学修番号 19860630

修士論文

マルチモーダル機械翻訳のための効果的な単語分割と単語分散表現の学習

平澤　寅庄

2021 年 2 月 19 日

東京都立大学大学院
システムデザイン研究科 情報科学域

平澤　寅庄

審査委員：

小町 守 准教授　（主指導教員）

山口 亨 教授　（副指導教員）

高間 康史 教授　（副指導教員）

マルチモーダル機械翻訳のための効果的な単語分割と単語分散表現の学習*

平澤　寅庄

修論要旨

近年、大規模な単言語コーパスで事前学習された単語分散表現は、対訳コーパスが乏しいドメインにおいて、ニューラル機械翻訳のモデルの性能向上に貢献することが示されてきた。本論文では、画像を補助的な入力として使用するマルチモーダル機械翻訳に着目し、単言語コーパスから事前学習される単語分散表現の効果的な利用方法を明らかにする。

マルチモーダル機械翻訳では、目的言語側の翻訳を生成する際に、原言語の入力文だけではなく、視覚的情報などの非言語的情報を用いる。近年、Attention 機構を持つエンコーダ・デコーダモデルの登場により、ニューラル機械翻訳は伝統的な統計的機械翻訳の性能を凌駕している。この新しい枠組みの中で、機械翻訳は系列変換問題として扱われ、デコードするときに入力文に注意をするように学習される。マルチモーダル機械翻訳では、このエンコーダ・デコーダモデルを拡張し、モデルがエンコードやデコード（もしくは、その両方）をする際に、文だけではなく、画像にも着目するよう訓練される。これにより、入力文や目的言語に含まれる曖昧性を解消することで、翻訳性能が向上されることが期待される。

しかし、マルチモーダル機械翻訳には、対訳コーパスが少ないという欠点がある。原言語・目的言語から構成される一般的な機械翻訳の対訳コーパスに比べ、マルチモーダル機械翻訳に使用される対訳コーパスは原言語・目的言語・画像の組を収集する必要があり、データセットの規模は小さい。現在主に使用されているマルチモーダルの対訳コーパスの規模は 3 万文程度であり、他の機械翻訳のデータセットに比べて非常に小さい。そのため、高品質なニューラル機械翻訳を訓練することは難しい。

*東京都立大学大学院 システムデザイン研究科 情報科学域 修士論文, 学修番号 19860630, 2021 年 2 月 19 日.

この問題を低減するため、本研究では大規模な単言語コーパスに着目し、マルチモーダル機械翻訳モデルの性能を改善する手法を提案する。本研究では、まず大規模な単言語コーパスを用いて単語分割を行う方法について提案する。ニューラル機械翻訳モデルでは、パラメーター数の制限から、すべての単語を使うことは難しく、単語レベルの機械翻訳モデルでは未知語が発生し、翻訳性能を低下させる。一般に、未知語の問題に対処するため、単語をより小さな単位であるサブワードに分解する Byte Pair Encoding が用いられる。本研究では、まず対訳コーパスから学習されたサブワードの効果を確認したのち、大規模な単言語コーパスから学習されたサブワードをマルチモーダル機械翻訳モデルに適用する手法を提案し、その翻訳性能を評価する。

次に、大規模な単言語コーパスで訓練された単語分散表現を使い、マルチモーダル機械翻訳モデルを初期化する手法を検討する。自然言語処理の多岐にわたるタスクにおいて、事前学習された単語分散表現はニューラル・ネットワークモデルを使う際に重要な役割を担うと考えられている。ニューラル機械翻訳においては、事前学習された単語分散表現は低資源領域において有用であることが示された。先行研究では、FastText フレームワークで事前学習された単語分散表現を用いて、エンコーダ・デコーダの単語埋め込み層を初期化することで、低資源の言語対における翻訳性能を向上させた。しかし、事前学習された単語分散表現では、特定の単語が他の複数の単語の近傍として頻繁に出現する問題が存在する。この問題は機械学習ではハブネス問題と呼ばれ、事前学習された単語分散表現の効用を低下させることが知られている。先行研究では、意味や語彙のラベルを人手で注釈をつけることでこの問題に対処したが、高価な人手により作業が必要であるという問題があった。一方で、例えば、word analogy タスクでは単語分散表現に含まれる局所的なバイアスや全域的なバイアスを取り除くことで、モデルの予測精度が向上することが報告されている。本研究では、マルチモーダル機械翻訳で使用する単語分散表現に対して、バイアスを取り除くことでハブネス問題を解消する手法を提案し、翻訳精度の向上を確認した。

最後に、本研究ではデコーダの出力層で単語分散表現を予測するマルチモーダル機械翻訳モデルを提案する。一般的なデコーダの出力層では、出力単語の分布を予測し、正解単語の予測尤度を最大化されるように学習されるが、このとき単語間の

類似性は考慮されない。しかし、提案手法では予測した単語（分散表現）と正解単語（分散表現）を比較することで、予測単語と正解単語の類似度に応じた損失を用いてモデルを訓練することができる。これにより、事前学習した単語分散表現に含まれる単語間の関係性をニューラル機械翻訳に組み込むことができる。

本論文の構成は以下の通りである。第 1 章では、本研究の提案、貢献、概要について述べる。第 2 章と第 3 章ではそれぞれ、ニューラル機械翻訳モデル、および、単語分散表現の技術について紹介する。第 4 章ではマルチモーダル機械翻訳で使われるデータセットや翻訳モデルについて紹介する。第 5 章では大規模な単言語コーパスを使用してサブワード分割する手法を評価する。第 6 章ではバイアスを消去された単語分散表現を機械翻訳モデルに組み込む手法を提案する。第 7 章では単語分散表現を予測するマルチモーダル機械翻訳モデルを提案する。最後に、第 8 章では本研究のまとめを述べる。

Effective Tokenization and Pre-training Word Embedding for Multimodal Machine Translation*

Tosho Hirasawa

Abstract

Recently, pre-trained word embedding have been proved useful for neural machine translation (NMT) models in low-resource domain. In this thesis, we explore approaches to leverage word embedding pre-trained on large-scale monolingual corpus to improve multimodal machine translation models that use images as an auxiliary modality.

