# 翻訳と見出し文生成の同時学習による言語横断見出し文生成モデル

高瀬 翔　岡崎 直観
東京工業大学
sho.takase@nlp.c.titech.ac.jp　okazaki@c.titech.ac.jp

## 1 はじめに

言語横断要約は，ある文書についての要約を，元の文書とは別の言語で生成するタスクである [1, 2, 3, 4]．不慣れな言語で書かれた文書について，馴染み深い言語で書かれた要約を言語横断要約によって得ることができれば，その文書の要旨を手早く掴むことができる．本論文では，言語横断要約の一種として，言語横断見出し文生成に取り組む．

言語横断要約は，伝統的には元の文書を対象言語に翻訳してから要約する，あるいは，元の文書の要約を生成し，要約を対象言語に翻訳するといったパイプライン手法が主流である [1, 2]．一方で，近年，翻訳や要約タスクで高い性能を達成している，ニューラルネットワークを用いたエンコーダ・デコーダを適用し，元の文書から対象言語の要約を直接生成する手法も提案されている [3, 4]．

エンコーダ・デコーダを用いた場合には，言語横断要約の大規模な訓練データを準備することが肝要である．Duan らや Zhu らは単言語要約コーパスの元文書や要約に機械翻訳を適用することにより，言語横断要約の訓練データを構築した [3, 4]．これら既存研究にならい，本研究でも，単言語の文書と見出し文対コーパスから言語横断見出し文生成の訓練データを構築し，エンコーダ・デコーダの学習を行う．加えて，本研究では対訳コーパスからも言語横断見出し文生成の訓練データを構築し，訓練データを増強する．

言語横断要約の性能を向上させるために，Zhu らはデコーダを 2 つ用意し，言語横断要約と言語横断要約での対象言語への翻訳（あるいは単言語での要約生成）を同時に学習する手法を提案した [4]．この手法ではエンコーダはパラメータを共有しているが，デコーダのパラメータはそれぞれ独立である．一方で，デコーダもパラメータを共有することにより，各パラメータの学習に用いる事例数が増え，さらなる性能向上が期待できる．これをふまえ，本研究では，エンコーダ・デコーダをひとつのみ用意し，言語横断見出し文生成，言語横断見出し文生成での対象言語への翻訳，単言語での見出し文生成の同時学習手法を提案する．これにより，翻訳，単言語での見出し文生成，言語横断見出し文生成の訓練事例をすべて学習に用いることができ，エンコーダ・デコーダの質の向上が期待できる．提案手法では，多言語の翻訳をひとつのエンコーダ・デコーダで行う手法（GNMT）[5] を参考にし，入力文に，要請する出力のタグ（例えば見出し文生成では<S>）を付与し，学習を行う．

提案手法は極めて単純ながら，言語横断見出し文生成の性能を向上させ，既存研究が作成した中国語から英語への短文要約データセット [3] において，既存研究を上回る性能を達成した．また，提案手法は翻訳，単言語の見出し文生成の性能向上にも寄与することを示す．特に翻訳タスクにおいて，対訳コーパスのみを用いて学習した場合と比べて性能が大きく向上し，最高性能のスコアと同程度の値を達成した．

## 2 各タスクの定義

本節では，本論文で扱う各タスクの定義を説明する．

### 2.1 翻訳

ある言語で書かれた文を対象言語の文に変換する．具体的には元の言語の語彙数を $V_s$，対象言語の語彙数を $V_t$ としたときに，与えられた長さ M の系列 $\boldsymbol{x} = \{\boldsymbol{x_1}, ..., \boldsymbol{x_M}\}$，$\boldsymbol{x_i} \in \{0,1\}^{V_s}$ について，下記を満たす系列 $\boldsymbol{y^t} = \{\boldsymbol{y^t_1}, ..., \boldsymbol{y^t_N}\}$，$\boldsymbol{y^t_j} \in \{0,1\}^{V_t}$ を出力する．

