F_1 値 ($\mathbf{F_1}$)[5] を用いた．この評価尺度において，適合率は出力された圧縮文に含まれるトークンのうち，正解となる圧縮文に含まれるものの割合であり，再現率は正解となる圧縮文に含まれるトークンのうち，出力された圧縮文に含まれるものの割合である．

より頑健な評価のために，我々はさらに ROUGE-1 (**R-1**)，ROUGE-2 (**R-2**)，ROUGE-L (**R-L**) [21] を自動評価の尺度として用いた．[*5]我々はさらに，圧縮文が適切な長さを出力しているかについても確認するために，$\Delta C =$ システムが出力圧縮文の圧縮率 $-$ 正解となる圧縮文の圧縮率 [6] を評価尺度として用いた．なお，正解となる圧縮文の圧縮率については，対象が Google データセットの全文の場合が 43.7，長文のみの場合に 32.4 となっており，BNC コーパスにおいては 70.8 である．全てのスコアには文単位のマクロ平均の結果を用いており，学習時の揺らぎの影響を弱めるために．3 回の平均を行なったものを報告値としている．

4.2.2 結果

表 2 に Google データセットにおける結果を示す．表より，**SLAHAN** は全ての文を対象とした場合と，長文を対象とした場合の両方の設定において，最高のスコアを達成

[*4] `https://nlp.stanford.edu/software/`

[*5] 我々は ROUGE の計算に ROUGE-1.5.5 スクリプトに “-n 2 -m -d -a” のオプションを付与して使用した．

		全文					長文				
		F_1	**R-1**	**R-2**	**R-L**	**ΔC**	**F_1**	**R-1**	**R-2**	**R-L**	**ΔC**
Evaluator-LM [9]		85.0	-	-	-	-2.7	-	-	-	-	-
Evaluator-SLM [9]		85.1	-	-	-	-4.7	-	-	-	-	-
Tagger		85.0	78.1	69.9	77.9	-3.1	83.0	75.4	66.8	74.9	-3.1
LSTM		84.8	77.7	69.6	77.4	-3.4	82.7	74.8	66.3	74.4	-3.5
LSTM-Dep		84.7	77.8	69.7	77.5	-3.3	82.6	74.9	66.5	74.4	-3.3
Attn		84.5	77.3	69.3	77.1	-3.8	82.3	74.7	66.4	74.3	-3.6
Base		85.4	78.5	70.4	78.2	-2.9	83.4	75.8	67.4	75.3	-3.0
Parent	w/ syn	85.0	78.3	70.3	78.1	-2.5	82.8	75.3	67.0	74.9	-2.9
Parent	w/o syn	85.3	78.3	70.4	78.1	-3.4	83.3	75.6	67.3	75.2	-3.4
Child	w/ syn	85.4	78.8	70.7	78.5	-2.9	83.0	75.8	67.3	75.4	-3.0
Child	w/o syn	85.2	78.6	70.8	78.4	-3.1	83.2	76.3	68.2	75.8	-2.8
SLAHAN	w/ syn	**85.5**	**79.3**†	**71.4**†	**79.1**†	**−1.5**†	83.3	**76.6**	68.3	**76.1**	**−1.9**†
SLAHAN	w/o syn	85.4	78.9†	71.0†	78.6†	-3.0	**83.6**	76.5†	**68.5**†	**76.1**†	-2.9

表 2: Google データセットにおける自動評価の結果. 太字は最も高いスコアを表し，† は提案手法の結果がベースライン中で最もスコアが高いものと比較して，統計的に優位な差があることを示している．検定については，ペアードブートストラップリサンプリング法 [17] を使用し，1,000,000 回のサンプルを取得することで実施した ($p < 0.05$).