In **multimodal machine translation**, a target sentence is translated from a source sentence together with related non-linguistic information such as visual information. Recently, NMT models have superseded traditional statistical machine translation model owing to the introduction of the attentive encoder-decoder model, in which machine translation is treated as a sequence-to-sequence learning problem and is trained to pay attention to the source sentence while decoding. The latest multimodal machine translation models extend attentive encoder-decoder models to pay attention to both sentences and images while encoding or/and decoding. These multimodal models are supposed to be useful for disentangling ambiguity in source and target sentences, and sequentially improve translation quality.

However, the dataset available to train multimodal machine translation models has limited size of data. Comparing to the dataset for text-only models that consists of source and target sentences, the multimodal dataset is required

*Master's Thesis, Department of Computer Science, Graduate School of System Design, Tokyo Metropolitan University, Student ID 19860630, February 19, 2021.

to contain additional images, which makes the composition more expensive and of limited size. The well-established dataset for multimodal machine translation has only 30 thousands samples, which is quite small for training NMT models of good quality.

To alleviate this problem, I propose to leverage large-scale monolingual corpora to improve multimodal machine translation models. Firstly, I propose to apply subword tokenization using the subwords learned from a large-scale monolingual corpus. Subword tokenization is considered as an essential technique for NMT models to handle out-of-word tokens. However, subwords learned from a small corpus may lead improper subword tokenization. To address this problem, we use a large-scale monolingual corpus to comprise proper subwords, which bring proper tokenization.

Secondary, I propose to initialize NMT models with word embedding that is trained on a large-scale monolingual corpus. The pre-trained word embedding is proved to be useful for neural models in a wide range of natural language processing task. The previous work in NMT domain reveals that a NMT model in low-resource domain improves its translation quality by initializing the embedding layers in its encoder and decoder with FastText word embedding. However, pre-trained word embedding in high dimensional spaces has been reported to suffer from the hubness problem, in which certain words appear frequently in the neighbors for other words. This phenomenon harms the utility of the pre-trained word embedding. A previous work proposed annotating sense labels or lexical labels to address this problem, which is expensive and time-consuming. On the other hand, it is known to be effective to debias the word embedding based on their bias for word analogy tasks, which does not require extra expensive annotations and references. In this thesis, we leverage pre-trained word embedding for multimodal NMT models and show that debiasing of pre-trained word embedding improves translation quality.

Lastly, I introduce an NMT model with embedding prediction for multimodal machine translation that fully uses pre-trained word embedding to improve the

translation accuracy for rare words. Other than the traditional NMT model that outputs the distribution over its vocabulary, the model with embedding prediction predicts the embedding of a output word. This makes NMT model to learn the gap between the predicted word and the ground-truth word in more fine granular way than predicting the distribution and using cross-entropy loss. As the result, the relations between words will be transferred into the multimodal NMT model.

This paper comprises as follows: In chapter 1, we introduce the overview of this thesis. In chapter 2, 3 and 4 we introduce neural machine translation, word embedding, and multimodal NMT, respectively. In chapter 5, we propose the subword approach using subword codes learned from a large-scale monolingual corpus. In chapter 6, we propose the debiasing method for word embedding trained on a large-scale monolingual corpus. In chapter 7, we propose a multimodal NMT model with embedding prediction. Finally, in chapter 8, we summarize this work.

目次

図目次

第 1 章　はじめに

近年、機械翻訳（Machine Translation: MT）では、ニューラル機械翻訳（Neural Machine Translation: NMT）が盛んに研究されている。入力文の各トークン（単語）をベクトル（分散表現）に符号化するエンコーダとその分散表現から出力トークンを 1 つずつ予測するデコーダを組み合わせた Encoder–Decoder モデルと呼ばれる系列変換モデルで初めて、NMT が従来の統計的機械翻訳（Phrase-Based Statistical Machine Translation: PBSMT）を上回る性能を達成している [1]。また、デコーダで予測する際、出力トークン毎にエンコーダの分散表現に対する重み付けを動的に変える Attention 機構の登場により、Encoder–Decoder モデルで問題であった長文の翻訳性能を大きく改善した [2]。

マルチモーダル機械翻訳（Multimodal Machine Translation: MMT）は、翻訳を行う際に、入力文に加え入力文と紐付けられた非言語情報（例えば、画像や音声）の 2 つ以上の入力形式（モダリティ）を用いる機械翻訳技術のことである。本研究では MMT のうち、入力文とそれに対応する画像を利用した MMT に着目する。翻訳において画像を考慮することにより、目的言語側で複数の意味を持つように原言語の単語の語義曖昧性を解消することや、原言語では存在しないが目的言語に存在する言語現象（例えば、男性・女性・中性名詞などの文法的性）を決定することが考えられる。このことから、画像を利用する MMT モデルは、入力文のみを利用する NMT モデルと比べ、翻訳性能が向上することが期待される。MMT 技術の発展により、ニュースや映像などのマルチモーダルな情報を扱う産業での応用が望まれる。

MMT においても近年の NMT の発展とともに、Attention 機構を用いた Encoder–Decoder モデルを基にしたニューラルモデルが提案されており、いくつかのモデルでは入力文のみを使用した NMT モデルを上回る性能を達成している。MMT モデルにおいて、画像はまず Convolutional Neural Network（CNN）などを用いたニューラル画像識別モデルによって単一のベクトル（全域特徴量）や領域ごとのベクトル列（局所特徴量）へと抽出される。その後、画像の特徴量はエンコーダ [3] やデコーダ [4, 5] に組み込まれる。例えば、Calayan ら [4] は画像から抽出した全域特徴量を用いてデコーダを初期化した。また、Calixto ら [5] は入力文

