T2R2 東京科学大学 リサーチリポジトリ
Science Tokyo Research Repository

論文 / 著書情報
Article / Book Information

題目(和文)	ニューラル文書要約の高度化に関する研究
Title(English)	
著者(和文)	牧野拓哉
Author(English)	Takuya Makino
出典(和文)	学位:博士(工学), 学位授与機関:東京工業大学, 報告番号:甲第11676号, 授与年月日:2020年12月31日, 学位の種別:課程博士, 審査員:奥村 学,船越 孝太郎,熊澤 逸夫,高村 大也,篠崎 隆宏
Citation(English)	Degree:Doctor (Engineering), Conferring organization: Tokyo Institute of Technology, Report number:甲第11676号, Conferred date:2020/12/31, Degree Type:Course doctor, Examiner:,,,,
学位種別(和文)	博士論文
Type(English)	Doctoral Thesis

Powered by T2R2 (Science Tokyo Research Repository)

ニューラル文書要約の高度化に関する研究

東京工業大学

大学院総合理工学研究科

工学院 情報通信系 情報通信コース専攻

博士論文

指導教員: 教授 奥村学

牧野拓哉

2020 年 10 月

審査委員会

主査:　教授 奥村学

副査:　教授 熊澤逸夫

副査:　教授 高村大也

副査:　准教授 船越孝太郎

副査:　准教授 篠崎隆宏

概要

文書要約は与えられた入力文書を簡潔に言い表した短い文書や文を作成する課題である．要約があることによって我々は文書全文を読むことなく，情報の要旨を容易に把握できる．一方で，要約作成は人手に頼っており，要約作成者は要約対象の文書の内容を把握し，重要な情報を誤りなく，漏れなく含めるように要約を作成する必要がある．そのため，要約作成にかかるコストは少なくなく，文書自動要約技術による要約作成が求められている．

文書要約は要約対象の文書数によって別の課題として扱われる．一つの文書を要約する場合は単一文書要約，複数の文書を要約する場合は複数文書要約と呼ばれる．たとえば，新聞記事の見出しや電光掲示板に流れる短い要約は単一文書要約の応用例，Web 検索エンジンに発行したクエリに対して得られる検索結果のまとめは複数文書要約の応用例として挙げられる．新聞記事の見出しや電光掲示板に流れる短い要約の作成作業は日々発生するため，文書要約技術が求められることが多い．このような背景から，本論文では単一文書要約を対象とする．

文書要約手法は抽出型手法と生成型手法に大別される．抽出型手法は原文書中の主に単語や文を抽出することで要約を作成する手法であり，生成型手法はテキストを生成することで要約を作成する手法である．単一文書要約の応用事例として挙げた見出しや電光掲示板の要約はデバイスに応じて要約長に制約がある場合があり，制約長に収まるように原文書の情報を言い換えて作成されることが多い．そのため，本論文では言い換えが可能である生成型手法の中でも，特に高い要約性能が得られる生成型ニューラル要約モデルの高度化に焦点を当てる．

生成型ニューラル要約モデルに関する研究は盛んに取り組まれているものの，依然として課題が残されている．本論文ではその中でも重要な課題である，要約長制約を超える要約の生成の抑制，要約特有な単語を多く生成するための生成用語彙構築手法の改善の二つの課題に取り組む．一つ目の課題においては，既存の要約長制御が可能な生成型ニューラル要約モデルは要約長制約付近の長さの要約を生成することができるが，要約長制約を超えた要約を生成することも少なくない．実際の要約作成ではたとえばデバイスの大きさや要約を配信する媒体に応じて要約長制約があり，それを超えないように要約を作成する必要がある．本論文は要約長制約を超えないように品質の高い要約の生成をおこなうための学習方法を提案する．二つ目の課題においては，生成型ニューラル要約モデルに対する既存の生成用語彙構築手法では，要約特有の単語の生成割合が低下し，要約精度に影響しうるという課題である．生成

型ニューラル要約モデルの中でも，原文書からの単語のコピーが可能なモデルが高い要約精度を示しているが，そのようなモデルに対して従来の生成用語彙構築手法では，冗長な単語が語彙に登録され，生成可能な要約特有の単語の数が減り，要約精度の低下に影響する恐れがある．本論文は学習データ中で対となる原文書と要約を比較し，要約のみに出現する単語を対象とすることで要約特有な単語を生成用語彙に含める語彙構築手法を提案する．

提案手法の有効性を複数のデータセットにおける実験によって定量的に示すとともに，今後残された課題についてまとめる．

目次

表目次

図目次

第1章　序論

1.1　背景

文書要約は与えられた文書に対する簡潔な文書や文を作成する課題である．新聞社をはじめとして，ニュースを配信する企業では読者に簡潔に情報を伝えるため，ニュース記事の概要を見出しや短い文書として作成し，電光掲示板や小型携帯端末を介して世の中に配信している．この短い文書を作成する作業は文書要約として捉えることができる．ニュースを即座に世の中の様々な媒体に配信するため，テレビ局や新聞社などの企業では要約作成作業が昼夜問わず発生しうる．しかしながら，要約作業は人手に大きく頼っており，要約作成にかかるコストは少なくない．そのため，効率的に要約を作成するための技術が求められている．このような背景から，本論文は文書要約の研究に取り組む．

文書要約は，入力の文書数によって，異なる課題として研究されている．入力の文書 (原文書) 数が 1 つであれば単一文書要約 (Luhn 1958)，複数であれば複数文書要約 (McKeown and Radev 1995) と別の研究課題として扱われる．単一文書要約は文書に対する見出しの生成や新幹線の電光掲示板に流れる短い要約が応用として挙げられる．複数文書要約は Web 検索エンジンに発行したクエリに対して得られる検索結果のまとめの作成が挙げられる．これら二つの研究課題のうち，本論文では，新聞社において日々おこなわれている記事の見出し作成や電光掲示板向けの要約作業といった単一文書要約を対象とする．

文書要約手法は抽出型手法と生成型手法に大別される．抽出型手法は原文書中の主に単語や文を抽出することによって要約を出力する手法である．原文書の情報の一部をそのまま要約として生成するため，誤った情報を要約に含める恐れが少ない一方で，短い要約長制約内での要約生成，語順変化および，言い換えへの対応が難しい．生成型手法はテキストを生成することで要約を出力する手法である．言い換えや語順の変化への対応が可能な一方で，原文書の情報を誤って言い換えてしまい，原文書と意味が異なる要約を生成する可能性がある．単一文書要約の応用例として挙げた見出しや電光掲示板に流れる要約は，デバイスの大きさにより要約長に制約がある場合が多く，制約内で要約に多くの情報を含めるために原文書の情報をできるだけ短く言い換えるように編集されることがある．そのため，見出しやより簡潔な要約の作成においては，生成型要約手法が適しているといえる．

生成型要約手法は，機械翻訳と同じく入力テキストに対応するテキストを生成する問題として扱うことが可能であり，機械翻訳で用いられるアプローチが適用されてきた．たとえば，ルールベースによる手法 (Fum, Guida, and Tasso 1986) が提案されたが，ルールベースでは考慮しなければならないルールの数が多いという課題があった．そこで，統計的機械翻訳と同様に，原文書中の単語と要約中の単語の翻訳確率や，要約中の単語の並びを考慮した手法 (Witbrock and Mittal 1999) が提案された．しかしながら，学習データの不足や，要約器の内部で利用されるモジュールは個別に学習されており，要約モデル全体としての学習がされないといったことから，要約精度の改善に課題があった．その後，機械翻訳分野においては，大規模な学習データの構築が進んだこと，計算機資源の性能向上を背景として，翻訳モデル全体として学習が可能なニューラルネットワークに基づく機械翻訳手法の研究が進んだ．その結果，ニューラルネットワークに基づく機械翻訳により従来の統計的機械翻訳を超える精度が得られることが示された．機械翻訳での成功をきっかけに，文書要約においても，大規模な文書要約データが構築され，生成型のニューラル要約手法 (Rush, Chopra, and Weston 2015) の有用性が示された．しかしながら，この手法は生成用語彙語彙には存在しない単語を要約に含められないという課題があった．そのため，文書要約における，原文書の一部がそのまま要約にも出現するという特徴にうまく対応できなかった．この課題に対処するために，生成用語彙に存在しない単語であっても原文書に出現する単語をコピーして要約に含めることが可能な，コピー機能を伴う生成型ニューラル要約モデルが提案され，高い要約性能が得られることが報告されている (Gulcehre, Ahn, Nallapati, Zhou, and Bengio 2016; Huang, Wu, and Wang 2020; See, Liu, and Manning 2017; Wang, Zhao, Jia, Li, and Liu 2019a).

このように，ニューラル要約手法では大きな改善がみられたが，既存手法には二つの重要な課題が残されている．一つ目の課題は生成型のニューラル要約手法の要約長制御に関する研究がされているものの，要約長制約を超えた要約の生成が少なくないという課題である．実際の要約作成業務では要約長制約内で要約を作成する必要があるため，要約長制約を超える要約が多いほど，人手で自動要約結果を書き換えて要約長制約に収まるように修正する必要がある．二つ目は，従来の生成用語彙構築手法では要約特有の単語の生成割合が低下し，要約精度に影響しうるという課題である．従来手法は要約に高頻度で出現する単語の集合を生成用語彙として構築する．コピー機能を伴う生成型ニューラル要約手法では原文書からの単語のコピーを用いて要約を作成可能であることと，要約における高頻度語は原文書でも高頻度であることから，生成用語彙には冗長な単語が含まれているといえる．生成用語彙サイズは一般的に計算機のメモリや処理速度の観点から高頻度語に制限することが多い．そのため，

コピーによって要約に含めることができる高頻度な単語が生成用語彙に登録されることで，要約特有の単語が生成用語彙に登録されず，生成可能な要約特有の単語の数が減ることで要約精度に影響を及ぼす.

1.2 本論文の貢献

本論文の貢献は前述の二つの課題を解決するための方法を提案し，複数のデータセットにおける実験によって有効性を示すことである.

要約長制約内で要約を生成するための学習方法 前述したように，既存のニューラル要約手法では指定した要約長に近い長さの要約を生成することができるが，その長さを超えた要約を生成することも少なくない．また，要約の品質においても改善の余地がある．そこで本論文では要約長制約を超えた要約の生成を削減しつつ，要約長制約内において要約の品質を向上させる学習方法を提案する．既存の学習方法は参照要約の長さを要約長制約として，参照要約の生成確率を向上させるものであるため，学習中に要約長制約を超えた要約を生成することがない．そのため，要約長制約を超えた要約の生成確率を減少させる方法が自明ではない．提案手法は要約長制約内の要約および要約長制約を超えた要約をサンプリングによって生成し，要約モデルの良さを評価する．この良さは，要約長制約内で品質の高い要約が多くサンプリングされるほど高くなり，要約長制約を超えた要約がサンプリングされるほど低くなるように計算される．結果として，提案手法で要約モデルを学習させることで，要約長制約内で品質の高い要約ほど生成確率が向上する一方で，要約長制約を超えた要約の生成確率は減少する．CNN/Daily Mail コーパスおよび毎日新聞コーパスを用いた実験によって提案手法は要約長制約を超えた要約の生成を減らしつつ，ROUGE 値が改善することを示す.

要約特有の単語を多く生成するための生成用語彙構築方法 コピー機能を伴う生成型ニューラル要約モデルは，高い要約性能が得られることが報告されているが，既存の語彙構築方法を用いて生成用語彙を構築すると，要約特有の単語の生成割合の低下に影響し，要約精度に影響しうるという課題が残されている．従来手法は学習データの参照要約における高頻度語の集合を生成用語彙とする方法である．コピー機能を伴う生成型ニューラル要約モデルは，原文書からの単語のコピーが可能である点と，要約における高頻度語は原文書でも高頻度語であるという点から，従来手法に基づいて構築された生成用語彙には冗長な単語が含まれているといえる．本論文では，学習データ中の原文書と要約

の対を比較することで，原文書からの抽出によって対処可能な単語の生成用語彙への登録を除きつつ，要約のみに出現している単語を対象に語彙を構築する方法を提案する．CNN/Daily Mail コーパスおよび NEWSROOM コーパスを用いた実験によって，既存の生成用語彙構築手法よりも高い ROUGE 値を示しつつ， 要約特有の単語を多く生成できることを示す．

1.3　本論文の構成

本章以降の構成を述べる．2 章では本研究に関連する既存のニューラル要約手法を中心に説明する．3 章では本研究で用いるニューラルネットワークについて説明する．4 章では 1 つ目の貢献である要約長制約内での要約を生成するためのニューラル要約手法の学習方法について説明する．5 章では 2 つ目の貢献である要約特有の単語を多く生成するための生成用語彙構築手法について説明する．6 章では本論文の貢献についてまとめるとともに，今後の課題を議論する．

第2章　関連研究

本章では抽出型手法と生成型手法における文書要約研究の発展について述べる．次に本論文と関連する既存の生成型のニューラル要約モデルについて述べる．特に要約長を制御する生成型ニューラル要約モデルの関連研究，生成型ニューラル要約モデルの語彙構築手法および，コピー機能を伴う生成型ニューラル要約モデルの関連研究を中心に説明する．また，評価方法についても説明する．

2.1　抽出型手法

2.1.1　文分類器に基づく要約手法

単一文書要約において，文の出現位置などを素性として文が要約に含まれるかどうかを予測する文分類器を利用する方法が提案された．ナイーブベイズ分類器 (Kupiec, Pedersen, and Chen 1995)， 決定木 (Lin 1999)，最大エントロピー法 (Osborne 2002)，SVM (Hirao, Isozaki, Maeda, and Matsumoto 2002) による文分類器が適用されてきた．

2.1.2　組み合わせ最適化問題に基づく要約手法

複数文書要約では同一の事柄に関する記事が複数入力されるため，入力には同じような情報を持つ文が複数存在する．このような冗長性を排除するため，既に要約として抽出された文との類似度を見て次の文を要約に含めるかどうかを逐次的に判定するための仕組みが提案されている (Goldstein, Mittal, Carbonell, and Kantrowitz 2000)．Filatova and Hatzivassiloglou (2004) は文を概念単位 (たとえば単語) の集合とみなし，重要な概念単位をなるべく多く被覆するように文の組み合わせを抽出する問題を最大被覆問題として定式化した．文を概念単位に分解し，かつ概念単位の重要度を判定する必要がある．McDonald (2007) は文に対して重要度を付与し，それらの文を要約に含めるかどうかをナップサック問題として定式化した．また，冗長性を排除するために選択された文の間の類似度が高くならないよう考慮した．Takamura and Okumura (2009) は文間の含意関係を考慮して，より多くの文を含意できるような文の組み合わせを施設配置問題として定式化した．厳密解を得ることができ，また含意関係という非対称性を考慮することができるため柔軟といえる．一方で要約に含まれた文の一貫性を考慮することができない．

単一文書要約においても組み合わせ最適化問題として文を抽出する方法が提案されている．複数文書要約と異なり，単一文書要約では選択する情報の一貫性に焦点を当てた研究が多い．Hirao et al. (2013), Kikuchi et al. (2014) は修辞構造理論に基づいて原文書を木として表現し，部分木の組み合わせを選択する問題をナップサック問題として定式化した．要約に含まれる文の内容の一貫性を考慮することができるが，要約の精度が談話構造解析器の精度に依存する．Durrett et al. (2016) は文の圧縮まで考慮した整数計画問題として定式化した．構文木に基づく圧縮や照応情報に基づく圧縮を考慮できる．また部分木の重要度は Structured SVM によって ROUGE 値が向上するように学習される．Parveen and Strube (2015), Parveen et al. (2015) は原文書中の文と，それらが表すトピックからなる二部グラフを原文書から構築し，要約の一貫性を考慮した文の組み合わせの選択を整数計画問題として定式化した．トピックは Latent dirichlet allocation (Blei, Ng, and Jordan 2003) に基づいて推定するため，談話構造解析器を必要としない．

2.1.3　系列ラベリングに基づく要約手法

単一文書要約においては系列ラベリングに基づく要約手法が提案されている．系列ラベリングに基づく要約手法では，文書を木に変換する構文解析器を必要とせず，機械学習に基づいて局所的な一貫性を学習するアプローチが盛んに取り組まれてきた．系列ラベリングには，隠れマルコフモデル (Barzilay and Lee 2004)，条件付確率場 (Shen, Sun, Li, Yang, and Chen 2007)，隠れ半マルコフモデル (Nishikawa, Arita, Tanaka, Hirao, Makino, and Matsuo 2014)，ニューラルネットワーク (Cheng and Lapata 2016; Nallapati, Zhai, and Zhou 2017) が文を単位とする系列ラベリングに用いられてきた．また，単語単位で系列ラベリングをモデル化する研究もある．Filippova et al. (2015) はニューラルネットワークに基づいて原文書の単語を要約として抽出する方法を提案した．他に，ニューラルネットワークに基づいて文を抽出したのちに，別のニューラルネットワークを用いて構文木上の部分木を抽出する文の抽出と圧縮をおこなう手法も提案されている (Xu and Durrett 2019).