		F_1	**R-1**	**R-2**	**R-L**	**ΔC**
Tagger		54.6	36.8	27.7	36.4	-39.1
LSTM		54.8	36.6	28.0	36.2	-39.2
LSTM-Dep		55.1	36.9	28.2	36.5	-38.8
Attn		54.1	36.1	27.4	35.6	-39.6
Base		55.4	37.4	28.5	36.9	-38.6
Parent	w/ syn	54.2	36.3	27.7	35.9	-39.1
Parent	w/o syn	54.0	35.8	27.2	35.4	-40.1
Child	w/ syn	55.6	37.8	28.5	37.3	-38.2
Child	w/o syn	54.8	36.7	28.1	36.3	-39.2
SLAHAN	w/ syn	**57.7**†	**40.1**†	**30.6**†	**39.6**†	**−35.9**†
SLAHAN	w/o syn	54.6	36.4	27.8	36.0	-39.5

表 3: BNC コーパスにおける結果．表中の表記は表 2 と同様である．

	可読性		情報性	
Tagger	3.90	(73.4)	3.79	(72.9)
Base	3.86	(72.4)	3.80	(73.6)
Parent w/ syn	3.82	(70.5)	3.77	(71.5)
Child w/ syn	**3.94**	(**75.8**)	3.85†	(74.9)
SLAHAN w/ syn	3.91	(74.8)	**3.90**†	(**77.9**†)

表 4: 人手評価の結果．( ) 内の数字は 4 以上のスコアの割合を表す．† については表 2 と同様である．

している．これらの改善より，**SLAHAN** は依存関係にある親と子を辿ることにより，重要語の情報を捉えることに成功していると考えられる．**Child** は **Parent** よりも高いスコアを達成している．この結果は我々が観測した図 2 における，特に長文においては依存関係にある子の情報を考慮することが重要であるという結果と一貫している．そして，**SLAHAN w/o syn** は **SLAHAN w/ syn** よりも高いスコアを達成している．この結果はドメイン内のデータに対しては明示的な依存構造の情報が与えられなくとも，重要語の情報を捉えるための依存構造グラフが学習できることを示している．

BNC コーパスにおける自動評価の結果を表 3 に示す．この結果からは **SLAHAN w/ syn** が他のモデルと比較し，極めて高いスコアを達成していることが読み取れる．**SLAHAN w/ syn** を **Base**, **Parent**, **Child**, **SLAHAN** と比較すると，textbfSLAHAN w/ syn は BNC コーパスにおいても Google データセットと同様に他のモデルよりも重要語を正しく捉えられていることが分かる．**SLAHAN w/syn** の高い結果は明示的な文法情報を用いることの重要性を示している．これらより，ドメイン外のデータにおいては文法情報を明示的に用いずに，学習データのみから構築された依存構造グラフは明示的な文法情報を用いたモデルよりもスコアが低くなることが分かる．この結果は従来研究 [14] において観測されている結果とも一貫する．これらの結果より，SLAHAN は長文とドメイン外のデータの両方で効果的であることが分かる．

4.3 人手評価

人手評価では，我々は自動評価において高い **R-L** スコ

入力: British mobile phone giant Vodafone said Tuesday it was seeking regulatory approval to take full control of its Indian unit for $ 1.65 billion , after New Delhi relaxed foreign ownership rules in the sector .
正解: Vodafone said it was seeking regulatory approval to take full control of its Indian unit .
Base: Vodafone said it was seeking regulatory approval to take control of its unit .
Parent w/ syn: Vodafone said it was seeking approval to take full control of its Indian unit .
Child w/ syn: Vodafone said it was seeking regulatory approval to take control of its Indian unit .
SLAHAN w/ syn: Vodafone said it was seeking regulatory approval to take full control of its Indian unit .

入力: Broadway 's original Dreamgirl Jennifer Holliday is coming to the Atlanta Botanical Garden for a concert benefiting Actor 's Express .
正解: Broadway 's Jennifer Holliday is coming to the Atlanta Botanical Garden .
Base: Jennifer Holliday is coming to the Atlanta Botanical Garden .
Parent w/ syn: Broadway 's Jennifer Holliday is coming to the Atlanta Botanical Garden .
Child w/ syn: Jennifer Holliday is coming to the Atlanta Botanical Garden .
SLAHAN w/ syn: Broadway 's Jennifer Holliday is coming to the Atlanta Botanical Garden .