に加え、画像から抽出した局所特徴量に対して Attention 機構を適用するデコーダを用いるモデルを提案した。

しかし、MMT タスクでは、モデルを訓練するのに用いるマルチモーダル対訳コーパスはデータ数が少ないという欠点がある。一般的な NMT モデルの訓練に使用する対訳コーパスは原言語文と目的言語文のみで構成されるのに比べ、マルチモーダル対訳コーパスは原言語文・目的言語文に加え、画像が必要であるため、利用できるデータセットの規模は小さくなる。例えば、機械翻訳の国際的なコンペティションである Conference on Machine Translation (WMT) Shared Task において、ニュース翻訳タスク向けに提供される対訳コーパスは最大で 8,260 万の文対を含むのに対して、MMT モデルの訓練で使用される対訳コーパスに含まれるデータは 3 万程度である。NMT においては学習に使用するデータサイズが少ないと、モデルの翻訳性能が顕著に低下することが知られており [6]、高性能な MMT モデルを訓練する際の大きな問題となっている。これまで、この問題に対応するため、画像を含まない対訳コーパスを利用して MT モデルを事前訓練する手法が提案されており、顕著に翻訳性能を向上させることをわかっている [7]。

本論文では、単一言語で構成される単言語コーパスに着目し、MMT モデルの性能を向上させるための手法を提案する。はじめに、大規模な単言語コーパスを用いてサブワード分割を行う方法について提案する。NMT モデルは計算機スペックの制限により、モデルで使用できるパラメーターのサイズに上限がある。そのため、訓練データに含まれるすべての単語をモデルの出力語彙として使うことは難しく、NMT モデルでは使用する語彙を限定することが一般的である。このような NMT モデルでは、訓練データには含まれるがモデルの語彙には含まれない単語（未知語）が発生し、モデルの翻訳性能を低下する。このような未知語の問題に対処するため、多くの NMT モデルでは、単語をより小さな単位であるサブワードに分解する Byte Pair Encoding が用いられる。本研究では、まず対訳コーパスから学習されたサブワードの効果を確認したのち、大規模な単言語コーパスから学習されたサブワードをマルチモーダル機械翻訳モデルに適用する手法を提案し、その翻訳性能を評価する。

次に本研究では、大規模な単言語コーパスから学習した単語分散表現を MMT モデルに組み込む手法に加え、組み込む単語分散表現に含まれるバイアスを取り除く

ことを提案し、MMT モデルへ適用したときに翻訳精度が向上すること確認した。近年、大規模な単言語コーパスで事前学習された単語分散表現は、対訳コーパスが乏しいドメインにおいて、NMT モデルの性能向上に貢献することが報告されている [8]。だが、事前学習された単語分散表現では、特定の単語が他の複数の単語の近傍として頻繁に出現する問題が存在する。この問題は機械学習ではハブネス問題と呼ばれ、事前学習された単語分散表現の効用を低下させることが知られている。先行研究では、意味や語彙のラベルを人手で注釈をつけることでこの問題に対処したが、高価な人手により作業が必要であるという問題があった [9]。一方で、語義曖昧性解消タスクでは単語分散表現に含まれる局所的なバイアスや全域的なバイアスを取り除くことで、モデルの予測精度が向上することが報告されている [10, 11]。本論文では、語義曖昧性解消タスクで提案されたバイアス解消手法を事前学習した単語分散表現に適用する。

最後に、本研究ではデコーダの出力層で単語分散表現を予測する MMT モデルを提案する。一般的なデコーダの出力層では、出力単語の分布を予測し、正解単語の予測尤度を最大化されるように学習されるが、このとき単語間の類似性は考慮されない。しかし、提案手法では予測した単語（分散表現）と正解単語（分散表現）を比較することで、予測単語と正解単語の類似度に応じた損失を用いてモデルを訓練することができる。これにより、事前学習した単語分散表現に含まれる単語間の関係性を MMT モデルに組み込むことができる。

本論文の構成は以下の通りである。第 1 章では、本研究の提案、貢献、概要について述べる。第 2 章と第 3 章ではそれぞれ、ニューラル機械翻訳モデル、および、単語分散表現の技術について紹介する。第 4 章ではマルチモーダル機械翻訳で使われるデータセットや翻訳モデルについて紹介する。第 5 章では大規模な単言語コーパスを使用してサブワード分割する手法を評価する。第 6 章ではバイアスを消去された単語分散表現を機械翻訳モデルに組み込む手法を提案する。第 7 章では単語分散表現を予測するマルチモーダル機械翻訳モデルを提案する。最後に、第 8 章では本研究のまとめを述べる。

第 2 章　ニューラル機械翻訳

ニューラル機械翻訳（NMT）モデルとは計算機を使用して原言語の内容を出力言語へと翻訳するモデルのことである。近年、機械翻訳は Encoder–Decoder と呼ばれる系列変換モデル Attention 機構の登場により、従来のフレーズベース統計的機械翻訳を上回る性能を達成している。

この章では、まず現在一般的に使用されている NMT モデルについて概説し、NMT モデルを訓練する方法と訓練した NMT モデルを使用して目的言語文を推論する方法について説明する。その後、機械翻訳の出力の評価方法について紹介する。

2.1　ニューラル機械翻訳モデル

ニューラル機械翻訳モデルでは N 個のトークンで構成される原言語の文 $\mathbf{x} = \{x_1, x_2, \cdots, x_N\}$ を M 個のトークンで構成される目的言語の文 $\mathbf{y} = \{y_1, y_2, \cdots, y_M\}$ へ翻訳するように学習される。