2.2　生成型手法

2.2.1　統計的要約手法

統計的機械翻訳で提案された手法を生成型要約に適用し，見出しの生成をする研究がなされてきた (Alfonseca, Pighin, and Garrido 2013; Banko, Mittal, and Witbrock 2000; Witbrock and Mittal 1999)．他に人手で要約の骨子となるテンプレートを用意し，必要な情報を既存の情報

抽出器によって埋めることで要約を出力する研究がある (Genest and Lapalme 2012).

2.2.2 語彙からの単語生成に基づく生成型ニューラル要約手法

Rush et al. (2015) は，機械翻訳におけるニューラル翻訳モデルと同様に，事前に構築した語彙から順に単語を選択することで見出しを生成する生成型手法を提案した．その後，文書要約に特化した手法として，原文書の重要な箇所をニューラルネットワークによって自動で推定する方法 (Gehrmann, Deng, and Rush 2018; Zhou, Yang, Wei, and Zhou 2017) や，同一の単語の繰り返し生成を減らすためのモデルも提案されている (Kiyono, Takase, Suzuki, Okazaki, Inui, and Nagata 2018; Suzuki and Nagata 2017)．また，語彙に出現しなくても，原文書に出現する単語を要約へ含められるように，抽出型手法と生成型手法の組み合わせが提案されている (Gulcehre et al. 2016; See et al. 2017).

生成型ニューラル要約モデルは対象ドメインの拡大やアルゴリズムの改良など，様々な観点での研究が進められているが，本論文の一つ目の貢献である，要約長制約内で要約を生成するための学習方法は，要約長制御可能な生成型ニューラル要約モデルおよび生成型ニューラル要約モデルの学習方法と関連する．また，本論文の二つ目の貢献である，語彙構築手法は，コピー機能を伴う生成型ニューラル要約モデルと関連する．

要約長制御可能な生成型ニューラル要約モデル

生成型ニューラル要約モデルの要約長を制御する手法はいくつか提案されているが，既存研究ではニューラルネットワークのアーキテクチャに焦点を当てたものである (Hitomi, Taguchi, Tamori, Kikuta, Nishitoba, Okazaki, Inui, and Okumura 2019; Kikuchi, Neubig, Sasano, Takamura, and Okumura 2016; Liu, Luo, and Zhu 2018; Takase and Okazaki 2019)．一方で提案手法は要約長制約内で要約品質を向上させるための学習手法であり，要約長を制御可能な任意の生成型ニューラル要約モデルに適用可能である．

生成型ニューラル要約モデルの学習手法

強化学習や minimum risk training (MRT) のように評価尺度に対してモデルを最適化する方法を生成型ニューラル要約モデルへ適用することで要約性能が向上することが報告されている (Ayana, Shen, Lin, Tu, Zhao, Liu, and Sun 2017; Chen and Bansal 2018; Paulus, Xiong, and Socher 2018; Ranzato, Chopra, Auli, and Zaremba 2015)．提案手法は MRT を生成型ニューラル要約モデルへ適用するという点で Ayana et al. (2017) と類似する．提案手法の特徴は (1) ROUGE を評価する際は要約長制約内の自動要約結果のみを活用する点， (2) ROUGE 値に関

わらず要約長制約を超えた自動要約結果には罰則を与える点である．

コピー機能を伴う生成型ニューラル要約モデル

コピー機能を伴う生成型ニューラル要約モデルは生成用語彙からの単語の出力に加えて，原文書中の単語を要約の一部として抽出するコピー機能を伴うニューラル要約モデルである (Gulcehre et al. 2016; Huang et al. 2020; See et al. 2017; Wang et al. 2019a)．近年は大量のラベルなしテキストコーパスを用いた事前学習によって要約の精度改善がおこなわれているが (Lewis, Liu, Goyal, Ghazvininejad, Mohamed, Levy, Stoyanov, and Zettlemoyer 2020; Raffel, Shazeer, Roberts, Lee, Narang, Matena, Zhou, Li, and Liu 2019)，コピー機能を伴う生成型ニューラル要約モデルにおいても，高い要約精度が報告されており (Huang et al. 2020; Wang et al. 2019a)，コピー機能は依然として重要な役割を担っている．

本論文と同様に生成型ニューラル要約モデルの生成用語彙構築に着目した研究として，入力と関連する小さな生成用語彙を，事前に構築した大きな生成用語彙から動的に構築する手法が提案されている (Gulcehre et al. 2016)．一方で，提案手法は事前に構築する生成用語彙に対して焦点を当てている．そのため，提案手法はこの手法と組み合わせて利用することが可能である．(Zhang, Yao, and Yan 2018) は原文書からの単語のコピーのみで要約を作成する方法で CNN/Daily Mail において高い要約精度が報告されている pointer-generator (See et al. 2017) と同等の ROUGE 値が得られることを示した．ただし，この手法では言い換えや，要約特有の単語を生成することはできない．

(Gehrmann et al. 2018) は原文書の単語に対して，重要かそうでないかを系列ラベリングによって事前に二値分類し，その結果を用いて pointer-generator のアテンション確率計算時に，重要でないと分類された単語に対してはアテンション確率を 0 とする手法を提案した．(Wang et al. 2019a) は Transformer に対して pointer-generator を適用し，さらに事前学習を用いる手法を提案した．(Shen, Zhao, Su, and Klakow 2019) はコピー確率計算時に，類義語に対してもコピー確率を考慮できるように pointer-generator を拡張した．(Huang et al. 2020) は事前学習済みの RoBERTa (Liu and Lapata 2019) による単語のエンコードに加えて，OpenIE によって原文書から取得した 3 つ組をエンコードし，得られた結果を pointer-generator で活用することで精度が改善されることを示した．提案手法は生成用語彙構築方法に関するものであり，これらの研究と組み合わせることでさらなる精度改善が期待される．

2.3 評価方法

要約の評価には自動評価および人手評価が用いられることが多い．

ROUGE (Lin 2004) は自動生成された要約と参照要約 (集合) との類似度を測る自動評価尺度である．類似度はさまざまな単位に基づいて計算されるが，良く用いられるのは連続する n 個の単語列 (n-gram) に基づく ROUGE-N や最長共通部分列に基づく ROUGE-L である．ROUGE-N で良く用いられる N は 1，2 であるため，局所的な一致に基づく評価尺度といえる．一方で ROUGE-L は大域的な一致に基づく評価尺度といえる．

ここで，参照要約 r および自動生成された要約 s はともに単語列として表されるものとする．ROUGE-N は，自動生成された要約に含まれる n-gram が参照要約にも含まれるほど高い値をとる．具体的には，参照要約 r が与えられたとき，自動生成された要約 s に対して次の様に計算する．

$$\text{ROUGE-N}_R = \frac{\sum_{n \in \texttt{ngram}(r) \cap \texttt{ngram}(s)} \min(\texttt{count}(n, r), \texttt{count}(n, s))}{\sum_{n \in \texttt{ngram}(r)} \texttt{count}(n, r)} \tag{1}$$

$$\text{ROUGE-N}_P = \frac{\sum_{n \in \texttt{ngram}(r) \cap \texttt{ngram}(s)} \min(\texttt{count}(n, r), \texttt{count}(n, s))}{\sum_{n \in \texttt{ngram}(s)} \texttt{count}(n, s)} \tag{2}$$

$$\text{ROUGE-N}_F = \frac{\text{ROUGE-N}_P \cdot \text{ROUGE-N}_R}{(1-\alpha)\text{ROUGE-N}_P + \alpha\text{ROUGE-N}_R} \tag{3}$$

ただし，$\texttt{ngram}(\cdot)$ は単語列から n-gram の集合を返す関数とする．$\texttt{count}(n, r)$ と $\texttt{count}(n, s)$ はそれぞれ n-gram n が参照要約 r に出現した回数，自動生成された要約 s に出現した回数を返す関数とする．α はハイパーパラメータであり，適合率 (P) と再現率 (R) のバランスをとる．本論文では既存研究で良く用いられる ROUGE-1 および ROUGE-2 を評価指標とする．

また ROUGE-L は，最長共通部分列が長いほど高い値をとる．具体的には次の様に計算する．

$$\text{ROUGE-L}_R = \frac{\texttt{LCS}(r, s)}{\sum_{n \in r} \texttt{count}(n, r)} \tag{4}$$

$$\text{ROUGE-L}_P = \frac{\texttt{LCS}(r, s)}{\sum_{n \in s} \texttt{count}(n, s)} \tag{5}$$

$$\text{ROUGE-L}_F = \frac{\text{ROUGE-L}_P \cdot \text{ROUGE-L}_R}{(1-\alpha)\text{ROUGE-L}_P + \alpha\text{ROUGE-L}_R} \tag{6}$$

ただし $\texttt{LCS}(r, s)$ は参照要約の単語列と自動生成された要約の単語列の間の最長共通部分列の長さを返す関数とする．

要約長制約があるデータセットにおいては再現率 (R) が良く用いられる．これは限られた制約の中で原文書の情報を被覆できている方が良い要約であるという考えに基づいている．要

約長制約がない場合は再現率と適合率の調和平均 (F) が良く用いられる．これは原文書の重要な情報を被覆しつつも，重要でない情報は被覆しない方が良い要約であるという考えに基づいている．本論文では ROUGE-N および ROUGE-L の F 値を計算する際は $\alpha = 0.5$ とした．

ROUGE は表層の一致に基づいて値が計算されるため，意味的に類似している要約に対する妥当な評価ができない．また可読性についての妥当な評価も難しい．そこで既存研究では ROUGE による評価に加えて，人手による主観評価もおこなわれることが多い．本論文でも同様に ROUGE および人手による主観評価を実施する．

第3章　生成型ニューラル要約モデルの概要

本章では，生成型ニューラル要約モデルの概要を述べる．ここで原文書は単語列$\mathbf{x} = \langle x_1, ..., x_M \rangle$で表され，要約は$\mathbf{y} = \langle y_1, ..., y_N \rangle$で表されるものとし，学習データ$D$は原文書と要約の対の集合とする．

生成型ニューラル要約モデルはエンコーダとデコーダからなる．エンコーダは原文書を読み込み，固定長のベクトルを隠れ状態として出力する．デコーダはエンコーダが出力した隠れ状態ををを考慮して単語を生成する処理を繰り返すことで要約を作成する．単語は学習データから事前に構築した生成用語彙Vに含まれる単語を対象に生成確率を計算し，確率が最大となるものを出力する．

$$p(y_t|\mathbf{x}) = \frac{\exp(\mathbf{W}_{y_t}\mathbf{s}_t + \mathbf{b}_{y_t})}{\sum_{\bar{y}\in V}\exp(\mathbf{W}_{\bar{y}}\mathbf{s}_t + \mathbf{b}_{\bar{y}})} \tag{7}$$

ただし，$\mathbf{W}_{y_t}$はパラメータ行列$\mathbf{W} \in \mathbb{R}^{|V|\times h}$の$y_t$に対応する行ベクトル，$\mathbf{b}_{y_t}$はパラメータベクトル$\mathbf{b} \in \mathbb{R}^{|V|}$の$y_t$に対応するスカラーを表す．$\mathbf{s}_t$はデコード時の時刻$t$におけるデコーダの隠れ状態を表す．

生成型ニューラル要約モデルは原文書の単語列が長いほど，文脈情報を隠れ状態中で保持するのが難しくなり，結果として要約性能が低下する傾向にある．その対策としてデコーダは attention 機構 (Luong, Pham, and Manning 2015) を持つ．attention 機構では現在のデコーダの隠れ状態$\tilde{\mathbf{s}}_t$とエンコーダの各単語に対する隠れ状態$\mathbf{h}_1, ..., \mathbf{h}_i, ..., \mathbf{h}_M$との関連度を内積や多層パーセプトロンなどによって計算する．関連度を計算後，$\tilde{\mathbf{s}}_t$に加えて，得られた関連度で重みを付けてエンコーダの隠れ状態を考慮した隠れ状態$\mathbf{s}_t$を出力する．生成型ニューラル要約モデルの内部で利用するニューラルネットワークをf，attention 機構をgとすると，時刻tにおける隠れ状態$\mathbf{s}_t$は次の様に計算する．

$$\tilde{\mathbf{s}}_t = f(\tilde{\mathbf{s}}_{<t}, y_{t-1}) \tag{8}$$

$$\mathbf{s}_t = g(\tilde{\mathbf{s}}_t, \mathbf{h}_1, ..., \mathbf{h}_i, ..., \mathbf{h}_M) \tag{9}$$

ただし，y_{t-1}は直前に生成した単語とする．$\tilde{\mathbf{s}}_{<t} = \tilde{\mathbf{s}}_1, ..., \tilde{\mathbf{s}}_{t-1}$とする．どこまで遡って過去の隠れ状態を利用するかは，内部で用いられるニューラルネットワークによって異なる．たとえば，fが Long Short Term Memory (LSTM) (Hochreiter and Schmidhuber 1997) であれば直

前の隠れ状態 $\tilde{\mathbf{s}}_{t-1}$ のみを活用し，畳み込みニューラルネットワーク (Gehring, Auli, Grangier, Yarats, and Dauphin 2017) であれば，固定長 K 分だけ過去の隠れ状態 $\tilde{\mathbf{s}}_{t-K}, ..., \tilde{\mathbf{s}}_{t-1}$ を利用する．attention 機構からは時刻 t においてどれくらいの重みで原文書の単語 x_i を参照するかを表す attention 確率を得られる．抽出型と生成型の組み合わせ手法では，attention 確率を利用して原文書の単語に対する生成確率を計算するため，原文書に出現する単語を要約に含めることもできる．

生成型ニューラル要約モデルに対してよく用いられる学習方法は対数尤度の最大化 (Maximizing Log-likelihood Estimation; MLE) であり，次の式に基づいて参照要約の生成確率を最大化する．

$$L(\Theta) = \sum_{(\mathbf{x},\mathbf{y})\in D} \log p(\mathbf{y}|\mathbf{x};\Theta) \tag{10}$$

$$= \sum_{(\mathbf{x},\mathbf{y})\in D} \sum_{t=1}^{N} \log p(y_t|\mathbf{x};\Theta) \tag{11}$$

ただし Θ は学習対象となるパラメータ集合とする．

第4章　要約長制約下における生成型ニューラル要約モデルの学習

本章では，本論文の一つ目の貢献である，要約長制約下において品質の高い要約を生成するための生成型ニューラル要約モデルの学習方法について述べる．

4.1　研究概要

要約モデルが自動生成した要約は原文書の重要な情報を含むとともに，要約の長さを制御できることが重要である．たとえば，要約の長さはスマートフォンや電光掲示板などのデバイスに応じて要約長制約が存在するためである．人手で要約を作成する際も，要約長制約に収まるように，原文の情報を言い換えて要約が作成される．

文書要約研究においては抽出型手法と生成型手法が盛んに研究されている．抽出型手法は原文書中の文や単語などを組み合わせて要約として出力する手法である．一方で生成型手法は新しく文を生成することで要約として出力する手法である．それゆえ，生成型手法は言い換えや語順変化に対応できる．

要約長制御が可能な生成型手法は要約モデルに焦点を当てた既存研究がある (Kikuchi et al. 2016; Fan, Grangier, and Auli 2018; Liu et al. 2018; Takase and Okazaki 2019; Hitomi et al. 2019)．既存研究はデコード時において要約モデルに要約長制約を与える手法を提案している．これらの手法は少なくとも二つの点で改善の余地がある．一つ目は自動生成した要約が要約長制約を超える割合が少なくないという点である．これは要約モデルの学習時に参照要約のみを利用するためである．もう一つは要約性能に改善の余地があるという点である．これは既存研究の学習方法として用いられる対数尤度の最大化 (Maximizing Log-likelihood Estimation; MLE) が ROUGE に対して要約モデルを学習するものではないためである．