入力: Tokyo , April 7 Japan and India will hold working-level talks here Wednesday on Japan 's export of US2 rescue plane to India , Japan 's defence ministry said Monday .
正解: Japan and India will hold talks on Japan 's export of US2 rescue plane to India .
Base: Japan and India will hold talks Wednesday on export of plane to India .
Parent w/ syn: Japan and India will hold talks on Japan 's export plane .
Child w/ syn: Japan and India will hold talks on Japan 's export of US2 rescue plane to India .
SLAHAN w/ syn: Japan and India will hold talks on Japan 's export of plane to India .

表 5: 圧縮文の例.

アを達成した上位 5 件のモデル[*6]を対象とした．我々は Google データセットにおけるテストセットの結果から，全てのモデルの出力が同じものを除外し，その中から先頭 100 件入力文と圧縮文を抽出し，評価に用いた．これらの圧縮文は可読性と情報性の二つの尺度によって評価された．評価は 12 人の評価者によって 1（最低）から 5（最高）の 5 段階のリッカート尺度を用いて行い，最終的に最もスコアが高い評価者と最もスコアが低い評価者の結果を除外し，10 人の評価者の評価結果の平均値を評価結果として用いた．

表 4 に人手評価の結果を示す．この結果から，提案手法 **SLAHAN w/ syn** と **Child w/ syn** は可読性を損なうことなく情報性を向上させることに成功していることが分かる．これらの結果は自動評価における結果と合致している．

[*6] 評価に用いるモデルとしては，平均を取るために学習された 3 つのモデルのうち，最も F_1 スコアが高いものを選んだ．

5. 分析

表 5 はそれぞれのモデルにおける実際の出力を表している．最初の例において，我々は **SLAHAN** が文を正しく圧縮できていることが分かる．しかし，**Parent** と **Child** はそれぞれ “regulatory” と “full” を欠落させている．これは **Parent** と **Child** は依存構造中の親と子の両方を辿ることができないためであると考えられる．この結果は **SLAHAN** の選択ゲートが長文においても正しく動いていることを示している．

2 つ目の例では，**Child** が誤ってフレーズ “Broadway 's” を欠落させている一方で，**SLAHAN** と **Parent** は文を正しく圧縮できている．これは **Child** が依存関係の子のみに着目するために，明示的に “Broadway 's” から “Jennifer Holliday” を辿ることができないためであると考えられる．この結果から，**SLAHAN** のゲート機構が正しく親を辿るべきか子を辿るべきかを判断できていることが分かる．

3 つ目の結果では **Child** のみが正しく文を圧縮できている．この理由としては，図 1 に示したように，この文における重要語はデコーダの各時刻に位置する単語の子を辿ったものが多いためであると考えられる．対照的に，**SLAHAN** の圧縮文では “US2 rescue” が欠落してしまっている．これは選択ゲートが親を辿るべきか，子を辿るべきかを正しく判断できていないためであると考えられ，より高精度な圧縮文を生成するためには選択ゲートに対するさらなる改善が必要であると考えられる．

6. 関連研究

文圧縮タスクにおいては，Fillipova ら [5] により，従来の構文木の刈り込み [1], [2], [3], [4] に代わる手法として LSTM に基づく手法が提案され，構文解析誤りの影響を避けつつ圧縮文を生成することが可能となった．Fillipova ら [5] はさらに，LSTM において構文解析結果を用いるために，LSTM に基づくデコーダにおいて単語の依存関係にある親を考慮することが可能な手法を提案している．Wang ら [14] は LSTM に基づく出力スコアと整数計画問題に基づく木の刈り込みを組み合わせることで，LSTM がドメイン内のデータに過学習することを回避可能な手法を提案している．これらの手法は文法情報を明示的に用いることが可能な一方で，構文解析誤りの影響を受けやすいという問題が存在する．