2.1.1　系列変換モデル

系列変換モデルは 2 つの独立した再帰ニューラル・ネットワーク（recurrent neural network: RNN）で構成される。一方の RNN は入力文 $\mathbf{x}$ を高次元の実数ベクトルである隠れ状態に符号化（エンコード）し、もう一方の RNN は隠れ状態を目的言語の文に復号（デコード）するため、エンコーダ・デコーダモデルとも呼ばれる。このエンコーダ・デコーダモデルは Sutskever ら [1] により初めて PBSMT を上回る性能を達成した。

■エンコーダ　エンコーダでは入力文 $\mathbf{x}$ を隠れ状態 $\mathbf{h} = \{\mathbf{h}_1, \mathbf{h}_2, \cdots, \mathbf{h}_N\}$ に符号化する。i 番目の隠れ状態は $i-1$ 番目の隠れ状態と i 番目の入力文のトークンから RNN を使用して計算される。Sutskever ら [1] は RNN の一つである long short-term memory(LSTM) [12] を使用した。

$$\boldsymbol{h}_i = \mathrm{LSTM}(\boldsymbol{h}_{i-1}, \boldsymbol{e}_{enc}(x_i)) \tag{1}$$

$i \in [1, N]$ は入力文の位置である。$\boldsymbol{e}_{enc}(x_i)$ はトークン x_i の埋め込み表現である。

h_0 は一般に要素の値がゼロのベクトルで初期化される。h_i は h_{i-1} を使用して計算されるため、h_i はそれまでの入力 $x_{\leq i}$ のすべての情報を持っていると考えられる。

■デコーダ　デコーダでは入力文から得られた隠れ状態 $\mathbf{h}$ とそれまでに出力したトークン列 $\hat{\mathbf{y}}_{<j}$ から、$j \in [1, M]$ 番目の出力言語のトークン $\hat{y}_j$ を予測する。デコーダではまず、j 番目の隠れ状態 $\boldsymbol{s}_j$ を $j-1$ 番目の隠れ状態と $j-1$ 番目の出力トークンから LSTM を使用して計算される。

$$\boldsymbol{s}_j = \mathrm{LSTM}(\boldsymbol{s}_{j-1}, \boldsymbol{e}_{\mathrm{dec}}(\hat{y}_{j-1})) \tag{2}$$

$\boldsymbol{e}_{dec}(\hat{y}_{i-1})$ は直前に予測されたトークン $\hat{y}_{i-1}$ の埋め込み表現である。$\boldsymbol{h}_0$ はエンコーダの隠れ状態最後の要素 $\boldsymbol{h}_N$ を使用して初期化される。また、y_0 には文頭を表すトークンである**<bos>**を使用する。

次にデコーダは隠れ状態 $\boldsymbol{s}_j$ を目的言語側の語彙 $\mathcal{V}_d$ のサイズに写像し、softmax を行い、最終的な出力の分布を得る。

$$\boldsymbol{o}_j = \tanh(\boldsymbol{W}_h \boldsymbol{s}_j + \boldsymbol{W}_w \boldsymbol{e}_{\mathrm{dec}}(\hat{y}_{j-1})) \tag{3}$$

$$p(w|\hat{y}_{<j}) = \mathrm{softmax}(\boldsymbol{o}_j) \tag{4}$$

$$\hat{y}_j = \underset{w \in \mathcal{V}_d}{\mathrm{argmax}}(p(w|\hat{y}_{<j})) \tag{5}$$

$\boldsymbol{W}_h$ と $\boldsymbol{W}_w$ はパラメータである。

2.1.2　Attention 機構を用いたエンコーダ・デコーダモデル

単純な系列変換モデルは PBSMT を上回る翻訳精度を達成したが、系列長が長い入力に弱いことが指摘されている [13]。これはエンコーダの情報が $\boldsymbol{h}_N$ のみを通してデコーダに渡されているためである。この問題に対処するため、Bahdanau ら [2] は Attention 機構を用いたエンコーダ・デコーダモデルを提案した。このエンコー

ダ・デコーダモデルではまず、両方向の RNN を用いて入力文を符号化する。復号では、各位置のトークンを予測する際に、Attention 機構を用いてエンコーダの隠れ状態から文脈ベクトルを計算し、出力の計算に用いる。また、Bahdanau らのモデルでは LSTM ではなく、gated recurrent unit（GRU）[13] と呼ばれる RNN を使用する。

■エンコーダ　両方向エンコーダでは、順方向に加え逆方向の RNN を用いて 2 つの隠れ状態 $\overrightarrow{\boldsymbol{h}_i}$ および $\overleftarrow{\boldsymbol{h}_i}$ を計算する。これによりモデルは、left-to-right な言語モデルに加え、right-to-left な言語モデルを学習することになり、単方向の言語モデルでは捉えられない文の特徴を学習することができる。最終的な隠れ状態は、各位置の順方向および逆方向の隠れ状態ベクトルを結合することで得られる。

$$\overrightarrow{\boldsymbol{h}_i} = \overrightarrow{\mathrm{GRU}}(\overrightarrow{\boldsymbol{h}_{i-1}}, \boldsymbol{e}_{enc}(x_i)) \tag{6}$$
$$\overleftarrow{\boldsymbol{h}_i} = \overleftarrow{\mathrm{GRU}}(\overleftarrow{\boldsymbol{h}_{i+1}}, \boldsymbol{e}_{enc}(x_i)) \tag{7}$$
$$\boldsymbol{h}_i = [\overrightarrow{\boldsymbol{h}_i}; \overleftarrow{\boldsymbol{h}_i}] \tag{8}$$

$i \in [1, N]$ は入力文の位置である。$\overrightarrow{\mathrm{GRU}}$ および $\overleftarrow{\mathrm{GRU}}$ はそれぞれ順方向と逆方向の計算に用いる GRU である。$\boldsymbol{e}_{enc}(x_i)$ はトークン x_i の埋め込み表現である。