$$\arg\max_{\boldsymbol{y^t}} P(\boldsymbol{y^t}|\boldsymbol{x}) \tag{1}$$

### 2.2 見出し文生成

与えられた文に対応する見出し文を生成する．加えて，実際に見出し文生成器を活用する際には，与えられたスペースに応じた見出しを生成する必要がある．言い換えれば，見出し文生成では指定された長さ $L$ の見出し文を生成する．具体的には与えられた所望の長さ $L$ と系列 $\boldsymbol{x} = \{\boldsymbol{x_1}, ..., \boldsymbol{x_M}\}$，$\boldsymbol{x_i} \in \{0,1\}^{V_s}$ について，下記を満たす系列 $\boldsymbol{y^s} = \{\boldsymbol{y^s_1}, ..., \boldsymbol{y^s_L}\}$，$\boldsymbol{y^s_j} \in \{0,1\}^{V_s}$ を出力する．

$$\arg\max_{\boldsymbol{y^s}} P(\boldsymbol{y^s}|\boldsymbol{x}, L) \tag{2}$$

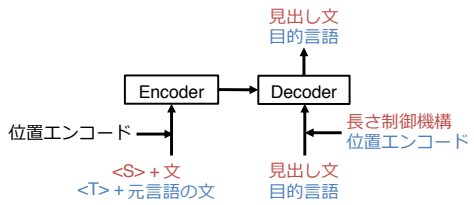


図 1: 提案手法の概要図．入力文に出力を指定するタグを付与し，翻訳と見出し文生成を同時に学習する．

### 2.3 言語横断見出し文生成

言語横断見出し文生成では，与えられた文に対し，所望の長さ $L$ で，かつ対象言語の見出し文を生成する．与えられた所望の長さ $L$ と系列 $\boldsymbol{x} = \{\boldsymbol{x_1}, ..., \boldsymbol{x_M}\}$, $\boldsymbol{x_i} \in \{0,1\}^{V_s}$ について，下記を満たす系列 $\boldsymbol{y^t} = \{\boldsymbol{y^t_1}, ..., \boldsymbol{y^t_L}\}$, $\boldsymbol{y^t_j} \in \{0,1\}^{V_t}$ を出力する．

$$\arg\max_{\boldsymbol{y^t}} P(\boldsymbol{y^t}|\boldsymbol{x}, L) \tag{3}$$

## 3 提案手法：出力タグを用いたエンコーダ・デコーダ

本論文での提案手法の概要を図 1 に示す．この図のように，提案手法では出力に応じたトークン（以降，出力タグと呼ぶ）を入力文の先頭に付与する．本手法は GNMT の，出力先の言語を示すタグを入力文の先頭に付与するという手法を元にしており [5]，出力タグは入力文には依存しない．言い換えれば，見出し文生成，言語横断見出し文生成のどちらも出力は見出し文とみなし，入力文にタグを付与する．これにより，翻訳，見出し文生成の訓練データのみしかない場合でも，言語横断見出し文生成を行うこと（Zero-shot な言語横断見出し文生成）が可能となる．訓練用の大規模なコーパスの存在しない言語横断見出し文生成タスクでは，Zero-shot での生成能力は重要であると考えられる．図 1 では，出力が見出し文の際には<S>を，翻訳の際には<T>をそれぞれ付与している．

2 節で述べたように，見出し文生成では所望の長さ $L$ を満たす見出し文を出力する必要がある．これを達成するために，出力タグに加え，出力長を制御する機構をエンコーダ・デコーダに加える．本論文では，Transformer [6] をベースのエンコーダ・デコーダとして採用するため，Transformer の位置エンコードを拡張した，出力長制御手法 [7] を採用する．この手法は，入力トークンの絶対位置を表す手法である位置エンコードについて，デコーダ側についてのみ，指定された出力長と入力トークンとの相対位置を示す値とする．提案手法では図 1 のように，デコーダ側について，出力タグが翻訳の際には通常の位置エンコードを，見出し文の際には出力長制御用の位置エンコードを用いることにより，見出し文生成，言語横断見出し文生成における出力長制御を可能とする．