MRT (Och 2003) は任意の評価尺度に対してモデルを最適化できる．文書要約においては見出しの生成において，参照要約との類似度に基づく評価尺度である ROUGE 値に対して生成型ニューラル要約モデルを最適化する方法が提案されている (Ayana et al. 2017)．しかしながら MRT を要約長制約がある要約課題に適用する方法は明らかではない．そこで本論文では要約長制約がある要約課題に対して生成型ニューラル要約モデルを最適化する大域的な学習方

法を提案する．

CNN/Daily Mail および毎日新聞を用いて実験をおこない，提案手法は MLE よりも高い ROUGE 値となりつつ，要約長制約を超える要約を生成する割合が従来手法が 20%から 50%であったのに対して，6.7%から 7.35%まで削減できることを示す．さらに，自動要約結果に対する人手による後編集実験をおこない，要約長制約内の要約の方が，要約長制約を超えた要約よりも後編集にかかる時間が 30%から 40%短くなることを示す．

4.2　要約長制御モデル

本節では提案手法を適用する二つの要約長制御モデルについて述べる．既存研究において，これらのモデルは生成型ニューラル要約モデルの学習でよく用いられる MLE によって学習される．

4.2.1　LSTM に基づく手法

Kikuchi et al. (2016) はデコード時に要約長制約までの残り長さを考慮する LSTM の変種 (*LenEmb*; LE) を提案した．要約長制約までの残り長さは，学習時は参照要約の長さ，生成時はハイパーパラメータで与えられた値で初期化される．デコード時の各ステップにおいて，生成された単語の長さが，要約長制約までの残り長さから差し引かれる．本論文では LE を pointer-generator (See et al. 2017) のデコーダ部に用いる．pointer-generator はコピー機能を伴う生成型ニューラル要約モデルの一種であり，高い要約性能が得られることが報告されている．

4.2.2　畳み込みニューラルネットワークに基づく手法

Liu et al. (2018) は要約長制御可能な畳み込みニューラルネットワークを提案した．このモデルは通常の畳み込みニューラルネットワーク (CNN) に基づくエンコーダ・デコーダにおけるデコーダ部で要約長制約を考慮する．CNN に基づくエンコーダ・デコーダでは単語は単語埋め込みと位置埋め込みの結合として表現される (Gehring et al. 2017)．デコード時には隠れ状態に要約長制約をスカラーとして掛けることで現在までに生成した要約と要約長制約を考慮する．このモデルも LSTM に基づくモデルと同様に生成した要約が要約長制約に近づくと `EOS` を生成するように学習する．

ここで LSTM に基づく手法が文字数単位での要約長制御であり，CNN に基づく手法は単語数単位での要約長制御であることに注意してほしい．

$$\Delta(\boldsymbol{y}, \boldsymbol{y}') = -\text{ROUGE1}(\\ \langle \text{malaysia, markets, closed, for, holiday} \rangle, \\ \langle \text{markets in, malaysia, closed, for, holiday} \rangle \\ )\text{=-1.0}$$

(a) 従来手法の ROUGE 値のみに基づく $\Delta(\mathbf{y}, \mathbf{y}')$ の例.

$$\text{trim}(\boldsymbol{y}', \text{byte}(\boldsymbol{y})): \\ \langle \text{markets, in, malaysia, closed, for} \rangle$$

$$\tilde{\Delta}(\boldsymbol{y}, \boldsymbol{y}') = -\text{ROUGE1}(\\ \langle \text{malaysia, markets, closed, for, holiday} \rangle, \\ \langle \text{markets, in, malaysia, closed, for} \rangle \\ ) + \max(0, 38 - 35) = -0.8 + 3.0 = 2.2$$

(b) 提案手法の $\tilde{\Delta}(\mathbf{y}, \mathbf{y}')$.

Figure 1: 従来手法の $\Delta(\mathbf{y}, \mathbf{y}')$ と提案手法の $\tilde{\Delta}(\mathbf{y}, \mathbf{y}')$ を ROUGE-1 の再現率に基づて計算する例．参照要約は “malaysia markets closed for holiday” で自動生成された要約は “markets in malaysia closed for holiday” とする．参照要約の長さは 38，自動生成された要約の長さは 35 である．

4.3 既存の学習方法

本節では生成型ニューラル要約モデルの既存の学習手法である MLE と MRT について説明する．ここで，原文書を $\mathbf{x} = \langle x_1, ..., x_N \rangle$，要約を $\mathbf{y} = \langle y_1, ..., y_M \rangle$ とする．

4.3.1 Maximum Log-likelihood Estimation

MLE は学習データ D における対数尤度を最大化する:

$$L_{MLE}(\theta) = \sum_{(\mathbf{x},\mathbf{y}) \in D} \log p_\theta(\mathbf{y}|\mathbf{x}), \tag{12}$$

ここで $p_\theta(\mathbf{y}|\mathbf{x}) = \prod_{t=1}^{M} p(y_t|\mathbf{y}_{<t}, \mathbf{x})$ とする．デコード時の各ステップ t において，モデルは参照要約の位置 t における単語 y_t の生成確率を計算する．y_t は次の時刻への入力として利用される．ここで MLE での学習時においては，参照要約の長さが要約長制約であり，参照要約の単語を入力として参照要約の単語を出力とするため，要約長制約を超えた要約は生成されない．そのため，要約長制約を超えた要約の生成確率を低下させる方法は自明ではない．

4.3.2 Minimum Risk Training

MRT は MLE と異なり，生成時と同様に，直前に生成した単語を現在の時刻の入力として単語の生成確率を計算する．MRT は学習データ D に対して期待値の最小化によって，モデルを与えられた評価尺度に最適化する．

$$L_{MRT}(\theta) = \sum_{(\mathbf{x},\mathbf{y})\in D} \sum_{\mathbf{y}'\in\tilde{S}(\mathbf{x})} Q_\theta(\mathbf{y}'|\mathbf{x})\Delta(\mathbf{y},\mathbf{y}'), \tag{13}$$

ただし $Q_\theta(\mathbf{y}'|\mathbf{x}) \propto p_\theta(\mathbf{y}'|\mathbf{x})^\gamma$ であり，$\Delta(\mathbf{y},\mathbf{y}')$ は負の ROUGE 値とする．$\mathbf{y}$，$\mathbf{y}'$ は参照要約と自動生成された要約，γ は平滑化係数を表す．$\tilde{S}(\mathbf{x}) = S(\mathbf{x}) \cup \{\mathbf{y}\}$ とする (Shen, Cheng, He, He, Wu, Sun, and Liu 2016)．$S(\mathbf{x})$ は原文書 $\mathbf{x}$ に対してモデルが生成した要約の集合である．参照要約を $S(\mathbf{x})$ に加えるのは，参照要約の生成確率を向上させるためである．これは 4.5 節にて議論する．

式 (13) より，要約を生成する確率はその要約の ROUGE 値によって重みが付けられている．MRT は ROUGE 値に対してモデルを最適化するため，要約の長さは考慮されない．例えば，ROUGE の再現率に基づいてモデルを学習すれば長い要約が生成されやすくなる．ROUGE の F 値に基づいてモデルを学習すれば長すぎず，短すぎない長さの要約を生成しうるが，要約長制約は考慮できない．

そのため，要約長制御モデルは MRT によって最適化することで要約長の制御を学習することが難しい．なぜなら，これらのモデルは残り長さが 0 に近づくと要約の生成を終了することを想定しており，MLE では参照要約の単語を生成し，残り長さが 0 となるときに<EOS>を生成するように学習しているためである．

4.4 Global Optimization under Length Constraint

従来手法と異なり，提案手法 (GOLC) は要約長制約があることを想定した要約モデルの学習方法である．要約長制約を考慮するために，提案手法は MRT で用いられる従来の Δ を，要約長制約を超えた要約に罰則を与える $\tilde{\Delta}$ を導入する．提案手法は損失関数を次の様に定式化する．

$$L_{GOLC}(\theta) = \sum_{(\mathbf{x},\mathbf{y})\in D} \sum_{\mathbf{y}'\in\tilde{S}(\mathbf{x})} Q_\theta(\mathbf{y}'|\mathbf{x})\tilde{\Delta}(\mathbf{y},\mathbf{y}'), \tag{14}$$

ただし，$Q_\theta(\mathbf{y}'|\mathbf{x}) \propto p_\theta(\mathbf{y}'|\mathbf{x})^\gamma$ とする．$\tilde{\Delta}(\mathbf{y},\mathbf{y}')$ は次の様に計算する．

$$\tilde{\Delta}(\mathbf{y},\mathbf{y}') = -\texttt{ROUGE}(\mathbf{y},\texttt{trim}(\mathbf{y}',c_*(\mathbf{y}))) + \max(0,c_*(\mathbf{y}') - c_*(\mathbf{y})), \tag{15}$$

ただし ROUGE は二つの要約の間の ROUGE 値を計算する関数とする．$\texttt{trim}(\mathbf{y}', c_*(\mathbf{y}))$ は要約長制約である，参照要約の長さ $c_*(\mathbf{y})$ のなかで最長の単語列を先頭から抽出する関数とする．要約長制約が文字数の場合は，英語であれば $c_b(\mathbf{y}) = \texttt{len}(\texttt{' '.join}(\mathbf{y}))$，日本語であれば $c_b(\mathbf{y}) = \texttt{len}(\texttt{''.join}(\mathbf{y}))$ とする．これらの違いは単語の間の空白を要約長制約に含めるかどうかである．要約長制約が単語数の場合は $c_w(\mathbf{y}) = |\mathbf{y}|$ とする．

式 (15) の初項は自動生成された要約が要約長制約内において参照要約の単語を被覆しているほど小さな値をとる．自動生成された要約の要約長制約を超えた部分については ROUGE 値計算の対象とならない．第二項は自動生成された要約が要約長制約を超えているほど大きな値をとる．図 1 に Ayana et al. (2017) で用いられている $\Delta(\mathbf{y}, \mathbf{y}')$ と提案手法の違いの例を示す．

4.5 提案手法の損失関数の分析

この節では，要約長制約がある状況において生成型ニューラル要約モデルを学習する際は，既存の学習手法よりも提案手法が向いていることを議論する．さらに，自動生成された要約の集合と同様に参照要約を利用することの貢献を分析する．

参照要約 $\mathbf{y}$ に対して，$\Delta(\mathbf{y}, \mathbf{y}) = -1$ であることから，式 (13) の $L_{MRT}(\theta)$ は次の様に書き換えられる．

$$L_{MRT}(\theta) = \sum_{(\mathbf{x},\mathbf{y})\in D} \Big\{ -Q_\theta(\mathbf{y}|\mathbf{x}) + \sum_{\mathbf{y}'\in S(\mathbf{x})} Q_\theta(\mathbf{y}'|\mathbf{x})\Delta(\mathbf{y}, \mathbf{y}') \Big\}, \tag{16}$$

この式より，Δ に負の ROUGE 値を用いる場合，$\Delta(\mathbf{y}, \mathbf{y}) = -1$ より ROUGE 値は最大値を取るとともに，参照要約は人手で作成されたものであることを想定しているため，生成確率が大きく向上する．また参照要約は人手で作成したものであることを想定しているため，可読性も向上することが期待される．しかしながら，長い要約ほど高い ROUGE の再現率を取りうるため，要約長制約を超えた要約の生成確率は向上する恐れがある．

対照的に式 (14) の $L_{GOLC}(\theta)$ は要約長制約を考慮するため

$$\begin{aligned} L_{GOLC}(\theta) \quad &= \textstyle\sum_{(\mathbf{x},\mathbf{y})\in D} \big\{ -Q_\theta(\mathbf{y}|\mathbf{x}) \\ &\quad - \textstyle\sum_{\mathbf{y}^-\in S^-(\mathbf{x})} Q_\theta(\mathbf{y}^-|\mathbf{x}) \left|\tilde{\Delta}(\mathbf{y}, \mathbf{y}^-)\right| \\ &\quad + \textstyle\sum_{\mathbf{y}^+\in S^+(\mathbf{x})} Q_\theta(\mathbf{y}^+|\mathbf{x})\tilde{\Delta}(\mathbf{y}, \mathbf{y}^+) \big\}, \end{aligned} \tag{17}$$

ただし $S^-(\mathbf{x}) = \{\mathbf{y}'|\mathbf{y}' \in S(\mathbf{x}) \ \wedge \tilde{\Delta}(\mathbf{y}, \mathbf{y}') < 0\}$，$S^+(\mathbf{x}) = \{\mathbf{y}'|\mathbf{y}' \in S(\mathbf{x}) \ \wedge \tilde{\Delta}(\mathbf{y}, \mathbf{y}') \geq 0\}$ とする．ここで右辺の第二項は絶対値 $|\tilde{\Delta}(\mathbf{y}, \mathbf{y}^-)|$ が使われていることに注意してほしい．定義

により，任意の $\mathbf{y}'$ に対して $Q_\theta(\mathbf{y}'|\mathbf{x}) \geq 0$ であることと，要約長制約を超えた要約に対しては $\tilde{\Delta}(\mathbf{y}, \mathbf{y}') \geq 0$ が真であることから，$L_{GOLC}(\theta)$ を最小化することによって，$\tilde{\Delta}(\mathbf{y}, \mathbf{y}^-) < 0$ より $Q_\theta(\mathbf{y}^-|\mathbf{x})$ は増加する．また，$\tilde{\Delta}(\mathbf{y}, \mathbf{y}^+) > 0$ より，要約長制約を超えた要約に対する $Q_\theta(\mathbf{y}^+|\mathbf{x})$ は減少する．結果として提案手法によって要約長制御モデルを学習することで，要約長制約を超えた要約を生成する確率は減少する．また，参照要約の生成確率は $\tilde{\Delta}(\mathbf{y}, \mathbf{y}) = -1$ より大きく向上する．

4.6　実験

4.6.1　データセット

CNN/Daily Mail

CNN/Daily Mail コーパスから See et al. (2017) に従って，固有表現が秘匿化されていない単一文書要約データを構築した．各事例は原文書と複数文からなる要約がペアである．学習事例数は 287,226，開発事例数は 13,368，評価事例数は 11,490 となった．

Mainichi

Mainichi は毎日新聞社より購入した 2012 年から 2017 年までの日本語の単一文書要約データである．各文書に対して，17 字，54 字を要約長制約とする 2 種類の要約が存在する．要約の多くは原文書の単語を利用して作成されているが，不要な部分の削除，言い換えなども含まれる．原文書と要約の単語分割には日本語形態素解析器 MeCab[1]を用いた．学習時に一部の長いテキストによってメモリ消費の増加するのを抑えるため，原文書は先頭の 200 単語用いる．本論文では 2012 年から 2016 年までの全事例と 2017 年の一部の事例を学習データとした．2017 年の残りの事例を評価データとした．2017 年の事例はランダムサンプリングによって学習データと評価データに分けた．学習事例数は 163,220，評価事例数は 2,000 となった．学習事例，評価事例ともに半分は 17 字，残りは 54 字を要約長制約とする要約である．

図 2 は CNN/Daily Mail および Mainichi における参照要約の長さの分布である．横軸が参照要約の文字数で，縦軸が参照要約の数を表す．CNN/Daily Mail と比較して，Mainichi の参照要約の長さは分散が小さい．

Hyperparameter	Data	PG	LC
batch size of MLE	C	16	8
	M	30	8
batch size of MRT,GOLC	C, M	5	5
word embedding size	C, M	128	128
hidden state size	C, M	256	256
number of hidden layers	C, M	1	4
sample size of $\tilde{S}$	C, M	10	10
smoothing factor γ	C, M	5e-3	5e-3
gradient clip	C, M	2.0	0.1
dropput	C, M	0	0.2

Table 1: CNN/Daily Mail (C) および Mainichi (M) における実験で設定したハイパーパラメータ

4.6.2 比較するモデル

要約モデルには要約長制御をしない一般的な生成型ニューラル要約モデルと要約長制御モデルを比較した．

PG は要約長制約を制御しない pointer-generator である．このモデルは要約長制御能力はないため，Mainichi においては短い要約と長い要約でそれぞれモデルを学習した．

PG w/ LE は要約長を制御する LSTM に基づく要約モデルである．pointer-generator のデコーダ部を LenEmb とした．要約長制約までの残り長さに対する埋め込みの次元を 100 とし，長さを 0 から 400 までとした．要約長制約までの残り長さが 400 を超える場合は，実際の残り長さが 400 以下になるまで 400 を与えた．

LC は要約長を制御する畳み込みニューラルネットワークに基づく要約モデルである (Liu et al. 2018).