■デコーダ　各位置 $j \in [1, M]$ の単語を推論するとき、デコーダはまず、仮の隠れ状態 $\boldsymbol{s}_j$ を計算する。

$$\boldsymbol{s}_j = \mathrm{GRU}(\hat{\boldsymbol{s}}_{j-1}, \boldsymbol{e}_{dec}(\hat{y}_{j-1})) \tag{9}$$

$\hat{\boldsymbol{s}}_{j-1}$ は $j-1$ 番目の隠れ状態である。$\boldsymbol{e}_{dec}(\hat{y}_{j-1})$ は $j-1$ 番目の出力トークン $\hat{y}_{j-1}$ の埋め込み表現である。一般に $\hat{\boldsymbol{s}}_0$ は全ての要素の値がゼロであるベクトル、$\hat{y}_0$ には文頭を表すトークンである **<bos>** を使用する。

文脈ベクトル $\boldsymbol{c}_j$ は仮の隠れ状態 $\boldsymbol{s}_j$ を使用し、Attention 機構と呼ばれる手法で計算される。Attention 機構ではまず、フィードフォワード層を使用してエンコーダの隠れ状態 $\boldsymbol{h}$ の重み $\alpha_{j,i}$ を計算する。文脈ベクトルは、エンコーダの隠れ状態

の重み付き和で表される。

$$z_{j,i} = \boldsymbol{v}_t \tanh(\boldsymbol{U}_\alpha \boldsymbol{s}_j + \boldsymbol{W}_\alpha \boldsymbol{h}_i) \tag{10}$$

$$\alpha_{j,i} = \frac{\exp(z_{j,i})}{\sum_{k=1}^{N} \exp(z_{j,k})} \tag{11}$$

$$\boldsymbol{c}_j = \sum_{i=1}^{N} \alpha_{j,i} \boldsymbol{h}_i \tag{12}$$

$\boldsymbol{U}_\alpha$ および $\boldsymbol{W}_\alpha$ はモデルパラメータである。

最終的な隠れ状態 $\hat{\boldsymbol{s}}_j$ は仮の隠れ状態 $\boldsymbol{s}_j$、文脈ベクトル $\boldsymbol{c}_j$ から計算される。

$$\boldsymbol{z}_j = \sigma_z(\boldsymbol{W}_z[\boldsymbol{c}_j; \boldsymbol{s}_j]) \tag{13}$$

$$\boldsymbol{r}_j = \sigma_r(\boldsymbol{W}_r[\boldsymbol{c}_j; \boldsymbol{s}_j]) \tag{14}$$

$$\boldsymbol{s}'_j = \tanh(\boldsymbol{W}_s \boldsymbol{c}_j + \boldsymbol{r}_j \odot (\boldsymbol{U} \boldsymbol{s}_j)) \tag{15}$$

$$\hat{\boldsymbol{s}}_j = (1 - \boldsymbol{z}_j) \odot \boldsymbol{s}'_j + \boldsymbol{z}_j \odot \boldsymbol{s}_j \tag{16}$$

σ_z と σ_r はシグモイド関数を活性化関数とするフィードフォワード層である。$\boldsymbol{W}_z$、$\boldsymbol{W}_r$、$\boldsymbol{W}_s$ および $\boldsymbol{U}$ はモデルパラメータである。

システム出力の分布は隠れ状態 $\hat{\boldsymbol{s}}_j$、前のシステム出力 $\hat{y}_{j-1}$、および文脈ベクトル $\boldsymbol{c}_j^t$ から計算される。

$$\boldsymbol{o}_j = \tanh(\boldsymbol{L}_s \hat{\boldsymbol{s}}_j + \boldsymbol{L}_w \boldsymbol{e}_{dec}(\hat{y}_{j-1}) + \boldsymbol{L}_c \boldsymbol{c}_j) \tag{17}$$

$$p(w|\hat{y}_{<j}) = \mathrm{softmax}(\boldsymbol{o}_j) \tag{18}$$

$$\hat{y}_j = \underset{w \in \mathcal{V}_d}{\mathrm{argmax}}(p(w|\hat{y}_{<j})) \tag{19}$$

$\boldsymbol{L}_s$、$\boldsymbol{L}_w$ および $\boldsymbol{L}_c$ はモデルパラメータである。

2.2 モデルの訓練

ニューラル機械翻訳のモデルは参照訳の出力尤度が最大になるように学習される。学習で最もよく使われる損失関数は cross entropy 損失関数で、正解ラベルの負の対数尤度として表される。

$$J_{\mathrm{CE}} = -\sum_{j=1}^{M} \log(p(y_j|\hat{y}_{<j})) \tag{20}$$

■Teacher forcing　ニューラル機械翻訳モデルの学習を安定させる手法の 1 つとして teacher forcing がある。この手法では訓練のとき、モデル出力 $\hat{y}_{<j}$ の代わりに、参照訳のトークン列 $y_{<j}$ を使用して出力の分布を計算する。

$$J'_{\mathrm{CE}} = -\sum_{j=1}^{M} \log(p(y_j|y_{<j})) \tag{21}$$

この手法でモデルの訓練が安定する一方、推論時とは異なる入力を使用する。Scheduled sampling [14] は訓練の進みに応じて徐々にモデル出力を使うようにしていくことでこの問題に対処した。

2.3　推論

訓練済みの機械翻訳モデルを使用し推論を行うとき、常に最尤のトークンを出力する貪欲法、または、k 個の候補を保持しつつ最終的に最尤であるトークン列を探索するビーム探索のいずれかを使用する。ビーム探索では、まず、1 つ前の位置で得られた k 個の仮説に対して、それぞれの仮説で次のトークンを出力したときの尤度を計算する。次に、これらの仮説のうち、尤度が上位 k 個の仮説だけを残す。この繰り返しを全ての仮説で **<eos>** が出力されるまで繰り返す。最終的には、保持されている仮説のうち、最尤であるものを出力とする。