Kamigaito ら [6] は彼らが提案した再帰的注意機構を用いて再帰的に依存関係にある親を辿ることにより，高次の依存関係を考慮することが可能な文圧縮手法を提案してい

る．従来手法とは異なり，彼らの手法は文圧縮と依存関係にある親となる確率を同時学習することにより，構文解析誤りの影響を避けることができる．同様に Zhao ら [9] も文法情報に基づく言語モデルを用いることで，明示的な構文解析木を用いることなく文を圧縮することが可能な手法を提案している．

我々の SLAHAN は ELMo や BERT などの強力な事前学習された言語モデルの特徴量を取り入れながら，構文解析誤りの影響を受けることなく，依存関係にある親と子の両方を辿って重要語を考慮可能である．さらに，SLAHAN は選択ゲートの働きにより，将来デコードされる単語を考慮することで，重要語を保持した圧縮文を生成することが可能である．

7. 結論

この論文では，我々は新たな Seq2Seq モデルとして，依存構造上の親と子を辿ることにより情報性が高い圧縮文を生成することが可能なモデルである *syntactically look-ahead attention network* (SLAHAN) を提案した．実験の結果，SLAHAN は F1，ROUGE-1，ROUGE-2，ROUGE-L の全ての評価尺度において，長文とドメイン外の両方において最高の精度を達成した．また，人手評価において，SLAHAN はは可読性を低下させることなく情報性を向上させることが可能であることを示した．これらの結果より，Seq2Seq モデルでは，依存関係を用いて将来の時刻においてデコードされる単語を事前に捉えることが長文の圧縮に有用であると結論づけることができる．

謝辞

NTT コミュニケーション科学基礎研究所の平尾努氏には様々な助言をいただきましたことを深く感謝しております．

参考文献

[1] Jing, H.: Sentence Reduction for Automatic Text Summarization, *Proceedings of the Sixth Conference on Applied Natural Language Processing*, Seattle, Washington, USA, Association for Computational Linguistics, pp. 310–315 (online), DOI: 10.3115/974147.974190 (2000).

[2] Knight, K. and Marcu, D.: Statistics-based summarization-step one: Sentence compression, *AAAI/IAAI*, Vol. 2000, pp. 703–710 (2000).

[3] Berg-Kirkpatrick, T., Gillick, D. and Klein, D.: Jointly Learning to Extract and Compress, *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, Portland, Oregon, USA, Association for Computational Linguistics, pp. 481–490 (online), available from ⟨http://www.aclweb.org/anthology/P11-1049⟩ (2011).

[4] Filippova, K. and Altun, Y.: Overcoming the Lack of Parallel Data in Sentence Compression, *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, Seattle, Washington, USA, Association for Computational Linguistics, pp. 1481–1491 (online), available from ⟨http://www.aclweb.org/anthology/D13-1155⟩ (2013).

[5] Filippova, K., Alfonseca, E., Colmenares, C. A., Kaiser, L. and Vinyals, O.: Sentence Compression by Deletion with LSTMs, *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, Lisbon, Portugal, Association for Computational Linguistics, pp. 360–368 (online), available from ⟨http://aclweb.org/anthology/D15-1042⟩ (2015).

[6] Kamigaito, H., Hayashi, K., Hirao, T. and Nagata, M.: Higher-Order Syntactic Attention Network for Long Sentence Compression, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, New Orleans, Louisiana, Association for Computational Linguistics, pp. 1716–1726 (online), DOI: 10.18653/v1/N18-1155 (2018).

[7] Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K. and Zettlemoyer, L.: Deep Contextualized Word Representations, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, New Orleans, Louisiana, Association for Computational Linguistics, pp. 2227–2237 (online), DOI: 10.18653/v1/N18-1202 (2018).