4.6.3 比較する学習方法

MLE は式 (12) に基づいて対数尤度を最大化する学習方法である．

MRT は式 (13) に基づいて ROUGE 値に対して要約モデルを最適化する方法である．

GOLC は式 (14) に基づく提案手法である．

[1]https://github.com/taku910/mecab

pointer-generator (PG)

Model (training method)	R-1 F	R-2 F	R-L F	Var_b	$\%over_b$	avg. time	avg. len (b)
PG (MLE)	37.74	15.78	34.35	19.35	58.35	15.25	55.70
PG w/ LE (MLE)	37.45	15.31	34.28	**4.50**	19.11	12.83	46.93
PG w/ LE (MRT)	**38.47**	**16.30**	**35.30**	18.74	43.32	24.13	74.21
PG w/ LE (GOLC)	38.27	16.22	34.99	5.13	**6.70**	**10.31**	45.77

length control CNN (LC)

Model (training method)	R-1 F	R-2 F	R-L F	Var_w	$\%over_w$	avg. time	avg. len (w)
LC (MLE)	30.67	11.00	**28.97**	**0.17**	44.67	16.93	17.72
LC (MRT)	**31.02**	**11.29**	28.54	0.21	61.67	17.19	17.97
LC (GOLC)	29.38	10.38	27.18	0.22	**21.55**	**16.41**	16.95

Table 2: CNN/Daily Mail における実験結果．太字はその列で最良の値を示す．要約長制約は参照要約の長さとした．PG w/ LE は文字数，LC は単語数を要約長制約とするため，各要約モデルにおいて学習方法を変えた場合で比較した．

MRT および GOLC を要約モデルに適用する際には，事前に MLE で学習したモデルを初期値とした．

PG，PG w/ LE のパラメータ更新には Adam (Kingma and Ba 2015) ($\alpha = 0.0001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$ を用いた．LC のパラメータ更新には Nesterov ’s Accelerated Gradient (Bengio, Boulanger-Lewandowski, and Pascanu 2013) を用いた．他のハイパーパラメータは表 1 に示す．LC における単語埋め込みの次元，隠れ状態の次元，畳み込み層の数は Liu et al. (2018) の設定の半分とした．これは MRT や GOLC を適用する際に GPU メモリ不足によるエラーを避けるためである．

4.6.4　評価指標

ROUGE CNN/Daily Mail では ROUGE F 値を用いた．F 値を計算する際は，自動生成された要約の要約長制約を超えた個所の除去などはしていない．Mainichi では要約長制約があるため，ROUGE の再現率を用いた．自動生成された要約は要約長制約を超えた個所を除いて再現率を計算した．

CNN/Daily Mail では See et al. (2017) に従って `pyrouge`[2]を用いた．Mainichi では `sumeval`[3]を用いた．

[2]https://github.com/andersjo/pyrouge
[3]https://github.com/chakki-works/sumeval

pointer-generator (PG)

Model (training method)	R-1 R	R-2 R	R-L R	Var_b	$\%over_b$	avg. time	avg. len (b)
PG (MLE)	55.77	39.35	51.36	0.035	16.18	**6.35**	31.94
PG w/ LE (MLE)	58.08	41.05	53.44	0.024	8.45	7.01	30.23
PG w/ LE (MRT)	**60.89**	**42.62**	**55.80**	1.419	20.90	11.75	44.99
PG w/ LE (GOLC)	59.15	41.62	54.20	**0.018**	**7.35**	6.69	33.41

length control CNN (LC)

Model (training method)	R-1 R	R-2 R	R-L R	Var_w	$\%over_w$	avg. time	avg. len (w)
LC (MLE)	47.53	30.5	43.39	**0.0049**	7.4	8.82	18.46
LC (MRT)	**49.72**	**32.08**	**45.63**	0.031	5.9	8.53	19.30
LC (GOLC)	44.22	27.97	40.77	0.027	**0.25**	**7.90**	17.05

Table 3: Mainichi における実験結果．ROUGE 値計算の際は要約長制約を超えた部分は計算の対象から除去した．PG w/ LE では短い要約では 17 字，長い要約では 54 字を要約長制約とした．LC では参照要約の単語数を要約長制約とした．

要約長制御 要約長制御の評価には二つの尺度を用いた．一つ目は要約長制約と自動生成された要約の長さ l_i の差の分散に基づく尺度である (Liu et al. 2018).

$$Var_* = 0.001 * \frac{1}{n}\sum_{i=0}^{n} |l_i - c_*(\mathbf{y})|^2. \tag{18}$$

もう一つは要約長制約を超えた割合を表す*%over* である．評価事例に対して，自動生成された要約が要約長制約を超えた事例数を，評価事例数で割った値とした．PG w/ LE と LC では要約長の単位が異なるため，要約モデル間の比較はできない．GOLC は学習方法であるため，各要約モデルを既存の学習方法で学習させた場合と比較する．

生成にかかる平均時間 (avg. time) CPU における評価事例 1 件当たりの生成時間を計測する．

人手評価 人手評価により可読性と情報性を評価した．

- 可読性 (Read.): 要約は文法的に正しい
- 情報性 (Info.): 要約は要約長制約内で重要な情報を含んでいる

また，自動生成された要約を人手で後編集した際にかかる時間も評価した．

4.6.5 ROUGE および要約長制御能力の評価

表 2 に CNN/Daily Mail における ROUGE F 値，*Var* および*%over* を示す．GOLC によって学習された PG w/ LE は MLE で学習するよりも高い ROUGE 値を示しつつ，*%over* が削減でき

Over or Not \ 要約長制約	17 chars.	54 chars.
Overlength	21.3 sec.	78.6 sec.
In length	12.90 sec.	55.7 sec.

Table 4: Mainichi における人手の後編集時間．各時間は後編集時間 (秒) の平均を示す．

ている．MRT によって学習された PG w/ LE は GOLC よりも高い ROUGE 値となるが，*%over* は GOLC よりも高い値となっている．これらの結果から，GOLC は ROUGE 値を保ちつつ，要約長制約内で多くの要約を生成できていることが分かる．LC の ROUGE 値は PG および PG w/ LE よりも低い値となった．これは LC が原文書の単語をコピーできないことが理由の一つとして挙げられる．LC を GOLC によって学習することで*%over* の削減はできているものの，ROUGE 値は MLE や MRT よりも低下した．これは GOLC で学習することで，他の手法よりも短い要約を生成する傾向になったためである．PG w/ LE は残り長さを直接モデル化しているのに対して，LC は要約長制約と現在の要約長の組み合わせで要約長を制御しているため，要約長制約付近で生成を終了するよう学習することが難しいことが一因である．

表 3 に Mainichi における ROUGE 再現率，*Var* および*%over* を示す．PG および PG w/ LE は短い要約では 17 字，長い要約では 54 字を要約長制約として自動要約結果を要約長制約内で評価した．PG w/ LE は PG よりも高い ROUGE 値を示した．これは PG w/ LE が PG の学習事例数を 2 倍にできたことが一因である．また GOLC で学習した PG w/ LE は既存の学習方法よりも*%over* を削減しつつ，MLE よりも ROUGE 値が高くなった．この結果から，要約長制約が存在する実際の要約データにおいても，提案手法は有効であるといえる．

表 2 と表 3 から GOLC によって学習した要約モデルの生成時間は既存の学習方法よりも速くなった．MRT によって学習したモデルの ROUGE 値は他の手法よりも高い値となったが，生成時間は他の手法よりも遅いことが分かる．これは MRT によって学習したモデルが他の手法よりも長い要約を生成していることが原因である．表 2 および表 3 において GOLC によって LC を学習することで*%over* は改善したものの ROUGE 値は低下した．GOLC で学習した LC は他の手法と比較して要約長制約よりも短い要約を生成することがあった．そのため再現率が低下し，結果として F 値も他の手法よりも低下した．PG w/ LE は要約長制約までの残り長さを直接モデル化した手法であるが，LC は要約長制約と現在までに生成した要約の長さの組み合わせによって要約長を制御するため，要約長制約の近くで，かつ超えないように要約を生成することが難しかったことが一因である．

Sum.	17 chars. Read.	17 chars. Info.	54 chars. Read.	54 chars. Info.
Inlen (no-edit)	2.8	2.4	3.4	2.8
Over (no-edit)	2.6	3.2	3.6	3.8
Inlen (edit)	4.2	4.2	4.0	4.2
Over (edit)	3.8	4.4	4.8	4.6

Table 5: 自動生成された要約 (non-edit) および後編集結果 (edit) に対する情報性と可読性の主観評価

4.6.6 毎日新聞コーパスにおける人手の後編集評価

要約長制約内に要約を生成することの重要性を評価するために，自動生成された要約の後編集にかかる時間を計測した．この実験では 7 名の日本語母語話者に評価を依頼した．

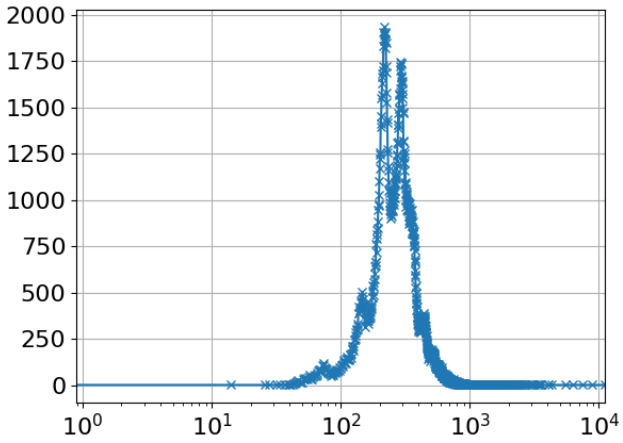
PG，PG w/ LE (MLE)，PG w/ LE (MRT)，PG w/ LE (GOLC) によって生成された要約から無作為に要約長制約を超えたものと，超えていないものを 10 ずつ抽出した．要約長制約を超えないことの重要性を評価するため，手法毎の比較はおこなっていない．

表 4 に後編集にかかった時間の平均値を示す．実験結果より，要約長制約を超えた要約の方が編集にかかる時間が長くなった．要約長制約内に要約を生成することで，17 字に対しては 39.4%，54 字に対しては 29.1%の編集時間短縮となった．この結果から，要約長制約内に要約を生成することは人手の後編集支援に有効であることが期待できる．表 3 と表 4 から提案手法によって要約モデルを学習することで，MRT で学習した場合と比較して約 10%の後編集時間削減に貢献する．

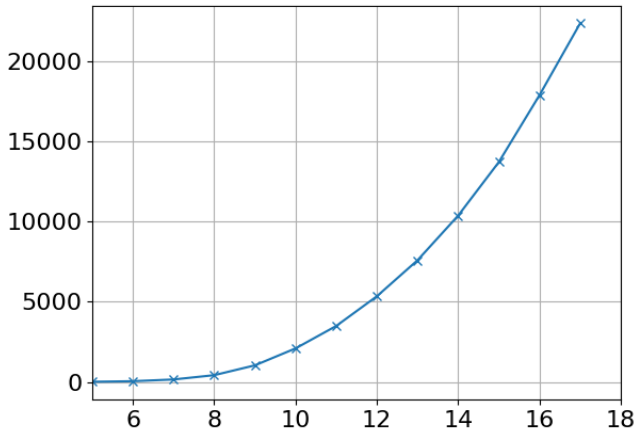
表 4 に自動生成された要約 (non-edit) および後編集結果 (edit) に対するの可読性と情報性の評価結果を示す．実験結果より，後編集結果の評価値は自動生成された要約よりも高くなっている．このことから，後編集では妥当な編集がおこなわれているといえる．

4.7 要約長制約下における生成型ニューラル要約モデルの学習方法のまとめ

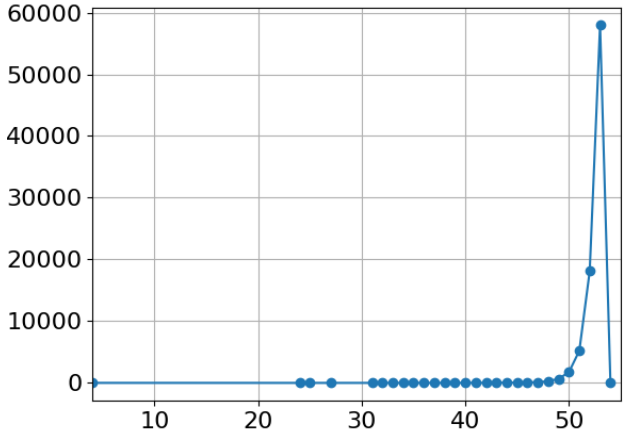
本論文では要約長制約下における生成型ニューラル要約モデルの学習方法を提案した．提案手法は要約長制約を超えた要約の生成を削減しつつ，MLE の ROUGE 値よりも向上することを確認した．要約長制約内で要約を生成することにより，人手による後編集時間が短縮されることを確認した．



(a) CNN/Daily Mail



(b) Mainichi (short)



(c) Mainichi (long)

Figure 2: CNN/Daily Mail および毎日新聞コーパスにおける参照要約の文字数の分布．横軸が参照要約の字数で縦軸が参照要約の件数を表す．

第5章　原文書と要約対の差分に基づく生成用語彙構築手法

5.1　研究概要

文書要約手法は抽出型要約手法と生成型要約手法に分けられる．抽出型要約手法は原文書に含まれる単語や文を選択して要約を作成する手法であるため，原文書の情報を言い換える必要がない文書要約問題に向いているといえる．生成型要約手法は生成によって要約を作成する手法であるため，原文書の情報を言い換える必要がある文書要約問題に向いているといえる．

生成型要約の中でも，コピー機構を伴う生成型ニューラル要約モデルが高い精度を示すことがよく知られている (Gulcehre et al. 2016; Nallapati, Zhou, dos Santos, Gùlchere, and Xiang 2016; See et al. 2017)．その特徴は，学習データから事前に構築した単語集合 (生成用語彙) からの単語生成に加え，原文書からの単語をコピーして要約に用いる点にある．一般的な生成型ニューラル要約モデルでは，生成用語彙をある程度のサイズに収めるため，学習データ中の参照要約に頻出する単語で構成する．しかし，人名や地名のような固有表現は頻度が低いため，生成 用語彙には含まれないことが多く，要約生成にこれらの語が利用できないという問題が生じる．これに対し，コピー機構を伴う生成型ニューラル要約モデルは，原文書の単語をコピーして要約に利用できるためこの問題を回避できる．特に新聞記事には固有表現がよく出現するため，その有効性は顕著であると考えられる．

一方，近年では大量のラベルなしコーパスを用いた事前学習を適用することで，高い精度が報告されている (Lewis et al. 2020; Raffel et al. 2019; Yan, Qi, Gong, Liu, Duan, Chen, Zhang, and Zhou 2020; Zhang, Zhao, Saleh, and Liu 2019)．事前学習はそれ単体でも有効であるが，コピー機構を 伴う生成型ニューラル要約モデルに適用すると精度がさらに改善されることが報告されており (Huang et al. 2020; Wang, Zhao, Jia, Li, and Liu 2019b)， コピー機構は自動要約において現在でも重要な役割を担っている．

本論文では，コピー機構を伴う生成型ニューラル要約モデルに対して，限られた語彙サイズでも要約特有な表現を生成できるような語彙構築方法を提案する．提案手法は，学習データ中の参照要約のみを利用するのではなく原文書と参照要約の対を比較し， 参照要約のみに

出現する語を生成用語彙とする．そのため，提案法は従来法よりも小さな語彙サイズで多様な単語を生成することが期待できる．

英語の単一文書要約のデータセットである CNN/Daily Mail および NEWSROOM において，コピー機構を伴うニューラル要約のうち高い精度を示している pointer-generator を要約モデルとして用い，提案手法の有効性を評価する．その結果，提案手法は既存の語彙構築方法よりも高い ROUGE 値を示しつつ，要約特有の表現を生成する割合が増加することを示す．

5.2　生成型ニューラル要約モデルで用いられる語彙

本節では生成型ニューラル要約モデルにおける語彙について説明する．生成型ニューラル要約モデルで用いられる語彙は 3 種類あり，役割は次の通りである．

原文書側の語彙 (Source Vocabulary; V_x): 原文書の単語を埋め込みに変換する際に利用する．要約モデルが原文書をエンコードする際に，単語に対して V_x の中で一意の ID を割り当てる．V_x にない単語は未知語として扱われる．