## 4 言語横断見出し文生成の訓練データ

提案手法では出力タグを用意して翻訳，見出し文生成を学習することにより，Zero-shot な言語横断見出し文生成が可能となる．加えて，言語横断見出し文生成のデータを自動的に生成し，学習に用いることで，性能向上が期待できる．本研究では，Duan らの用いた，逆翻訳を用いる手法 [3] に加え，対訳コーパスに見出し文生成器を適用することで，言語横断見出し文生成の訓練データを構築する．

### 4.1 逆翻訳

Duan らは単言語の見出し文とその入力文との対について，入力文に機械翻訳を適用することで，言語横断見出し文生成の訓練データを構築した [3]．本研究でも彼らと同様の手法を適用し，言語横断見出し文生成の訓練データを得る．

さらに，本研究では，翻訳結果と翻訳元の文とを機械翻訳の訓練データとして活用する．なお，この逆翻訳によって得た疑似対訳コーパスについては，入力側に出力タグではなく，疑似対訳コーパスであることを示すタグを付与する [14]．

### 4.2 対訳コーパスに紐づく見出し文生成

本研究では，対訳コーパス中に含まれる，一方の言語の文に対し，単言語の見出し文生成器を適用することで見出し文を生成し，言語横断見出し文生成の訓練データを構築する．また，対訳コーパスを一切要約の行われていない言語横断見出し文のデータとみなし，言語横断見出し文生成の訓練に用いる．

## 5 実験

### 5.1 データセット

本論文では，公開されている言語横断見出し文生成のテストセットとして，Duan らの構築した，中-英の言語横断見出し文生成データ [3] を用いる．このため，中-英の対訳コーパス，および，英語の見出し文生成データセットを用意し，エンコーダ・デコーダの訓練や，言語横断見出し文生成の訓練データ構築に用いる．

中-英の対訳コーパスとしては，LDC から抽出した，約 90 万文対を利用する[1]．なお，既存の，中-英の機械翻訳に取り組んだ研究 [10, 13] で使用したとされている LDC2003E14 に関しては，現在 LDC から配布されていないとのことで，入手することができなかった．このため，これら既存研究で訓練に用いている対訳コーパスと比べ，本研究での対訳コーパスは 30 万文対以上少なく，既存研究と比べると，ベースラインの翻訳性能が低い可能性がある点に留意されたい．中-英の翻訳性能に関しては LDC で公開している，NIST

[1] 具体的には，LDC2004T07，LDC2004T08（hansards の部分のみ），LDC2005T06，LDC2015T06 である．

2002 - 2006 を用い，mteval-v13a.pl を用いて BLEU 値を算出する．

英語の見出し文生成データセットとしては，Annotated English Gigaword[2]に Rush ら [8] の前処理プログラム[3]を適用して得た約 380 万対を利用する．なお，本研究では Rush らの前処理プログラムを文と見出し文の抽出のみに利用しており，数字や低頻度語を特殊なトークンに置き換える処理は行っていない．英語の見出し文生成の性能に関しては，Duan らの中-英の言語横断見出し文生成データの元となった，DUC 2004 task 1[4]を用いる．

本実験では，SentencePiece[5]の Unigram 言語モデル [15] を用いて語彙を構築した．語彙サイズは各コーパスにおいて 3 万 2 千とした．

本実験のベースライン，および，4 節で言及した，言語横断見出し文生成の訓練データ構築には，Transformer [6, 16] を用いる[6]．なお，中-英翻訳，英語の見出し文生成のベースラインとしては，各タスクの訓練コーパスのみを用いた．また，英語の見出し文生成では，長さ制御機構 [7] を加えた．