[8] Devlin, J., Chang, M.-W., Lee, K. and Toutanova, K.: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, *arXiv preprint arXiv:1810.04805* (2018).

[9] Zhao, Y., Luo, Z. and Aizawa, A.: A Language Model based Evaluator for Sentence Compression, *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, Melbourne, Australia, Association for Computational Linguistics, pp. 170–175 (online), available from ⟨https://www.aclweb.org/anthology/P18-2028⟩ (2018).

[10] Pennington, J., Socher, R. and Manning, C. D.: GloVe: Global Vectors for Word Representation, *Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543 (online), available from ⟨http://www.aclweb.org/anthology/D14-1162⟩ (2014).

[11] Hashimoto, K. and Tsuruoka, Y.: Neural Machine Translation with Source-Side Latent Graph Parsing, *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, Copenhagen, Denmark, Association for Computational Linguistics, pp. 125–135 (online), available from ⟨https://www.aclweb.org/anthology/D17-1012⟩ (2017).

[12] Zhang, X., Cheng, J. and Lapata, M.: Dependency Parsing as Head Selection, *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, Valencia, Spain, Association for Computational Linguistics, pp. 665–676 (online), available from ⟨http://www.aclweb.org/anthology/E17-1063⟩ (2017).

[13] Tran, N.-T., Luong, V.-T., Nguyen, N. L.-T. and Nghiem, M.-Q.: Effective Attention-based Neural Architectures for Sentence Compression with Bidirectional Long Short-term Memory, *Proceedings of the Seventh Symposium on Information and Communication Technology*, SoICT '16, New York, NY, USA, ACM, pp. 123–

130 (online), DOI: 10.1145/3011077.3011111 (2016).
[14] Wang, L., Jiang, J., Chieu, H. L., Ong, C. H., Song, D. and Liao, L.: Can Syntax Help? Improving an LSTM-based Sentence Compression Model for New Domains, *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Vancouver, Canada, Association for Computational Linguistics, pp. 1385–1393 (online), available from ⟨http://aclweb.org/anthology/P17-1127⟩ (2017).
[15] Klerke, S., Goldberg, Y. and Søgaard, A.: Improving sentence compression by learning to predict gaze, *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, San Diego, California, Association for Computational Linguistics, pp. 1528–1533 (online), available from ⟨http://www.aclweb.org/anthology/N16-1179⟩ (2016).
[16] Luong, T., Pham, H. and Manning, C. D.: Effective Approaches to Attention-based Neural Machine Translation, *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, Lisbon, Portugal, Association for Computational Linguistics, pp. 1412–1421 (online), available from ⟨http://aclweb.org/anthology/D15-1166⟩ (2015).
[17] Koehn, P.: Statistical Significance Tests for Machine Translation Evaluation, *Proceedings of EMNLP 2004*, Barcelona, Spain, Association for Computational Linguistics, pp. 388–395 (online), available from ⟨https://www.aclweb.org/anthology/W04-3250⟩ (2004).
[18] Glorot, X. and Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pp. 249–256 (2010).
[19] Kingma, D. P. and Ba, J.: Adam: A Method for Stochastic Optimization, *CoRR*, Vol. abs/1412.6980 (online), available from ⟨http://arxiv.org/abs/1412.6980⟩ (2014).
[20] Neubig, G., Dyer, C., Goldberg, Y., Matthews, A., Ammar, W., Anastasopoulos, A., Ballesteros, M., Chiang, D., Clothiaux, D., Cohn, T. et al.: DyNet: The Dynamic Neural Network Toolkit, *arXiv preprint arXiv:1701.03980* (2017).
[21] Lin, C.-Y. and Och, F. J.: Automatic Evaluation of Machine Translation Quality Using Longest Common Subsequence and Skip-Bigram Statistics, *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL'04), Main Volume*, Barcelona, Spain, pp. 605–612 (online), DOI: 10.3115/1218955.1219032 (2004).