要約側の語彙 (Target Vocabulary; V_y): 要約の単語を埋め込みに変換する際に利用する．要約モデルが要約を作成する際に，単語に対して V_y の中における一意の ID を割り当てる．V_y にない単語は未知語として扱われる．

生成用語彙 (Output Vocabulary; V_o): 要約モデルが生成候補とする単語を保持する．生成確率が最大となる ID を文字列に変換して要約として生成する．

ここで，ある語彙 V における単語 $\bar{x}$ の ID x への変換は $x = \mathrm{id}_V(\bar{x})$，ID x から単語 $\bar{x}$ への変換は $\bar{x} = \mathrm{word}_V(x)$ で得られるものとする．

5.3　従来の語彙構築手法

本章では生成型ニューラル要約モデルで用いられる従来の語彙構築方法を述べる．

一般的に，生成型ニューラル要約モデルでは，学習データに出現するすべての単語を語彙として利用するのではなく，語彙に含める単語を何らかの方法で限定することが多い．その理由は，計算資源や学習時間，要約生成時間とに起因する．語彙が大きいほど生成可能な単語種は多くなるが，一方でより多くのメモリを消費し，また処理速度の低下の原因となりうる (Chen, Grangier, and Auli 2016)．そのため，すべての単語を含んだ語彙を使って要約モデルを学習し，生成に利用するのは計算資源の観点で難しい場合があり，速度の低下から実用

Algorithm 1 Target Output shared vocabulary construction (TO) (左) と **Source Target Output shared vocabulary construction (STO)** (中) : 学習データ $\bar{D}$ の原文書と参照要約を対象に単語の頻度を集計し，高頻度語上位の単語集合を語彙として出力する．f_x，f_y，f は単語の頻度を保持するハッシュであり，頻度の初期値は 0 とする．`GetHighFreqWords` は指定された件数 (N_x, N_y, N, N_o) の高頻度語を返す関数とする．

Require: $\bar{D}, N_x, N_y$ ### TO
  $f_x = \{\}, f_y = \{\}$ # 単語を key，頻度を value とするハッシュ
  for $(\bar{\mathbf{x}}, \bar{\mathbf{y}}) \in \bar{D}$ **do**
    for $i = 1$ to $|\bar{\mathbf{x}}|$ **do**
      $f_x[\bar{x}_i] = f_x[\bar{x}_i] + 1$
    for $i = 1$ to $|\bar{\mathbf{y}}|$ **do**
      $f_y[\bar{y}_i] = f_y[\bar{y}_i] + 1$
  $V_x = \texttt{GetHighFreqWords}(f_x, N_x)$
  $V_y = \texttt{GetHighFreqWords}(f_y, N_y)$
  $V_o = V_y$
  return V_x, V_y, V_o

Require: $\bar{D}, N$ ### STO
  $f = \{\}$ # 単語を key，頻度を value とするハッシュ
  for $(\bar{\mathbf{x}}, \bar{\mathbf{y}}) \in \bar{D}$ **do**
    for $i = 1$ to $|\bar{\mathbf{x}}|$ **do**
      $f[\bar{x}_i] = f[\bar{x}_i] + 1$
    for $i = 1$ to $|\bar{\mathbf{y}}|$ **do**
      $f[\bar{y}_i] = f[\bar{y}_i] + 1$
  $V = \texttt{GetHighFreqWords}(f, N)$
  $V_x = V_y = V_o = V$
  return V_x, V_y, V_o

面で問題になる場合もある．そこで，語彙サイズは，計算機のメモリや，生成型ニューラル要約モデルの処理速度や精度などのバランスによって決められることが多い．

語彙に含める単語の選択基準として，多くの研究では，学習データにおける出現頻度を集計し，指定された件数の高頻度語集合を語彙として構築している．

Algorithm 1 および Algrithm 2 に従来の三つの語彙構築方法の疑似コードを示す．それぞれの特徴は次の通りである．

- **Target Output shared vocabulary construction (TO)**: 原文書側の語彙 V_x と要約側の語彙 V_y は独立に作成し，V_o は V_y と共有する．
- **Source Target Output shared vocabulary construction (STO)**: V_x と V_y と V_o ですべて共有する (See et al. 2017)．機械翻訳と異なり原文書の単語が生成されることもあるため文書要約に特化した構築方法である．
- **Source Target shared vocabulary construction (ST)**: V_x と V_y を共有し，V_o は V_x および V_y とは別に保持する．V_o は TO と同様に参照要約に出現する頻度に基づいて構築する (McCann, Keskar, Xiong, and Socher 2018).

Algorithm 2 Source Target shared vocabulary construction (ST) : 学習データ $\bar{D}$ の原文書と参照要約を対象に単語の頻度を集計し，高頻度語上位の単語集合を語彙として出力する．f_x，f_y，f は単語の頻度を保持するハッシュであり，頻度の初期値は 0 とする．`GetHighFreqWords` は指定された件数 (N_x, N_y, N, N_o) の高頻度語を返す関数とする．

Require: $\bar{D}, N, N_o$ ### ST
  $f = \{\}; f_o = \{\}$ # 単語を key，頻度を value とするハッシュ
  for $(\bar{\mathbf{x}}, \bar{\mathbf{y}}) \in \bar{D}$ **do**
    for $i = 1$ to $|\mathbf{x}|$ **do**
      $f[\bar{x}_i] = f[\bar{x}_i] + 1$
    for $i = 1$ to $|\bar{\mathbf{y}}|$ **do**
      $f[\bar{y}_i] = f[\bar{y}_i] + 1$
      $f_o[\bar{y}_i] = f_o[\bar{y}_i] + 1$
  $V = \texttt{GetHighFreqWords}(f, N)$
  $V_o = \texttt{GetHighFreqWords}(f_o, N_o)$
  $V_x = V_y = V$
  return V_x, V_y, V_o

各手法は以降，簡略化のため Target Output shared vocabulary construction であれば TO のように省略表記で呼ぶ．省略表記は，共有される語彙の頭文字を表す．たとえば TO であれば参照要約側の語彙 (Target Vocabulary) と生成用語彙 (Output Vocabulary) が共有されていることを表す．これらの語彙構築方法によって原文書の単語列 $\bar{\mathbf{x}} = \langle \bar{x}_1, \bar{x}_2, ..., \bar{x}_M \rangle$ と参照要約の単語列 $\bar{\mathbf{y}} = \langle \bar{y}_1, \bar{y}_2, ..., \bar{y}_N \rangle$ のペアからなる学習データ $\bar{D}$ の原文書と参照要約をそれぞれ独立に用いて高頻度語を抽出することで原文書側の語彙 V_x，要約側の語彙 V_y，生成用語彙 V_o を構築する．

5.4 Pointer-Generator

本章では，コピー機構を伴う生成型ニューラル要約モデルとして用いる pointer-generator (See et al. 2017) について述べる．pointer-generator は原文書をベクトル表現へエンコードする双方向 LSTM encoder とエンコード結果から要約を生成する attention 機構付きの LSTM decoder からなる．pointer-generator の学習の際には構築された語彙に基づいて一意に割り当てられる ID に変換された原文書 $\mathbf{x} = \langle x_1, ..., x_M \rangle$ $(x_i = \mathrm{id}_{V_x}(\bar{x}_i))$，参照要約 $\mathbf{y} = \langle y_1, ..., y_N \rangle$ $(y_t = \mathrm{id}_{V_y}(\bar{y}_t))$ のペアからなる学習データ D を用いる．生成の際は原文書 $\mathbf{x}$ を入力として要約を作成する．語彙に含まれない単語は未知語を表すトークン `UNK` に対応する ID に変換されるものとする．

双方向 LSTM encoder は先頭の単語から順に対応する隠れ状態を計算する前向き LSTM と末

尾の単語から順に対応する隠れ状態を計算する後ろ向き LSTM からなる．たとえば，前向き LSTM encoder ($\overrightarrow{f}$) は位置 i における隠れ状態 $\overrightarrow{\mathbf{h}}_i$ を計算する際には直前の隠れ状態 $\overrightarrow{\mathbf{h}}_{i-1}$ および現在の単語の埋め込み $\mathbf{e}_{x_i}^{(x)} \in \mathbb{R}^d$ を受け取り，隠れ状態を出力する: $\overrightarrow{\mathbf{h}}_i = \overrightarrow{f}(\overrightarrow{\mathbf{h}}_{i-1}, \mathbf{e}_{x_i}^{(x)})$ [1].

ただし $\mathbf{e}_{x_i}^{(x)}$ はパラメータ $\mathbf{E}^{(x)} \in \mathbb{R}^{|V_x| \times d}$ において原文書の単語 ID x_i に対応する列ベクトルとする．d は単語埋め込みの次元を表す．$\overrightarrow{\mathbf{h}}_i \in \mathbb{R}^h$ $(1 \leq i \leq M)$ であり，h は隠れ状態の次元を表す．後ろ向き LSTM encoder ($\overleftarrow{f}$) の隠れ状態 $\overleftarrow{\mathbf{h}}_i$ は $\overleftarrow{\mathbf{h}}_{i+1}$ および $\mathbf{e}_{x_i}^{(x)}$ を用いて同様に計算できる．$\overrightarrow{\mathbf{h}}_0$ および $\overleftarrow{\mathbf{h}}_{M+1}$ はゼロベクトルとした．位置 i における双方向 LSTM encoder の出力は $\mathbf{h}_i = \overrightarrow{\mathbf{h}}_i + \overleftarrow{\mathbf{h}}_i$ とする．LSTM decoder (f) も同様に直前の隠れ状態 $\mathbf{s}_{t-1}$ に基づいて現在の隠れ状態を計算する．

$$\mathbf{s}_t = f(\mathbf{s}_{t-1}, [\mathbf{e}_{y_t}^{(y)}; \mathbf{c}_{t-1}]) \tag{19}$$

ただし $\mathbf{e}_{y_t}^{(y)} \in \mathbb{R}^d$ はパラメータ $\mathbf{E}^{(y)} \in \mathbb{R}^{|V_y| \times d}$ における生成候補の単語 ID y_t に対応する列ベクトル，$\mathbf{s}_t \in \mathbb{R}^h (1 \leq t \leq N)$ である．$\mathbf{c}_{t-1} \in \mathbb{R}^h$ は直前の時刻における原文書の文脈ベクトル $\mathbf{h}_i$ の加重平均であり，$t = 1$ のときはゼロベクトルとする．$[\cdot;\cdot]$ を二つのベクトルの結合とする．現在の時刻における単語に加えて，直前の時刻の原文書の単語ベクトルの加重平均を考慮することで，原文書の単語と要約の単語のアラインメントを考慮する (Luong et al. 2015)．また，$\mathbf{s}_0$ は $\overleftarrow{\mathbf{h}}_0$ とした．

次に $\mathbf{s}_t$ および $\mathbf{h}_i$ を入力とする多層パーセプトロンの出力に基づいて，attention 確率 $a_{t,i}$ を計算する．

$$a'_{t,i} = \mathbf{w}_a^\top \tanh(\mathbf{W}_{as}\mathbf{s}_t + \mathbf{W}_{ah}\mathbf{h}_i + \mathbf{w}_{ac}\sum_{t'=1}^{t-1} a_{t',i}) \tag{20}$$

$$a_{t,i} = \frac{\exp(a'_{t,i})}{\sum_{i'=1}^{M}\exp(a'_{t,i'})} \tag{21}$$

ただし $\mathbf{w}_a, \mathbf{w}_{ac} \in \mathbb{R}^h$，$\mathbf{W}_{as}, \mathbf{W}_{ah} \in \mathbb{R}^{h \times h}$ とする．得られた attention 確率に基づいて得られた重み付き平均ベクトル $\mathbf{c}_t = \sum_i a_{t,i}\mathbf{h}_i$ に基づいて語彙 V_o 中に含まれる単語 ID y_t の生成確率を計算する．

$$p_g(y_t|\mathbf{y}_{<t}, \mathbf{x}) = \frac{\exp(\mathbf{W}_{o,y_t}[\mathbf{s}_t; \mathbf{c}_t] + b_{o,y_t})}{\sum_{y' \in V_o}\exp(\mathbf{W}_{o,y'}[\mathbf{s}_t; \mathbf{c}_t] + b_{o,y'})} \tag{22}$$

[1] LSTM は学習対象となるパラメータがあるが，簡略化のため省略する．

ただし $\mathbf{W}_{o,y_t} \in \mathbb{R}^{2h}$ であり，パラメータ $\mathbf{W}_o \in \mathbb{R}^{|V_o| \times 2h}$ における y_t に対応する行ベクトルとする．$b_{o,y_t} \in \mathbb{R}$ であり，$\mathbf{b}_o \in \mathbb{R}^{|V_o|}$ における y_t に対応する．ここで語彙構築方法を TO としても STO としても $V_y = V_o$ より，単語埋め込みのパラメータ $\mathbf{E}^{(y)}$ および softmax 関数内のパラメータ $\mathbf{W}_o$，$\mathbf{b}_o$ はともに V_y に基づいてサイズが決まることに注意してほしい．最終的な単語の生成確率は語彙の単語の生成確率分布と，原文書の単語に対する attention 確率分布の線形結合で表される．

$$p(y_t|\mathbf{y}_{<t}, \mathbf{x}) = \lambda_t p_g(y_t|\mathbf{y}_{<t}, \mathbf{x}) + (1 - \lambda_t) \sum_{i:\bar{x}_i = \bar{y}_t} a_{t,i} \tag{23}$$

$\mathbf{y}_{<t} = y_1, ..., y_{t-1}$ は要約の先頭から直前の時刻までの単語 ID 列，$\sum_{i:\bar{x}_i=\bar{y}_t} a_{t,i}$ は原文書の単語 $\bar{x}_i$ のうち $\bar{y}_t$ と一致する単語に対して割り当てられた attention 確率の総和を表す．つまり原文書中に何度も出現するほど高い値となり，その単語はコピーされやすくなる．ここで，V_y に含まれる単語のうち，V_o に含まれないものに関しては初項は常に 0 となることに注意してほしい．λ_t は値が大きいほど語彙からの単語の生成を重視する．この値は次の様に文脈に基づいて計算される．

$$\lambda_t = \sigma(\mathbf{w}_c \mathbf{c}_t + \mathbf{w}_s \mathbf{s}_t + \mathbf{w}_y \mathbf{e}_{y_t}^{(y)} + b) \tag{24}$$

ただし σ は標準シグモイド関数，$\mathbf{w}_c, \mathbf{w}_s \in \mathbb{R}^h$，$\mathbf{w}_y \in \mathbb{R}^d$，$b \in \mathbb{R}$ とする．pointer-generator は以下の目的関数の最小化に基づいてパラメータ θ を学習する．

$$L(\theta) = \sum_{(\mathbf{x},\mathbf{y}) \in D} \{L_{nll}(\theta) + \kappa L_{cov}(\theta)\} \tag{25}$$

$$L_{nll}(\theta) = \sum_t -\log p(y_t|\mathbf{y}_{<t}, \mathbf{x}) \tag{26}$$

$$L_{cov}(\theta) = \sum_t \min(a_{t,i}, \sum_{t'=0}^{t-1} a_{t',i}) \tag{27}$$

$L_{nll}(\theta)$ は負の対数尤度，$L_{cov}(\theta)$ は同じ単語が繰り返し生成されることを抑制するための罰則項である．κ はハイパーパラメータである．

実際に要約を作成する際には，ビームサーチに基づいて，各時刻において式 (23) から得られる確率が高い上位の単語を出力する．式 (23) の第二項によって，原文書の単語を出力することができるようになるが，原文書の単語をある時刻において出力した場合，次の時刻における単語埋め込みは，その単語が V_y に含まれていれば，その単語に対応する埋め込み，存在しなければ未知語に対応する埋め込みが式 (19) の計算に用いられる．

Algorithm 3 Source Target shared and Difference-based Output vocabulary construction (ST-DO): 原文書 $\bar{\mathbf{x}}$ と参照要約 $\bar{\mathbf{y}}$ の対を用いた語彙構築．学習データ $\bar{D}$ の原文書と参照要約を対象に単語の頻度を集計し，高頻度語上位 N，N_o 件の単語集合を語彙として出力する．