2.4　出力の評価

機械翻訳モデルの出力の評価には人手による評価（または主観評価）と自動評価がある。人手による評価は出力の品質について、人間が判断する評価方法であり、正しくモデルの翻訳性能を評価できることが期待される反面、時間とコストが掛かるため、モデルの開発時には使えない。一方で自動評価は人手によって作成された参照訳と出力の合致を計算することて、モデルの出力を評価する。代表的な自動評価の指標として BLEU [15] と METEOR [16] がある。

■BLEU　BLEU はドキュメントレベルの評価指標であり、モデル出力と参照訳の n-gram 適合率を計算する。一般に用いられる BLEU-4 では 1-gram から 4-gram

までの適合率を計算し、それらの幾何平均を評価指標として使用する。また、BLEUは適合率ベースの評価指標であるため、短い出力に対して不当に高いスコアをつける傾向にある。そのため、出力と参照文の長さ比を最大スコアとする簡潔ペナルティーが適用される。

■METEOR　METEOR はモデル出力と参照文の対応関係を計算し、その unigram の適合率と再現率の調和平均として計算される。単語の出現位置を無視し単語の合致率のみを考慮する BLEU と異なり、METEOR は文レベルの評価に適しており、言い換えや類義語を考慮したスコアを計算することができる。その反面、METEOR では表層ではない意味的な対応関係を考慮するために、WordNet に代表されるような語彙の関係性（例えば、類義語や対義語）をデータベース化したモジュールを利用する必要があり、全ての言語で使用できるわけではない。

第 3 章　単語分散表現

単語分散表現とは、単語（もしくはサブワード）を数百次元のベクトル空間に埋め込んだものである。分散表現を用いることで、単語の構成性や単語間の類似度や関係性を表現することができる。

単語分散表現は大きく 2 つの種類に分けられる。単語が使われている文脈によらず常に同じ分散表現を返す静的単語分散表現と、同じ単語でも文脈に応じて異なる分散表現を返す動的単語分散表現がある。前者の代表的なものに word2vec [17]、GloVe [18]、および FastText [19] がある。一方、ELMo [20] や BERT [21] は動的単語分散表現の代表的な手法である。

本研究では静的な単語分散表現に着目し、マルチモーダル機械翻訳への影響を調べる。本章では、静的な単語分散表現の代表的なものである word2vec、GloVe、および FastText について説明する。また、いずれの手法を用いて事前学習された単語分散表現からバイアスを取り除く手法について紹介する。

3.1　モデル

3.1.1　word2vec

Mikolov ら [17] はニューラル言語モデルである対数双線形モデルに基づいて単語分散表現を学習する手法を提案した。実際、word2vec には Continuous bag of words（CBoW）と skip-gram と呼ばれる 2 つの手法がある。CBoW では前後 k 単語を入力として、その間の単語を予測するように訓練する。一方で、skip-gram では 1 つの単語を入力として、その周辺の単語を予測するように訓練する。

3.1.2　GloVe

GloVe [18] も対数双線形モデルの 1 つである。word2vec と違い、GloVe ではコーパスに含まれる単語間の共起情報を使い、context window を用いて単語分散表現を学習する。非ゼロ要素のみで学習を行うことで、大規模なコーパスから効果

的にモデルを学習することができる。単語類似度タスクや固有表現認識タスクで word2vec や他の対数双線形モデルを上回る性能を達成した。

3.1.3 FastText

FastText [19] も対数双線形モデルの 1 つで、skip-gram を発展させたものである。その特徴は、単語分散表現を計算するときに、トークン自体だけではなく、そのサブワードの情報も使うところである。FastText では単語分散表現はトークンとそれを構成できるサブワードの和で表現される。サブワードを用いるという性質のため、訓練コーパスに存在しない単語に対しても単語分散表現を計算することができる。

3.2 ハブネス問題

高次元の単語分散表現に対して、内積による距離関数を使い 2 つの単語間の類似度を計算する問題では、特定の単語が他の単語の k-近傍に頻出する [22, 23]。これは機械学習においてハブネス問題と呼ばれている。この現象は事前学習された単語分散表現の効用を減少させる。特に NMT では、低頻度語は高頻度語に比べ翻訳性能が低いことが知られており [9]、これはハブネス問題によるものと考えられる。

この問題を解決するために、localized centering [10] や All-but-the-Top [11]、autoencoder を利用した方法 [24] が提案された。

3.2.1 Localized centering

Localized centering [10] はまず各単語の局所的なバイアスに基づいて、単語分散表現を移動させる。具体的には、各単語 x の局所的な中央を計算し、元の単語分散表現 $\boldsymbol{e}(x)$ から差し引くことで新しい単語分散表現 $\hat{\boldsymbol{e}}(x)$ が得られる。

$$\boldsymbol{c}_k(x) = \frac{1}{k} \sum_{x' \in k\mathrm{NN}(x)} \boldsymbol{e}(x') \tag{1}$$

$$\hat{\boldsymbol{e}}(x) = \boldsymbol{e}(x) - \boldsymbol{c}_k(x) \tag{2}$$

k は local segment size で、ハイパーパラメータである。$k\mathrm{NN}(x)$ は単語 x の k 近傍を返す。

3.2.2 All-but-the-Top

All-but-the-Top（AbtT）[11] は全語彙の全域的なバイアスを使用し、各単語の分散表現を移動させる。AbtT のアルゴリズムは 3 つの手順から成る。