## 5.2 結果

本論文で構築した言語横断見出し文生成の訓練データおよび，対訳コーパス，単言語の見出し文コーパスを訓練データとして用い，学習した提案手法の，言語横断見出し文生成での ROUGE 値[7]を表 1 に示す．加えて，この表には中-英の対訳コーパスのみで学習した Transformer により入力文を翻訳した際の結果，その翻訳結果に単言語の見出し文生成器を適用したパイプライン処理の結果，および先行研究である Duan らの報告値を記している．この表から，提案手法は他の手法よりも高い性能を達成していることが分かる．特に，提案手法は極めて単純な手法ながら，ほとんど同じ対訳コーパス，単言語の見出し文コーパスを利用している Duan ら[8]より高い性能を達成している点は特筆すべき点である．

また，表 1 には Zero-shot，すなわち，自動構築した言語横断見出し文生成の訓練データを用いずに学習を行ったモデルの結果も記している．この表から，Zero-shot の状況でも，パイプライン処理より ROUGE-1 については高いスコアを達成していることが分かる．

提案手法の単言語（英-英）の見出し文生成の結果，および，中-英の翻訳の結果をそれぞれ表 2，3 に示す．各表には，既存研究の報告値もあわせて記してある．なお，中-英翻訳については，本実験と同等程度の対訳コーパスを訓練データとして使用している研究の報告値を記してある．

[2]https://catalog.ldc.upenn.edu/LDC2012T21
[3]https://github.com/facebookarchive/NAMAS
[4]https://duc.nist.gov/duc2004/
[5]https://github.com/google/sentencepiece
[6]https://github.com/pytorch/fairseq
[7]DUC 2004 での設定と同様に，出力を 75 バイトまでで切り詰め，ROUGE 値を計算した．
[8]実際には，5.1 節で述べたように，LDC2003E14 の入手が不可能であり，本研究では対訳コーパスが 30 万文対ほど少ない．

| Method | R-1 | R-2 | R-L |
|---|---|---|---|
| Duan et al. (2019) [3] | 26.0 | 8.0 | 23.1 |
| Transformer による元文の翻訳 | 14.96 | 3.01 | 13.17 |
| パイプライン：翻訳 → 見出し文 | 18.82 | 5.08 | 17.00 |
| 提案手法（Zero-shot） | 19.20 | 4.92 | 17.01 |
| 提案手法 | **26.69** | **8.49** | **23.33** |

表 1: 中-英見出し文生成での各手法での ROUGE 値．

| Method | R-1 | R-2 | R-L |
|---|---|---|---|
| Rush et al. (2015) [8] | 28.18 | 8.49 | 23.81 |
| Li et al. (2017) [9] | 31.79 | 10.75 | 27.48 |
| Takase and Okazaki (2019) [7] | **32.29** | **11.49** | **28.03** |
| ベースライン | 29.59 | 9.98 | 25.82 |
| 提案手法 | 30.37 | 10.58 | 26.46 |

表 2: 英-英見出し文生成での提案手法の ROUGE 値と既存研究の報告値．

表 2 から，単言語の見出し文生成において，提案手法はベースラインよりも性能が良い，すなわち，翻訳，単言語の見出し文生成，言語横断見出し文生成の同時学習により，単言語の見出し文生成の性能が向上していることが分かる．一方で，提案手法の ROUGE 値は最高性能の値よりもわずかに低い．これは，デコーダの出力をサブワードとしたため，出力文字数が 75 バイトからずれてしまっていると考えられる．実際，Takase らはデコーダの出力単位を文字とし，出力長制御を行うことで，最高性能を達成している [7]．

中-英の翻訳では，表 3 のとおり，提案手法の性能はベースラインから大きく向上している．特に，MT02 と MT03 では，既存研究のようにモデルの構造に手を加えていないにも関わらず，既存研究のスコアと比較しても最高性能を達成しており，最先端の翻訳モデルと同等程度の性能であると考えられる．しかしながら，MT06 では既存研究と比べ，提案手法の BLEU 値は低い．これは，MT06 では Web 上の文書やニュース放送を含むため，本実験で用意できた対訳コーパスとはドメインが異なるためであると考えられる．入手できなかったために本実験で使用することのできなかった LDC2003E14 はニュース放送を収録したコーパスであり，これを用いることができれば MT06 での性能向上も可能であると推測している．また，提案手法は既存研究との組み合わせも可能であり，更に性能を上げられる余地があると考えられる．