Require: $\bar{D}, N, N_o$ ### ST-DO
  $f = \{\}; f_o = \{\}$ # 単語を key，頻度を value とするハッシュ
  for $(\bar{\mathbf{x}}, \bar{\mathbf{y}}) \in \bar{D}$ **do**
    for $i = 1$ to $|\bar{\mathbf{x}}|$ **do**
      $f[\bar{x}_i] = f[\bar{x}_i] + 1$
    for $i = 1$ to $|\mathbf{y}|$ **do**
      $f[\bar{y}_i] = f[\bar{y}_i] + 1$
      if $\bar{y}_i \notin \bar{\mathbf{x}}$ **then**
        $f_o[\bar{y}_i] = f_o[\bar{y}_i] + 1$
  $V = \texttt{GetHighFreqWords}(f, N)$
  $V_o = \texttt{GetHighFreqWords}(f_o, N_o)$
  $V_x = V_y = V$
  return V_x, V_y, V_o

5.5 提案手法

本論文では，各原文書と参照要約のペアに対して原文書には出現せずに参照要約には出現する単語を対象に生成用語彙を構築する方法 (Source Target shared and Difference-based Output vocabulary construction; ST-DO) を提案する．

5.5.1 単語埋め込みの共有と生成用語彙の構築

Figure 3 に既存の語彙構築方法と ST-DO の語彙構築方法の違いの例を示す．提案手法は V_x と V_y を共有し，V_o は原文書には出現せず，参照要約のみに出現した単語の中で高頻度となるものから構築する．既存手法は参照要約における単語の頻度に基づいて語彙を構築する．そのため，“.” や “a” といった単語も生成用語彙に含まれる．一方で，提案手法は原文書と参照要約の対を利用して，原文書には出現せず参照要約にのみ出現する単語を対象として語彙を構築する．たとえば “says” は参照要約にのみ出現するため，この単語は生成用語彙に含まれる．Algorithm 3 に疑似コードを示す．

多様な単語を生成できるようにするために，従来手法では語彙サイズを大きくする必要があったが，それと比較して提案手法ではより小さな語彙サイズで多様な単語を生成できるようになると期待できる．

Table 6: CNN/Daily Mail，NEWSROOM の統計値

	CNN/Daily Mail	NEWSROOM
学習データ		
事例数	287,227	995,041
原文書の平均単語数	791.58	771.77
参照要約の平均単語数	55.15	30.30
開発データ		
事例数	13,368	108,837
原文書の平均単語数	769.50	765.63
参照要約の平均単語数	61.41	30.66
評価データ		
事例数	11,490	108,862
原文書の平均単語数	778.50	763.75
参照要約の平均単語数	58.30	30.57

5.5.2　単語埋め込み層における単語 ID の変換

提案手法では softmax 層で出力した単語 ID と，単語埋め込み層で保持する単語 ID は基本的に一致しない．そこで softmax 層で出力した単語 ID を単語埋め込み層の対応する単語 ID へ変換する処理を追加する．

$$\mathbf{e}_{y_t}^{(y)} = \mathbf{e}_{V_{o:y_t} \to V_{y:\bar{y}_t}}^{(y)} \tag{28}$$

ただし $V_{o:y_t} \to V_{y:\bar{y}_t}$ は V_o で管理されている単語 ID y_t を V_y で管理されている単語 ID $\bar{y}_t$ へ変換する関数とする．たとえば，y_t が $1,000$ のとき，これは V_o における ID が $1,000$ であることを表す．この ID と対応する V_y における ID が $30,000$ であるとすると，単語の埋め込みを取得する際には，パラメータ $\mathbf{E}^{(y)} \in \mathbb{R}^{|V_y| \times d}$ における $1,000$ 番目の列ではなく，$30,000$ 番目の列ベクトルを取得することになる．提案手法は式 (19) と式 (24) の $\mathbf{e}_{y_t}^{(y)}$ を式 (28) で置き換えることで，V_y と V_o で管理する ID が対応をとれるようにする．y_t に対応する単語が V_o に登録されていない場合，y_t は UNK に対応する ID へ変換する．この処理は ST および ST-DO に対して適用する．

Table 7: 各モデルのパラメータ数．$d\ (= 128)$ は単語埋め込みの次元，$h\ (= 256)$ は LSTM の隠れ状態の次元を表す．$\#\mathbf{E}_x$ と $\#\mathbf{E}_y$ は単語埋め込みのパラメータ数，$\#\mathbf{W}_o$ は単語の生成確率分布を計算するための softmax 層のパラメータ数，#Total はモデルすべてのパラメータ数を表す．STO_{50K}，$\text{STO}_{SW:32K}$，ST および ST-DO は $\mathbf{E}_x = \mathbf{E}_y$ である．† がつく手法が提案手法を示す．

Method	$\#\mathbf{E}_x + \#\mathbf{E}_y$	$\#\mathbf{W}_o$	#Total
STO_{50K}	$50K \times d$	$50K \times 2h$	20.96M
$\text{STO}_{SW:32K}$	$32K \times d$	$32K \times 2h$	18.13M
TO_{1K}	$(50K + 1K) \times d$	$1K \times 2h$	8.49M
TO_{5K}	$(50K + 5K) \times d$	$5K \times 2h$	10.03M
ST_0	$50K \times d$	0	8.10M
ST_{1K}	$50K \times d$	$1K \times 2h$	8.37M
ST_{5K}	$50K \times d$	$5K \times 2h$	9.39M
$\text{ST-DO}^{\dagger}_{1K}$	$50K \times d$	$1K \times 2h$	8.37M
$\text{ST-DO}^{\dagger}_{5K}$	$50K \times d$	$5K \times 2h$	9.39M
$\text{ST-DO}^{\dagger}_{50K}$	$50K \times d$	$50K \times 2h$	20.96M

5.6 実験

5.6.1 実験設定

実験には固有表現が匿名化されていない版の CNN/Daily Mail (Hermann, Kocisky, Grefenstette, Espeholt, Kay, Suleyman, and Blunsom 2015) および NEWSROOM (Grusky et al. 2018) を用いた．Table 6 にデータサイズを示す．

未知語を削減するために，サブワード単位でテキストを分割する方法も考えられる．実験では，単語分割方法の影響を比較するため，Stanford CoreNLP (Manning, Surdeanu, Bauer, Finkel, Bethard, and McClosky 2014) と SentencePiece (Kudo and Richardson 2018) の 2 種類で実験を行った．Stanford CoreNLP は人が決めた単位に基づいてテキストを単語分割する単語分割器であり，SentencePiece は教師なしの単語分割器である．SentencePiece を含む教師なしの単語分割では，指定された語彙サイズ以内で，未知語を減らすようにサブワードと呼ばれる単位で語彙を構築する．そのため，Stanford CoreNLP と比較して，SentencePiece による分割のほうが，未知語が減る傾向にある．SentencePiece の学習には，学習データの原文書と参照要約を用いた．

Stanford CoreNLP での系列長の制限: ミニバッチで学習をする際に，一部の長いテキストによって学習時間や消費メモリの増加を避けるために，CNN/Daily Mail は学習時，生成時と

Table 8: ROUGE-F 値の平均値．Δ_{len} は自動要約結果の単語数から参照要約の単語数を引いた値の平均値を表す．PtrGen+Cov の ROUGE 値について，CNN/Daily Mail は (See et al. 2017), NEWSROOM は (Grusky et al. 2018) からそれぞれ引用した．$\text{Transformer}_{\text{BASE}}$ の ROUGE 値は (Zhang et al. 2019) から引用した．太字は本論文で実装した pointer-generator の ROUGE 値の中で最良の値を示す．

	CNN/Daily Mail				NEWSROOM			
Method	R-1 F	R-2 F	R-L F	Δ_{len}	R-1 F	R-2 F	R-L F	Δ_{len}
PtrGen+Cov	39.53	17.28	36.38	-	27.54	13.32	23.50	-
$\text{Transformer}_{\text{BASE}}$	38.27	15.03	35.48	-	40.28	27.93	36.52	-
LEAD	39.56	17.15	35.67	24.33	31.76	21.81	29.42	41.34
STO_{50K}	38.72	16.72	35.21	7.30	36.80	25.29	33.46	2.61
$\text{STO}_{sw:32K}$	38.66	**16.76**	35.18	8.91	36.83	25.44	33.43	0.80
TO_{1K}	38.42	16.36	34.86	11.70	37.17	25.67	33.83	4.91
TO_{5K}	38.46	16.54	35.07	19.90	37.21	25.79	33.9	6.20
ST_{0}	38.44	16.42	34.94	11.43	36.84	25.21	33.37	1.63
ST_{1K}	38.30	16.13	34.68	15.48	37.01	25.63	33.71	7.01
ST_{5K}	38.42	16.39	34.89	16.73	37.25	25.80	33.91	3.90
(提案手法) ST-DO_{1K}	38.39	16.02	34.91	14.82	37.37	25.86	34.02	7.54
(提案手法) ST-DO_{5K}	**38.78**	16.71	**35.23**	16.75	**37.69**	**26.18**	**34.36**	6.31
(提案手法) ST-DO_{50K}	38.68	16.65	35.12	7.52	37.19	25.73	33.84	1.62

もに原文書は先頭 400 単語を用いた．また学習時の参照要約は先頭 100 単語を生成対象とした．NEWSROOM は学習時，生成時ともに原文書は先頭 200 単語を用いた．また学習時の参照要約は先頭 100 単語を生成対象とした．

SentencePiece での系列長の制限: SentencePiece によるテキスト分割から得られる系列長は，Stanford CoreNLP によるテキスト分割で得られる系列長よりも長くなる傾向にある．そのため，学習時および生成時で利用する系列長は原文書について，先頭から最大で 512，学習時の参照要約の系列長も先頭から最大で 512 とした．

要約モデルの設定: すべての実験で pointer-generator を要約モデルとして用いた．また，実装には PyTorch (Paszke, Gross, Massa, Lerer, Bradbury, Chanan, Killeen, Lin, Gimelshein, Antiga, Desmaison, Kopf, Yang, DeVito, Raison, Tejani, Chilamkurthy, Steiner, Fang, Bai, and Chintala) を用いた．式 (25) の κ は (See et al. 2017) に従い 1.0 とした．STO に関しては，(See et al. 2017) の設定に合わせて Stanford CoreNLP (Manning et al. 2014) で単語を分割し $|V_o| = 50K$ としたモデル (**STO**$_{50K}$) と，未知語の割合を削減できる SentencePiece (Kudo and Richardson

2018) でテキストを分割し $|V_o| = 32K$ としたモデル (**STO**$_{SW:32K}$) を評価した．語彙構築方法 TO，ST に対しては Stanford CoreNLP でテキストを分割し，$|V_o|$ を $1K$，$5K$ とした．語彙構築方法を TO としたモデルを，それぞれ **TO**$_{1K}$，**TO**$_{5K}$ と呼ぶ．同様に，語彙構築方法を ST としたモデルを，それぞれ **ST**$_{1K}$，**ST**$_{5K}$ と呼ぶ．ST-DO に対しては $|V_o|$ を $1K, 5K, 50K$ とした設定 (**ST-DO**$_{1K}$，**ST-DO**$_{5K}$，**ST-DO**$_{50K}$) で実験した．また生成用語彙からの単語の生成を用いずに原文書からの単語のコピーのみで要約を生成するモデル (Zhang et al. 2018) を **ST**$_0$ とした．

語彙構築方法が STO，ST，ST-DO の場合には $V_x = V_y$ のため原文書側の単語埋め込みパラメータ $\mathbf{E}_x$ は要約側の単語埋め込みパラメータ $\mathbf{E}_y$ と共有した．単語埋め込みの次元 (d) を 128，LSTM は 1 層で隠れ状態の次元 (h) を 256 とした．Table 7 に比較手法のパラメータ数を示す．

要約モデルの学習には Tesla T4 GPU を用いた．パラメータの更新は Adam (Kingma and Ba 2015) ($\alpha = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-9}$) を用いた．学習が安定するように，パラメータの勾配の L2 ノルムが 2.0 となるように gradient clipping を適用した．1 エポック終了するたびに開発データにおける目的関数の値を計算し，累計で 3 回，最良の値を改善しなければ学習を終了するようにした．ミニバッチサイズは CNN/Daily Mail で 16，NEWSROOM で 32 とした．結果的にすべての要約モデルは 30 エポック以内に学習を終了した．(Ott, Edunov, Baevski, Fan, Gross, Ng, Grangier, and Auli 2019) に従い，学習を高速化するために学習データは原文書の系列長で並び替え，原文書の系列長が近い学習事例が同じバッチになるようにした．各エポックの開始時にミニバッチの順序をランダムに並び替えて学習に用いた．

要約の生成時には開発データにおける損失が最も低い値となったモデルを用いた．また，要約生成には幅 4 のビームサーチを用いて要約の終端記号 (EOS) が生成されるまでデコード処理を繰り返した．要約のスコアは $\log p(\mathbf{y}|\mathbf{x})$ を単語数で割った値として短い要約が生成されにくくなるようにした．生成時の各時刻のデコード処理において，生成用語彙に登録されている UNK に対して割り当てられる確率を 0 にして UNK を生成しないようにした．これによって式 (23) の初項における V_o 中の UNK の確率は 0 となるが，原文書の単語は未知語であるかどうかにかかわらず，第二項における attention 確率が 0 以上となることに注意してほしい．CNN/Daily Mail に対しては (See et al. 2017) に従い，ビームサーチの過程で，35 単語未満もしくは 120 単語より多い単語で構成される要約はビームから除去し，極端に短いあるいは長い要約を生成しないように制限した．

5.6.2　実験結果

Table 8 に文書要約の標準的な自動評価尺度である ROUGE-1 (R-1)，ROUGE-2 (R-2)，ROUGE-L (R-L) (Lin 2004) の F 値を示す．ROUGE は自動要約結果と参照要約との間の単語の一致に基づく類似度である．ROUGE 値の計算には `pyrouge`[2] を用いた[3]．CNN/Daily Mail，NEWSROOM には要約長の制約がないため，F 値を評価値として採用した．各手法に対して乱数のシードを変えて 3 回実験を実施し，ROUGE 値の平均値を算出した．pointer-generator に加えて，LEAD 法の結果も記載する．LEAD 法はピリオドで原文書を文分割し，先頭の 3 文を要約として出力する手法である (Grusky et al. 2018; See et al. 2017)．また参考のため，既存研究の PtrGen+Cov (See et al. 2017) および Transformer$_{\mathrm{BASE}}$ (Zhang et al. 2019) の ROUGE 値も記載する．PtrGen+Cov は STO$_{50K}$ と同等のモデルであるが，学習の初期段階ではテキストの系列長制限を短くし，学習が進むにつれて制限長を長くするなど，学習の設定が異なる．Transformer$_{\mathrm{BASE}}$ はサブワード単位でテキストを分割し，要約モデルに Transformer (Vaswani, Shazeer, Parmar, Uszkoreit, Jones, Gomez, Kaiser, and Polosukhin 2017) を用いた手法である．提案手法は事前学習を用いないため，既存研究も事前学習を用いていないものとした．

提案手法の ST-DO$_{5K}$ は CNN/Daily Mail および NEWSROOM において既存の辞書構築手法よりも高い ROUGE 値を示した．これらの結果から，提案手法は pointer-generator の ROUGE 値改善に寄与していることがわかる．ST-DO$_{5K}$ と STO$_{SW:32K}$ の ROUGE 値を比較すると，CNN/Daily Mail の R-2 F は STO$_{SW:32K}$ が高いものの，ほかの ROUGE 値は ST-DO$_{5K}$ が高くなった．この結果から，提案手法によって語彙を構築することで，小さな語彙サイズでも要約特有な単語を生成でき，ROUGE 値改善に寄与していることがわかる．ST-DO$_{5K}$ と ST-DO$_{50K}$ の ROUGE 値を比較すると，ST-DO$_{5K}$ の方が高くなった．このことから，提案手法は語彙サイズを大きくしても，ROUGE 値の改善には寄与しないことがわかる．これは，語彙サイズを大きくしても，参照要約にあまり出現しない単語が多く登録され，その単語が生成されることが一因である．

CNN/Daily Mail において，PtrGen+Cov は STO$_{50K}$ と同等のモデルであることから，PtrGen+Cov の実装に対して提案手法を適用することでさらなる精度改善が期待される．NEWSROOM において，Transformer$_{\mathrm{BASE}}$ が最も高い ROUGE 値を示した．Transformer に対して pointer-generator を適用した手法も提案されており (Gehrmann et al. 2018; Wang et al. 2019b)，今後の課題として，提案手法とコピー機構を伴う Transformer を組み合わせることが考えら

[2] https://github.com/andersjo/pyrouge
[3] ROUGE 計算時のオプションは `-c 95 -r 1000 -n 2 -m -a` とした．