1. 各単語の分散表現から全語彙の単語分散表現の平均を差し引く
2. 差し引いたあとのベクトル空間の主成分を計算する
3. 各単語の差し引いたあとの分散表現から上位 D 個の主成分を取り除く

$$\boldsymbol{e}'(x) = \boldsymbol{e}(x) - \frac{1}{|\mathcal{V}|}\sum_{w\in\mathcal{V}}\boldsymbol{e}(w) \tag{3}$$

$$\boldsymbol{u}_1, \cdots, \boldsymbol{u}_D = \mathrm{PCA}(\boldsymbol{e}', w \in \mathcal{V}) \tag{4}$$

$$\hat{\boldsymbol{e}}(x) = \boldsymbol{e}'(x) - \sum_{i=1}^{D}(\boldsymbol{u}_i^{\mathsf{T}}\boldsymbol{e}'(x))\boldsymbol{u}_i \tag{5}$$

3.2.3 Autoencoder

Kaneko ら [24] は autoencoder（AE）を用いて、事前学習した単語分散表現からバイアスを取り除く手法を提案した。AbtT に比べ AE ではバッチサイズ程度の計算を行うだけで、同等かそれ以上の性能を達成することができる。AE ではまず、単語 $\mathbf{X}$ を中心をゼロにした単語分散表現に埋め込む。その後、デコーダで元の単語分散表現に近づけるように、以下の損失関数を最小化するように訓練する。

$$J = \sum_{w\in\mathcal{V}} ||\mathbf{W}_d F(\mathbf{W}_e\boldsymbol{e}(w) + \boldsymbol{b}_e) + \boldsymbol{b}_d||^2 \tag{6}$$

$\mathbf{W}_e$ と $\boldsymbol{b}_e$ はエンコーダ（埋め込み）、$\mathbf{W}_d$ と $\boldsymbol{b}_d$ はデコーダのパラメータである。F は element-wise 活性化関数である。

第 4 章　マルチモーダル機械翻訳

マルチモーダル機械翻訳（MMT）は視覚情報などの非言語的情報を用いて入力文を目的言語へ翻訳する技術である。近年、Encoder–Decoder モデルや Attention 機構の登場により、NMT モデルは従来の統計的機械翻訳を上回る性能を達成している [2]。マルチモーダルニューラル機械翻訳も多くはこれらの NMT モデルを基に研究が行われてきた。

多くの MMT の研究には Multi30K データセット [25] が用いられる。このデータセットは画像とその説明のデータセットである Flickr30K [26] を拡張して作られ、現在はドイツ語・フランス語・チェコ語の翻訳を含む。また、日本語のデータセットには Flickr30kEnt-JP [27] がある。このデータセットは Flickr30k Entity [28] のキャプションに日本語翻訳をつけたものであるが、最も特徴的なところは、日本語翻訳だけではなく、画像と日本語翻訳のフレーズレベルの対応を含むところである。

本章ではまず、MMT で使用されるデータセットについて説明する。次に、MMT モデル、MMT 特有の評価手法について紹介し、最後に現在提案されている MMT モデルの性能を評価する。

4.1　データセット

4.1.1　Multi30K

Multi30K [25] は現在、マルチモーダル機械翻訳の研究で一般的に用いられているデータセットである。このデータセットは画像とその説明のデータセットである Flickr30K [26] を拡張して作成されている。Flickr30K には 1 画像につき 5 つの説明がついており、Multi30K ではそのうち 1 つを選び、職業翻訳者がドイツ語訳を付けている [25]。このとき、翻訳者は画像を使わずに英語説明をドイツ語に翻訳している。

Frank ら [29] は職業翻訳者を使い、Multi30K の評価およびテストデータに含まれるドイツ語翻訳を画像を見ながらポストエディットした。ポストエディットは必

	文数	% ポストエディット
評価データ	1,014	6.1
テストデータ	1,000	13.8

表 4.1: Multi30K の評価およびテストデータに含まれるポストエディットされたドイツ語翻訳の割合 [29]

要なときのみに行われ、単語のみを修正の対象としており、嗜好やスタイルに起因する修正を避けた。表 4.1 は各データに含まれるポストエディットされた翻訳の割合である。ポストエディットされた割合はテストデータが評価データよりも多くなっているが、この理由はテストデータに含まれる英語説明により多くの間違いや不正確な表現があったからだと考えられている [29]。

現在、Multi30K を拡張する形で複数のデータが公開されている。フランス語データ [30] やチェコ語データ [31]、多様性を高めたテストセット [30]、単語の曖昧性が高いテストセット [30] が利用可能である。

本論文では、主に英独翻訳・英仏翻訳・英チェコ翻訳で Multi30K を使用する。テストデータには WMT16 で公開されたもの（test_2016_flickr）および WMT17 で公開されたもの（test_2017_flickr) を使用する。

4.1.2 Flickr30kEnt-JP

Flickr30kEnt-JP [27] は Flickr30k Entity [28] を拡張したものである。このデータセットでは、各画像についている 5 つの英語説明のすべてを日本語に翻訳している。また、画像内の物体と日本語翻訳のフレーズの対応を含んでいる。

本論文では、英日翻訳で Flickr30kEnt-JP を使用する。訓練・評価・テストデータへの分割は Multi30K の分割に揃えた。また、データ量を揃えるため、1 つの画像に付き、1 つの英語説明・日本語翻訳のみを使用した。

4.2 画像特徴量

画像特徴量は事前学習された画像分類モデルを使用して抽出する。ImageNet [32] を用いて訓練された ResNet-50 [33] はよく利用される識別モデルの 1 つである。MMT モデルは ResNet-50 から抽出できる全域特徴量 [4, 34] や局所特徴量 [5, 3] を用いる。全域特徴量は画像を 1 つのベクトルで表したもので、ResNet-50 では出力層の入力に使われる隠れ状態を用いる。局所特徴量は画像を複数のベクトルの系列で表したもので、ResNet-50 の中間層の隠れ状態を用いる。