提案手法である，出力タグの使用，および，構築した訓練データについて，それぞれどの程度性能向上に寄与したのかを明らかにするため，対訳コーパスのみを用いた学習を基点とし，各要素を追加した際の，中-英翻訳，英-英の見出し文生成，中-英の見出し文生成の性能を表 4 に示した．この表では言語横断見出し文生成を Zero-shot で行う状況かどうかで大別している．

まず，出力タグを付与することで，中-英 DUC の ROUGE-1 値が上昇することから，出力タグを用いて，出力を区別することにより，Zero-shot での言語横断見出し文生成の性能が向上することが分かる．中-英の対訳コーパスを言語横断見出し文生成の訓練コーパスとみなして学習した場合には，見出し文生成に関

| Method | MT02 | MT03 | MT04 | MT05 | MT06 |
|---|---|---|---|---|---|
| Wang et al. (2017) [10] | - | 39.35 | 41.15 | 38.07 | 37.29 |
| Cheng et al. (2018) [11] | 46.10 | 44.07 | 45.61 | 44.06 | 44.44 |
| Cheng et al. (2019) [12] | 48.13 | 47.83 | 49.13 | **49.04** | 47.74 |
| Zhang et al. (2019) [13] | - | 48.31 | **49.40** | 48.72 | **48.45** |
| ベースライン | 40.86 | 40.71 | 40.51 | 37.22 | 33.08 |
| 提案手法 | **48.64** | **50.23** | 48.45 | 47.84 | 40.79 |

表 3: NIST 中-英翻訳テストセットでの提案手法の BLEU 値と既存研究の報告値.

| 訓練データ | | | | | | | | | | | 英-英 | 中-英 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Clean | | Pseudo（言語横断見出し文生成） | | | Pseudo | MT02 | DUC | DUC |
| 出力タグ | 長さ制御 | 対訳 | 見出し文 | 対訳から | 見出し文から | 対訳を見出し文とみなす | （対訳） | BLEU | R-1 | R-1 |
| Zero-shot での言語横断見出し文生成 | | | | | | | | | | |
| | | ✓ | | | | | | 40.86 | - | 14.96 |
| | | ✓ | ✓ | | | | | 41.99 | 26.73 | 14.96 |
| ✓ | | ✓ | ✓ | | | | | 41.95 | 26.68 | 15.56 |
| ✓ | ✓ | ✓ | ✓ | | | | | 41.87 | 30.36 | 17.77 |
| ✓ | ✓ | ✓ | ✓ | | | ✓ | | 43.80 | 30.16 | 17.40 |
| ✓ | ✓ | ✓ | ✓ | | | | ✓ | 47.87 | 29.97 | 19.20 |
| ✓ | ✓ | ✓ | ✓ | | | ✓ | ✓ | **49.31** | 30.11 | 18.06 |
| 自動生成した言語横断見出し文生成の訓練データを使用 | | | | | | | | | | |
| ✓ | ✓ | ✓ | ✓ | ✓ | | | | 43.99 | 29.16 | 24.68 |
| ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | | 44.48 | 29.82 | 25.42 |
| ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | 45.26 | 29.68 | 26.05 |
| ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | 48.64 | **30.37** | **26.69** |

表 4: Transformer における各要素を用いた際の中-英翻訳（MT02）での BLEU 値と英-英見出し文生成（英-英 DUC），中-英見出し文生成（中-英 DUC）での ROUGE-1 値.