Table 9: 原文書には出現せず自動要約結果に出現する単語の異なり数

Method	CNN/Daily Mail	NEWSROOM
LEAD	0	0
STO_{50K}	292	633
$\text{STO}_{sw:32K}$	1801	1296
TO_{1K}	120	293
TO_{5K}	214	294
ST_{0}	0	0
ST_{1K}	97	160
ST_{5K}	190	333
(提案手法) ST-DO_{1K}	133	181
(提案手法) ST-DO_{5K}	259	465
(提案手法) ST-DO_{50K}	308	787

れる．

5.6.3 分析

自動要約結果に出現する単語の多様性を比較するために，原文書には出現せず自動要約結果に出現する単語の異なり数を Table 9 に示す．$\text{STO}_{SW:32K}$ はサブワードからなる系列を単語列にデコードした．サブワードを単位とした $\text{STO}_{SW:32K}$ はいずれの比較手法よりも多様な表現を生成している．一方で，ROUGE 値はほかの手法と同等か，やや下回る値となっていることから，参照要約には出現しない単語を多く生成している．提案手法は単語を単位としたほかの手法と比較して，多様な表現を生成している．このことから，提案手法は要約特有な表現を多く生成することに寄与している．また，$\text{STO}_{SW:32K}$ の自動要約結果において，原文書には出現せず自動要約結果のみに出現したすべての単語のうちで，参照要約にも出現したものの割合は CNN/Daily Mail では 7.4%，NEWSROOM では 13.4%となった．一方で，単語を単位とした手法の自動要約結果における平均は，CNN/Daily Mail で 19.8%，NEWSROOM では 15.8%となった．そのため，サブワードを単位とした $\text{STO}_{SW:32K}$ は参照要約に出現しない単語を多く生成する傾向にある．

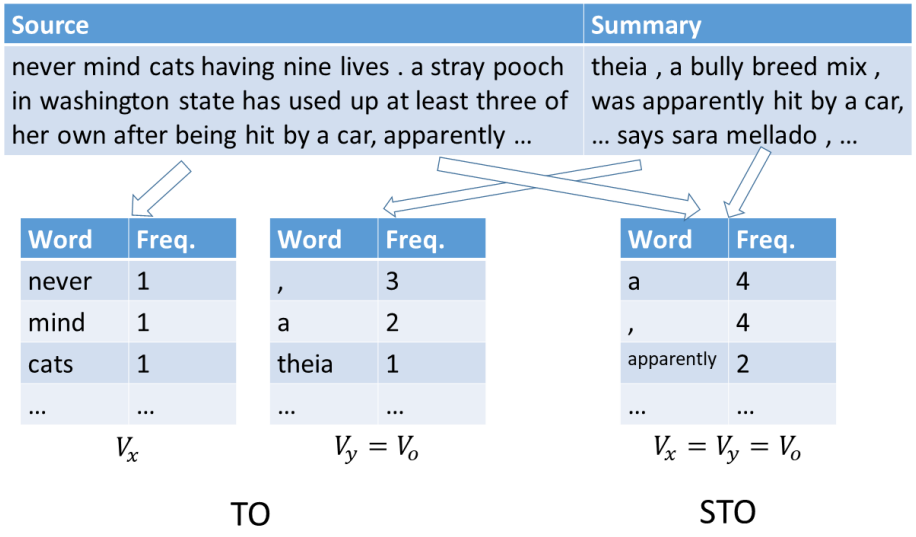
Table 10 に CNN/Daily Mail における自動要約結果の例を示す．提案手法は入力には出現せず，参照要約には出現する単語 *reveals* を自動要約結果として生成している．この単語は ST_{5K} の生成用語彙には存在しない一方で，ST-DO_{5K} の生成用語彙には存在する．

Table 10: CNN/Daily Mail における ST_{5K}，ST-DO_{5K} の生成結果．(...) 以降はスペースの都合で省略する．下線がついた単語は原文書には出現せず参照要約には出現した単語を表す．

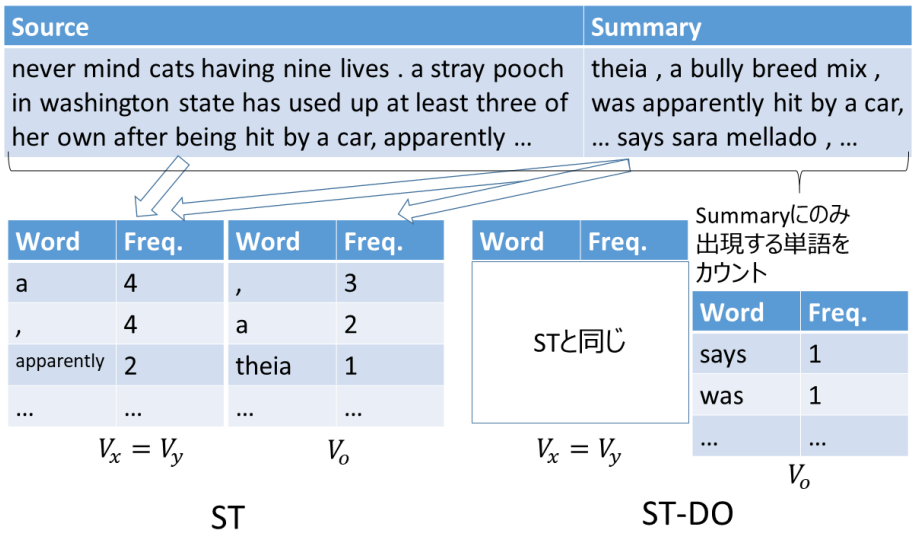
原文書
chelsea defender kurt zouma has revealed he has dreams of winning the ballon d'or . zouma has impressed at the heart of the chelsea defence this season and proved his versatility by seemlessly switching into a holding midfield role when called upon (...)
参照要約
kurt zouma reveals he has an ambition to win the ballon d'or . chelsea youngster has had an impressive first season at stamford bridge . 20-year-old has been compared (...)
ST_{5K} の生成結果
chelsea defender kurt zouma claims he has dreams of winning the ballon d'or . zouma has impressed at the heart of the chelsea defence this season . zouma is (...)
(提案手法) ST-DO_{5K} の生成結果
kurt zouma <u>reveals</u> he has dreams of winning the ballon d'or . zouma has impressed at the heart of the chelsea defence this season . zouma is hoping (...)

5.7　原文書と要約対の差分に基づく語彙構築手法のまとめ

本論文ではコピー機構を伴う生成型ニューラル要約モデルに対して，原文書には出現せず参照要約には出現する単語を対象に語彙を構築する手法を提案した．CNN/Daily Mail および NEWSROOM の実験において，提案手法は既存の語彙構築方法よりも少ない語彙サイズでも多様な単語を生成することができ，高い ROUGE 値となることを確認した．



(a) 既存手法: TO, STO



(b) 既存手法: ST, 提案手法: ST-DO

Figure 3: 既存手法と提案手法の語彙構築の違いの例．ST-DO は参照要約 (summary) には出現して，ペアとなる原文書には出現しない単語を対象に頻度をカウントする．

第6章　結論と今後の課題

6.1　結論

本論文では単一文書要約に対する生成型ニューラル要約手法の二つの重要な課題に着目し，それらの課題を緩和する新たな手法を提案することで要約性能の向上に貢献することを確認した．

一つ目の貢献は既存の要約長制御可能な生成型ニューラル要約モデルが要約長制約を超えた要約の生成を削減した点である．本論文では既存の生成型ニューラル要約モデルは要約長制約を超えた要約を生成することが少なくないという課題に対して，要約長制約内で品質の高い要約を生成するための学習方法を提案した．CNN/Daily Mail コーパスおよび毎日新聞コーパスでの実験によって，提案手法で既存のニューラル要約モデルを学習させることで，要約長制約を超えた要約の生成が削減できることを確認した．LSTM に基づく要約モデルに関しては，既存の学習手法である MLE よりも高い ROUGE 値を示した．毎日新聞コーパスにおける自動要約結果の人手による後編集実験では要約長制約内に要約を生成することで，要約長制約を超えた要約を編集するよりも後編集にかかる時間が削減することを確認した．

二つ目の貢献は，原文書と要約対の差分に基づく語彙構築手法によって要約特有の単語の生成割合を改善した点である．本論文では既存の語彙構築手法が，コピー機能を伴う生成型ニューラル要約モデルにおいて，冗長な単語を生成用語彙に少なからず含み，要約特有の生成の割合が低下するという課題に対して，原文書と参照要約の対を利用して，参照要約のみに出現する単語を対象にして生成用語彙を構築する手法を提案した．CNN/Daily Mail コーパスおよび NEWSROOM コーパスでの実験によって，提案手法は既存の語彙構築手法よりも ROUGE 値を改善させつつ，要約特有の単語を多く生成できることを確認した．

6.2　今後の課題

本章では本論文で取り組んだ課題以外にも重要な課題について述べる．

文書要約で良く用いられる ROUGE は表層の一致に基づく自動評価尺度であるため，意味的に類似する自動要約結果の評価が難しいという課題がある．このような課題に対処するために，二つのテキストの間の類似度を評価するニューラルネットワークを導入する研究が進

められている (Shimanaka, Kajiwara, and Komachi 2018; Mathur, Baldwin, and Cohn 2019; Sun and Nenkova 2019)．また Generative Adversarial Network によって言語モデルとは別にテキストの良さを評価するニューラルネットワークを導入し，交互に二つのニューラルネットワークを学習する方法も提案されている (Yu, Zhang, Wang, and Yu 2017; Lin, Li, He, Zhang, and Sun 2017)．

生成型ニューラル要約モデルは学習データにない語句の短縮表現や言い回しの生成が難しいという課題もある．この課題に対処するために，たとえば，外部知識の利用によって，固有表現などの短縮表現を獲得し，要約モデルに反映させる方法が考えられる．他に，教師なしデータの活用も考えられる．古くは単語埋め込みを skip-gram (Mikolov, Sutskever, Chen, Corrado, and Dean 2013) や GloVe (Pennington, Socher, and Manning 2014) といった手法で事前学習し，生成型ニューラル要約モデルの単語埋め込みの初期値として利用してきた．近年は単語埋め込みだけでなく，生成型ニューラル要約モデルのエンコーダや全体を事前学習する研究で要約性能の向上が報告されており (Liu and Lapata 2019; Wang et al. 2019a)，今後も研究が盛んにおこなわれる領域の一つと考える．

生成型ニューラル要約モデルによる自動生成結果を人手の作業支援に使う場合の課題もある．たとえば，要約の候補を複数提示する場合，できるだけ多様な候補作成が望ましいが，既存手法ではほぼ同じ単語からなる候補を複数生成してしまうという課題である．生成型ニューラル要約モデルを利用して複数の要約を生成するアルゴリズムはビームサーチがよく用いられる．しかしながらビームサーチは似た単語から構成される要約が生成されることが多い．これは，学習時において原文書と参照要約の対応が 1 対 1 として扱われて，原文書に対する参照要約の生成確率を最大化するように要約モデルが更新されることが一因である．最近では多様な複数の要約を生成する研究も進められている (Vijayakumar, Cogswell, Selvaraju, Sun, Lee, Crandall, and Batra 2018; Cho, Seo, and Hajishirzi 2019)．

最後に，本研究では生成型ニューラル要約モデルの高度化に焦点を当てたが，生成型ニューラル要約モデルによる人手の要約作業の支援を考えた場合，原文書を誤って言い換えた個所や原文書から漏れた重要な情報の人手による確認作業が課題となる．たとえば，原文書のどの箇所を参照しているのか，削除されているのか，といった情報を可視化することにより，作業の効率化が期待できる．

参考文献

Alfonseca, E., Pighin, D., and Garrido, G. (2013). "HEADY: News Headline Abstraction through Event Pattern Clustering." In *Proceedings of ACL*, pp. 1243–1253.

Ayana, Shen, S., Lin, Y., Tu, C., Zhao, Y., Liu, Z., and Sun, M. (2017). "Recent Advances on Neural Headline Generation." *J. Comput. Sci. Technol.*, **32** (4), pp. 768–784.

Banko, M., Mittal, V. O., and Witbrock, M. J. (2000). "Headline Generation Based on Statistical Translation." In *Proceedings of ACL*, pp. 318–325.

Barzilay, R. and Lee, L. (2004). "Catching the Drift: Probabilistic Content Models, with Applications to Generation and Summarization." In *Proceedings of NAACL-HLT*, pp. 113–120.

Bengio, Y., Boulanger-Lewandowski, N., and Pascanu, R. (2013). "Advances in Optimizing Recurrent Networks." In *Proceedings of ICASSP*.

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). "Latent Dirichlet Allocation." *J. Mach. Learn. Res.*, **3**, pp. 993–1022.

Chen, W., Grangier, D., and Auli, M. (2016). "Strategies for Training Large Vocabulary Neural Language Models." In *Proceedings of ACL*, pp. 1975–1985.

Chen, Y.-C. and Bansal, M. (2018). "Fast Abstractive Summarization with Reinforce-Selected Sentence Rewriting." In *Proceedings of ACL*, pp. 675–686.

Cheng, J. and Lapata, M. (2016). "Neural Summarization by Extracting Sentences and Words." In *Proceedings of ACL*, pp. 484–494.

Cho, J., Seo, M., and Hajishirzi, H. (2019). "Mixture Content Selection for Diverse Sequence Generation." In *Proceedings of EMNLP-IJCNLP*, pp. 3119–3129.

Durrett, G., Berg-Kirkpatrick, T., and Klein, D. (2016). "Learning-Based Single-Document Summarization with Compression and Anaphoricity Constraints." In *Proceedings of ACL*, pp. 1998–2008.

Fan, A., Grangier, D., and Auli, M. (2018). "Controllable Abstractive Summarization." In *Proceedings of Workshop on Neural Machine Translation and Generation*, pp. 45–54.

Filatova, E. and Hatzivassiloglou, V. (2004). "A Formal Model for Information Selection in Multi-Sentence Text Extraction." In *Proceedings of COLING*, pp. 397–403.

Filippova, K., Alfonseca, E., Colmenares, C. A., Kaiser, L., and Vinyals, O. (2015). "Sentence Compression by Deletion with LSTMs." In *Proceedings of EMNLP*, pp. 360–368.

Fum, D., Guida, G., and Tasso, C. (1986). "Tailoring Importance Evaluation to Reaer's Goals:: A Contribution to Descriptive Text Summarization." In *Proceedings of COLING*.

Gehring, J., Auli, M., Grangier, D., Yarats, D., and Dauphin, Y. N. (2017). "Convolutional Sequence to Sequence Learning." *CoRR*, **abs/1705.03122**.

Gehrmann, S., Deng, Y., and Rush, A. (2018). "Bottom-Up Abstractive Summarization." In *Proceedings of EMNLP*, pp. 4098–4109. Association for Computational Linguistics.

Genest, P.-E. and Lapalme, G. (2012). "Fully Abstractive Approach to Guided Summarization." In *Proceedings of ACL*, pp. 354–358.

Goldstein, J., Mittal, V., Carbonell, J., and Kantrowitz, M. (2000). "Multi-Document Summarization by Sentence Extraction." In *Proceedings of NAACL-ANLP: Workshop on Automatic Summarization - Volume 4*, pp. 40–48.

Grusky, M., Naaman, M., and Artzi, Y. (2018). "Newsroom: A Dataset of 1.3 Million Summaries with Diverse Extractive Strategies." In *Proceedings of NAACL-HLT*, pp. 708–719.

Gulcehre, C., Ahn, S., Nallapati, R., Zhou, B., and Bengio, Y. (2016). "Pointing the Unknown Words." In *Proceedings of ACL*, pp. 140–149.

Hermann, K. M., Kocisky, T., Grefenstette, E., Espeholt, L., Kay, W., Suleyman, M., and Blunsom, P. (2015). "Teaching Machines to Read and Comprehend." In *Proceedings of NIPS*, pp. 1693–1701.

Hirao, T., Isozaki, H., Maeda, E., and Matsumoto, Y. (2002). "Extracting Important Sentences with Support Vector Machines." In *Proceedings of COLING*, pp. 1–7.

Hirao, T., Yoshida, Y., Nishino, M., Yasuda, N., and Nagata, M. (2013). "Single-Document Summarization as a Tree Knapsack Problem." In *Proceedings of EMNLP*, pp. 1515–1520.

Hitomi, Y., Taguchi, Y., Tamori, H., Kikuta, K., Nishitoba, J., Okazaki, N., Inui, K., and Okumura, M. (2019). "A Large-Scale Multi-Length Headline Corpus for Improving Length-Constrained Headline Generation Model Evaluation." In *Proceedings of INLG*.

Hochreiter, S. and Schmidhuber, J. (1997). "Long Short-Term Memory." *Neural Comput.*, **9** (8), pp. 1735–1780.

Huang, L., Wu, L., and Wang, L. (2020). "Knowledge Graph-Augmented Abstractive Summarization with Semantic-Driven Cloze Reward." In *Proceedings of ACL*.

Kikuchi, Y., Hirao, T., Takamura, H., Okumura, M., and Nagata, M. (2014). "Single Document Summarization based on Nested Tree Structure." In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 315–320, Baltimore, Maryland. Association for Computational Linguistics.

Kikuchi, Y., Neubig, G., Sasano, R., Takamura, H., and Okumura, M. (2016). "Controlling Output Length in Neural Encoder-Decoders." In *Proceedings of EMNLP*.

Kingma, D. P. and Ba, J. (2015). "Adam: A Method for Stochastic Optimization." In *Proceedings of ICLR*.

Kiyono, S., Takase, S., Suzuki, J., Okazaki, N., Inui, K., and Nagata, M. (2018). "Reducing Odd Generation from Neural Headline Generation." In *Proceedings of PACLIC*.

Kudo, T. and Richardson, J. (2018). "SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing." In *Proceedings of EMNLP*, pp. 66–71.

Kupiec, J., Pedersen, J., and Chen, F. (1995). "A Trainable Document Summarizer." In *Proceedings of SIGIR*, pp. 68–73.

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., and Zettlemoyer, L. (2020). "BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension." In *Proceedings of ACL*.

Lin, C.-Y. (1999). "Training a Selection Function for Extraction." In *Proceedings of CIKM*, pp. 55–62.

Lin, C.-Y. (2004). "ROUGE: A Package for Automatic Evaluation of Summaries." In *Text Summarization Branches Out*, pp. 74–81.

Lin, K., Li, D., He, X., Zhang, Z., and Sun, M.-t. (2017). "Adversarial Ranking for Language Generation." In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (Eds.), *Advances in Neural Information Processing Systems 30*, pp. 3155–3165. Curran Associates, Inc.

Liu, Y. and Lapata, M. (2019). "Text Summarization with Pretrained Encoders." In *Proceedings of EMNLP-IJCNLP*, pp. 3728–3738.

Liu, Y., Luo, Z., and Zhu, K. (2018). "Controlling Length in Abstractive Summarization Using a Convolutional Neural Network." In *Proceedings of EMNLP*, pp. 4110–4119.

Luhn, H. P. (1958). "The Automatic Creation of Literature Abstracts." *IBM J. Res. Dev.*, **2** (2), pp. 159–165.

Luong, T., Pham, H., and Manning, C. D. (2015). "Effective Approaches to Attention-based Neural Machine Translation." In *Proceedings of EMNLP*, pp. 1412–1421.

Manning, C., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S., and McClosky, D. (2014). "The Stanford CoreNLP Natural Language Processing Toolkit." In *Proceedings of ACL*, pp. 55–60.

Mathur, N., Baldwin, T., and Cohn, T. (2019). "Putting Evaluation in Context: Contextual Embeddings Improve Machine Translation Evaluation." In *Proceedings of ACL*, pp. 2799–2808.

McCann, B., Keskar, N. S., Xiong, C., and Socher, R. (2018). "The Natural Language Decathlon: Multitask Learning as Question Answering." *arXiv preprint arXiv:1806.08730*.

McDonald, R. (2007). "A Study of Global Inference Algorithms in Multi-document Summarization." In *Proceedings of ECIR*, pp. 557–564.

McKeown, K. and Radev, D. R. (1995). "Generating Summaries of Multiple News Articles." In *Proceedings of SIGIR*, pp. 74–82.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). "Distributed Representations of Words and Phrases and their Compositionality." In Burges, C. J. C., Bottou, L., Welling, M., Ghahramani, Z., and Weinberger, K. Q. (Eds.), *Proceedings iof NIPS*, pp. 3111–3119.

Nallapati, R., Zhai, F., and Zhou, B. (2017). "SummaRuNNer: A Recurrent Neural Network Based Sequence Model for Extractive Summarization of Documents." In *Proceedings of AAAI*, pp. 3075–3081.

Nallapati, R., Zhou, B., dos Santos, C., Gülchere, C., and Xiang, B. (2016). "Abstractive Text Summarization using Sequence-to-sequence RNNs and Beyond." In *Proceedings of CoNLL*, pp. 280–290.

Nishikawa, H., Arita, K., Tanaka, K., Hirao, T., Makino, T., and Matsuo, Y. (2014). "Learning to Generate Coherent Summary with Discriminative Hidden Semi-Markov Model." In *Proceedings of COLING*, pp. 1648–1659.

Och, F. J. (2003). "Minimum Error Rate Training in Statistical Machine Translation." In *Proceedings of ACL*, pp. 160–167.

Osborne, M. (2002). "Using Maximum Entropy for Sentence Extraction." In *Proceedings of the ACL-02 Workshop on Automatic Summarization - Volume 4*, pp. 1–8.

Ott, M., Edunov, S., Baevski, A., Fan, A., Gross, S., Ng, N., Grangier, D., and Auli, M. (2019). "fairseq: A Fast, Extensible Toolkit for Sequence Modeling." In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pp. 48–53, Minneapolis, Minnesota.

Parveen, D., Ramsl, H.-M., and Strube, M. (2015). "Topical Coherence for Graph-based Extractive Summarization." In *Proceedings of EMNLP*, pp. 1949–1954.

Parveen, D. and Strube, M. (2015). "Integrating Importance, Non-redundancy and Coherence in Graph-based Extractive Summarization." In *Proceedings of IJCAI*, pp. 1298–1304.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani,

A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. "PyTorch: An Imperative Style, High-Performance Deep Learning Library." In *Proceedings of NeurIPS, editor = H. Wallach and H. Larochelle and A. Beygelzimer and F. d'Alché-Buc and E. Fox and R. Garnett, pages = 8024–8035, year = 2019, url = http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf*.

Paulus, R., Xiong, C., and Socher, R. (2018). "A Deep Reinforced Model for Abstractive Summarization." In *Proceedings of ICLR*.

Pennington, J., Socher, R., and Manning, C. (2014). "Glove: Global Vectors for Word Representation." In *Proceedings of EMNLP*, pp. 1532–1543.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. (2019). "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer." *ArXiv*, **abs/1910.10683**.

Ranzato, M., Chopra, S., Auli, M., and Zaremba, W. (2015). "Sequence Level Training with Recurrent Neural Networks." *CoRR*, **abs/1511.06732**.

Rush, A. M., Chopra, S., and Weston, J. (2015). "A Neural Attention Model for Abstractive Sentence Summarization." In *Proceedings of EMNLP*, pp. 379–389.

See, A., Liu, P. J., and Manning, C. D. (2017). "Get To The Point: Summarization with Pointer-Generator Networks." In *Proceedings of ACL*, pp. 1073–1083.

Shen, D., Sun, J.-T., Li, H., Yang, Q., and Chen, Z. (2007). "Document Summarization Using Conditional Random Fields." In *Proceedings of IJCAI*, pp. 2862–2867.

Shen, S., Cheng, Y., He, Z., He, W., Wu, H., Sun, M., and Liu, Y. (2016). "Minimum Risk Training for Neural Machine Translation." In *Proceedings of ACL*, pp. 1683–1692.

Shen, X., Zhao, Y., Su, H., and Klakow, D. (2019). "Improving Latent Alignment in Text Summarization by Generalizing the Pointer Generator." In *Proceedings of EMNLP-IJCNLP*.

Shimanaka, H., Kajiwara, T., and Komachi, M. (2018). "Metric for Automatic Machine Translation Evaluation based on Universal Sentence Representations." In *Proceedings of NAACL: Student Research Workshop*, pp. 106–111.

Sun, S. and Nenkova, A. (2019). "The Feasibility of Embedding Based Automatic Evaluation for Single Document Summarization." In *Proceedings of EMNLP-IJCNLP*, pp. 1216–1221.

Suzuki, J. and Nagata, M. (2017). "Cutting-off Redundant Repeating Generations for Neural Abstractive Summarization." In *Proceedings of EACL*, pp. 291–297.

Takamura, H. and Okumura, M. (2009). "Text Summarization Model Based on Maximum Coverage Problem and its Variant." In *Proceedings of EACL*, pp. 781–789.

Takase, S. and Okazaki, N. (2019). "Positional Encoding to Control Output Sequence Length." In *Proceedings of NAACL*, pp. 3999–4004.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. (2017). "Attention is All you Need." In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (Eds.), *Proceedings of NIPS*, pp. 5998–6008.

Vijayakumar, A. K., Cogswell, M., Selvaraju, R. R., Sun, Q., Lee, S., Crandall, D. J., and Batra, D. (2018). "Diverse Beam Search for Improved Description of Complex Scenes." In *Proceedings of AAAI*, pp. 7371–7379.

Wang, L., Zhao, W., Jia, R., Li, S., and Liu, J. (2019a). "Denoising based Sequence-to-Sequence Pre-training for Text Generation." In *Proceedings of EMNLP-IJCNLP*, pp. 4001–4013.

Wang, L., Zhao, W., Jia, R., Li, S., and Liu, J. (2019b). "Denoising based Sequence-to-Sequence Pre-training for Text Generation." In *Proceedings of EMNLP-IJCNLP*, pp. 4003–4015.

Witbrock, M. J. and Mittal, V. O. (1999). "Ultra-Summarization: A Statistical Approach to Generating Highly Condensed Non-Extractive Summaries." In *Proceedings of SIGIR*, pp. 315–316.

Xu, J. and Durrett, G. (2019). "Neural Extractive Text Summarization with Syntactic Compression." In *Proceedings of EMNLP*.

Yan, Y., Qi, W., Gong, Y., Liu, D., Duan, N., Chen, J., Zhang, R., and Zhou, M. (2020). "ProphetNet: Predicting Future N-gram for Sequence-to-Sequence Pre-training." *ArXiv*, **abs/2001.04063**.

Yu, L., Zhang, W., Wang, J., and Yu, Y. (2017). "SeqGAN: Sequence Generative Adversarial Nets with Policy Gradient." In *Proceedings of AAAI*, pp. 2852–2858.

Zhang, F., Yao, J.-g., and Yan, R. (2018). "On the Abstractiveness of Neural Document Summarization." In *Proceedings of ACL*, pp. 785–790.

Zhang, J., Zhao, Y., Saleh, M., and Liu, P. J. (2019). "PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization." *ArXiv*, **abs/1912.08777**.

Zhou, Q., Yang, N., Wei, F., and Zhou, M. (2017). "Selective Encoding for Abstractive Sentence Summarization." In *Proceedings of ACL*, pp. 1095–1104.

研究業績

論文誌

- 牧野拓哉，岩倉友哉: ニューラル要約モデルのための原文書と要約対の差分に基づく語彙構築方法. 人工知能学会論文誌 35 巻 6 号
- 牧野拓哉，野呂智哉，岩倉友哉: 自動生成した学習データを用いた文書分類器に基づく FAQ 検索システム. 自然言語処理 24 巻 1 号

査読付き国際会議

- Takuya Makino, Tomoya Iwakura, Hiroya Takamura, Manabu Okumura: Global Optimization under Length Constraint for Neural Text Summarization. ACL 2019: 1039-1048
- Takuya Makino, Tomoya Noro, Hiyori Yoshikawa, Tomoya Iwakura, Satoshi Sekine, Kentaro Inui: An FAQ Search Training Method Based on Automatically Generated Questions. AIRS 2018: 67-73
- Takuya Makino, Tomoya Iwakura: A Boosted Supervised Semantic Indexing for Reranking. AIRS 2017: 16-28
- Takuya Makino, Tomoya Noro, Tomoya Iwakura: An FAQ Search Method Using a Document Classifier Trained with Automatically Generated Training Data. PRICAI 2016: 295-305
- Takuya Makino: Automatic Selection of Reference Pages in Wikipedia for Improving Targeted Entities Disambiguation. EACL 2014: 106-110
- Takuya Makino, Hiroya Takamura, Manabu Okumura: Balanced Coverage of Aspects for Text Summarization. CIKM 2012: 1742-1746

ワークショップ

- Takuya Makino, Seiji Okura, Seiji Okajima, Shuangyong Song, Hiroko Suzuki: FLL: Answering World History Exams by Utilizing Search Results and Virtual Examples. NTCIR 2014
- Takuya Makino, Seiji Okajima, Tomoya Iwakura: FLL: Local Alignments based Approach for NTCIR-10 RITE-2. NTCIR 2013
- Hajime Morita, Takuya Makino, Tetsuya Sakai, Hiroya Takamura, Manabu Okumura: TTOKU Summarization Based Systems at NTCIR-9 1CLICK task. NTCIR 2011
- Takuya Makino, Hiroya Takamura, Manabu Okumura: Balanced Coverage of Aspects for Text Summarization. TAC 2011

国内会議

- 牧野拓哉, 岩倉友哉, 高村大也, 奥村学: Minimum Risk Training に基づく要約モデルの出力長制御. 言語処理学会 2018
- 牧野拓哉, 野呂智哉, 吉川和, 岩倉友哉, 関根聡, 乾健太郎: 自動生成した質問に基づく質問応答学習手法の提案と評価. 言語処理学会 2018
- 牧野拓哉, 野呂智哉: 自動収集した学習データを用いた文書分類器に基づく FAQ 検索システム. 言語処理学会 2016 言語処理の応用ワークショップ
- 牧野拓哉, 高村大也, 奥村学: アスペクトの被覆を実現するための最小値最大化問題に基づく文書要約モデル. 言語処理学会 2012
- 牧野拓哉, 高村大也, 奥村学: アスペクト被覆を可能にした最小値最大化問題に基づく文書要約モデル. 情報処理学会 第 204 回自然言語処理研究会

学会活動

- 2016 年–2019 年 情報処理学会 自然言語処理研究会 運営委員

博士論文の要件とする研究業績 (抜粋)

- 牧野拓哉，岩倉友哉: ニューラル要約モデルのための原文書と要約対の差分に基づく語彙構築方法. 人工知能学会論文誌 35 巻 6 号
- Takuya Makino, Tomoya Iwakura, Hiroya Takamura, Manabu Okumura: Global Optimization under Length Constraint for Neural Text Summarization. ACL 2019: 1039-1048

謝辞

本学位論文は著者が東京工業大学大学工学院通信工学系に在籍中 の成果をまとめたものです．本論文の審査を引き受けてくださった小林隆夫教授，熊澤逸夫教授，高村大也教授，船越孝太郎准教授，篠崎隆宏准教授に感謝申し上げます．長い学位審査の各段階において俯瞰的な視野からアドバイスを頂き，研究を加速させることができました．本研究の遂行に当たり，指導教員であるに奥村学教授，高村大也教授には修士課程在学時から丁寧にご指導を賜りました．感謝申し上げます．奥村先生は，なかなか論文が採択されず，くじけそうになっている私に，あきらめないようにコメントをくださりました．そのおかげで，粘り強く論文を修正することができ，結果として自然言語処理分野で最も難関である ACL に論文が採択されることができました．あきらめずに自分の研究を洗練していく作業は，私にとってこれからの社会人生活でも非常に価値のある経験となりました．また私の結婚式におきましては，スピーチを受けてくださり，私の修士課程時代の話を友人や同僚をはじめ，両家の親族に伝えていただいたことや，富士通研究所から参加していた上司や同僚に社会人博士を推薦してくださったことはとても嬉しかったです．高村先生は，研究を進める中で，常に自分の研究を魅力的に見せるにはどうしたらよいかということを学んだように感じています．論文の書き方をはじめ，人を魅了するような研究を高村先生のようにできるように，より一層研究に励みたいと思います．また社会人博士中はできませんでしたが，修士課程在学中に研究の合間にフットサルができたことは楽しかったです．また言語処理学会などで一緒にフットサルができることが楽しみにしております．社会人博士として学位を取得するにあたり，ご指導をいただきました富士通研究所の岩倉友哉さんにはことあるごとに相談に乗っていただき，研究を進めることができました．修士課程を卒業したばかりで知識や技術が未熟だった私に様々なことをご指導いただきました．感謝申し上げます．最後に，博士課程に進学するにあたり，私をサポートしてくれた家族，特に妻に感謝を申し上げます．論文が通らなかった時も元気に励ましてくれたことは支えになりました．多くの方に支えられてここまで本論文を作成することができました．