する性能はわずかに下がるが，中-英の翻訳性能は向上している．既に使用している学習事例と同一の事例を追加しているにも関わらず，性能が向上する理由としては，デコーダ側での位置エンコードが翻訳と見出し文生成で異なるため，位置エンコードに対する過学習が防がれていると推測される．また，逆翻訳で構築した疑似対訳コーパスを翻訳の訓練データとして用いた場合には中-英見出し文生成での ROUGE-1 値は 19.20 となり，表 1 における，パイプラインでの性能を上回る．さらに，この 2 つを組み合わせた際には翻訳の性能が最も高くなった．

逆翻訳や見出し文生成器を用いて構築した言語横断見出し文生成の訓練データを用いた場合には，向上幅の違いはあるが，データを追加するごとに性能が上がっている．最終的に，対訳コーパスと単言語の見出し文コーパスのみを訓練データとして用いた場合と比較して，言語横断見出し文生成では ROUGE-1 値が 10 以上，単言語の見出し文生成では ROUGE-1 値が 4 程度，中-英翻訳では BLEU 値が 6.5 程度向上した．対訳コーパスや単言語の見出し文コーパスを元に自動生成したデータを用い，翻訳と見出し文生成を同時に学習することで，劇的に性能が向上している．

## 6 おわりに

本論文では，出力を示すタグを入力に付与することにより，翻訳と見出し文生成，および言語横断見出し文生成を同時に学習する手法を提案した．提案手法は Zero-shot の言語横断見出し文生成においても，翻訳を行った後に見出し文生成を行うというパイプライン処理よりも高い性能を達成している．さらに，対訳コーパスや単言語の見出し文コーパスを元に自動生成したコーパスを用いて学習を行うことにより，言語横断見出し文生成だけでなく，翻訳，単言語の見出し文生成での性能を向上させることが可能であることを示した．提案手法は極めて単純な手法ながら，中-英の言語横断見出し文生成において最高性能を達成し，また，中-英の翻訳においても最高性能と同程度のスコアを達成した．

**謝辞** 本研究成果は，国立研究開発法人情報通信研究機構（NICT）の委託研究「多言語音声翻訳高度化のためのディープラーニング技術の研究開発」により得られたものです．

## 参考文献

[1] A. Leuski *et al.*, Cross-lingual c*st*rd: English access to hindi information (2003).
[2] X. Wan *et al.*, Cross-language document summarization based on machine translation quality prediction, in: ACL, 2010, pp. 917–926.
[3] X. Duan *et al.*, Zero-shot cross-lingual abstractive sentence summarization through teaching generation and attention, in: ACL, 2019, pp. 3162–3172.
[4] J. Zhu *et al.*, NCLS: Neural cross-lingual summarization, in: EMNLP-IJCNLP, 2019, pp. 3052–3062.
[5] M. Johnson *et al.*, Google's multilingual neural machine translation system: Enabling zero-shot translation, TACL 5 (2017) 339–351.
[6] A. Vaswani *et al.*, Attention is all you need, in: NIPS, 2017, pp. 5998–6008.
[7] S. Takase, N. Okazaki, Positional encoding to control output sequence length, in: NAACL-HLT, Minneapolis, Minnesota, 2019, pp. 3999–4004.
[8] A. M. Rush *et al.*, A neural attention model for abstractive sentence summarization, in: EMNLP, 2015, pp. 379–389.
[9] P. Li *et al.*, Deep recurrent generative decoder for abstractive text summarization, in: EMNLP, 2017, pp. 2091–2100.
[10] M. Wang *et al.*, Deep neural machine translation with linear associative unit, in: ACL, 2017, pp. 136–145.
[11] Y. Cheng *et al.*, Towards robust neural machine translation, in: ACL, 2018, pp. 1756–1766.
[12] Y. Cheng *et al.*, Robust neural machine translation with doubly adversarial inputs, in: ACL, 2019, pp. 4324–4333.
[13] W. Zhang *et al.*, Bridging the gap between training and inference for neural machine translation, in: ACL, 2019, pp. 4334–4343.
[14] I. Caswell *et al.*, Tagged back-translation, in: WMT, 2019, pp. 53–63.
[15] T. Kudo, Subword regularization: Improving neural network translation models with multiple subword candidates, in: ACL, 2018, pp. 66–75.
[16] M. Ott *et al.*, fairseq: A fast, extensible toolkit for sequence modeling, in: NAACL-HLT, 2019, pp. 48–53.