また、近年は画像検出モデルの利用が盛んである。Faster R-CNN [35] は画像から物体の特性を示すオブジェクトレベル特徴量である Region of Interest（RoI）を抽出することができ、RoI 特徴量は様々な MMT モデル [36, 37] に組み込まれている。

本論文では ResNet-50 を用いて抽出した全域特徴量 $\boldsymbol{v}_g$ と局所特徴量 $\boldsymbol{v}_r$ を利用する。オブジェクトレベル特徴量（例えば RoI）を利用するモデルの評価は今後の課題とする。

4.3 モデル

マルチモーダルニューラル機械翻訳モデルでは、 N 個のトークンで構成される入力言語の文 $\mathbf{x} = \{x_1, x_2, \cdots, x_N\}$ と画像から抽出した特徴量 $\boldsymbol{v}_g$ および $\boldsymbol{v}_r$ を使用し、M 個のトークンで構成される目的言語の文 $\mathbf{y} = \{y_1, y_2, \cdots, y_M\}$ へ翻訳するように学習される。

4.3.1 Decoder initialization

Caglayan ら [4] は画像の全域特徴量 $\boldsymbol{v}_g$ を使用してデコーダの隠れ状態を初期化する Decoder Initialization（dec–init）を提案した。この手法では画像の全域特徴量をデコーダの次元に写像し、デコーダの隠れ状態の $\boldsymbol{s}_0$ として使用する。このモデルの他の部分は Bahdanau ら [2] と同じである。

$$\boldsymbol{s}_0 = \tanh(\boldsymbol{W}_{\mathrm{v}\to\mathrm{d}}\boldsymbol{v}_g) \tag{1}$$

$\boldsymbol{W}_{\mathrm{v}\to\mathrm{d}}$ はパラメータで、画像特徴量を MMT モデルの次元に写像する。

4.3.2 IMAGINATION

IMAGINATION [34] はマルチタスク学習を用いたモデルである。このモデルでは機械翻訳と潜在共有空間構成の 2 つのタスクを学習する。後者では、文と画像を同一の潜在共有空間にマッピングするとき、ある文とそれに対応する画像の距離が近くになるように学習する。2 つのタスク間でエンコーダを共有する。

■アーキテクチャ　共有エンコーダおよび機械翻訳デコーダは Bahdanau ら [2] と同じものを使用する。

潜在共有空間構成タスクのデコーダでは、まずエンコーダの隠れ状態 $\boldsymbol{h}$ の平均ベクトルを計算し、潜在共有空間へ写像し、最終的なベクトル $\hat{\boldsymbol{v}}$ を計算する。

$$\hat{\boldsymbol{v}} = \tanh(\boldsymbol{W}_{\mathrm{v}} \cdot \frac{1}{N}\sum_{i}^{N} \boldsymbol{h_i}) \tag{2}$$

$\boldsymbol{W}_{\mathrm{v}}$ はモデルパラメータである。

■損失関数　IMAGINATION の損失関数は 2 つのタスクの線形補間で与えられる。

$$J = \lambda J_{\mathrm{T}}(\theta, \phi_{\mathrm{T}}) + (1-\lambda) J_{\mathrm{V}}(\theta, \phi_{\mathrm{V}}) \tag{3}$$

θ は共有エンコーダのパラメータ、ϕ_{T} と ϕ_{V} はそれぞれ機械翻訳モデルおよび潜在共有空間モデルのパラメータである。λ は線形補間の係数である*。

機械翻訳モデルの損失関数 $J_{\mathrm{T}}(\theta, \phi_{\mathrm{T}})$ には式 21 の cross entropy 損失関数を使用する。

$$J_{\mathrm{T}}(\theta, \phi_{\mathrm{T}}) = -\sum_{j=1}^{M} \log(p(y_j | \hat{y}_{<j})) \tag{4}$$

*我々の実験では $\lambda = 0.5$ を用いた。

潜在共有空間モデルの損失関数 $J_{\mathrm{V}}(\theta, \phi_{\mathrm{V}})$ には max margin 損失関数を使用する。これによりモデルは対応する入力文と画像を近づけるように学習する。

$$J_{\mathrm{V}}(\theta, \phi_{\mathrm{V}}) = \sum_{\boldsymbol{v}'_g \neq \boldsymbol{v}_g} \max\{0, \alpha - d(\hat{\boldsymbol{v}}, \boldsymbol{v}_g) + d(\hat{\boldsymbol{v}}, \boldsymbol{v}'_g)\} \tag{5}$$

$\boldsymbol{v}'$ は対応していないバッチ内の他の画像の特徴量である。d はコサイン類似度で文と画像の類似度を計算する。α はマージンで潜在共有空間のばらつきを調整する変数である。[†]。

4.3.3 Doubly-Attentive NMT

Doubly-attentive NMT [5]（DA-NMT）は Bahdanau ら [2] を基に、画像への Attention 機構を追加した MMT モデルである。エンコーダと画像の両方にそれぞれ別の Attention 機構を用いて文脈ベクトルを計算し、それら 2 つの文脈ベクトルを用いて最終的な文脈ベクトルを計算する。

■アーキテクチャ　エンコーダは Bahdanau ら [2] と同じものを使用する。

デコーダでは、まず Bahdanau ら [2] と同様に仮の隠れ状態 $\boldsymbol{s}$ を計算する（式 9）。

入力文の文脈ベクトルと画像の文脈ベクトルは 2 つの独立した Attention 機構を使用して計算される。位置 j の符号化のとき、入力文の文脈ベクトル $\boldsymbol{c}_j^t$ は式 10 を使用して計算される。

同様に、画像の文脈ベクトルは画像の局所特徴量 $\boldsymbol{v}_r$ を使い、文の文脈ベクトルを計算した方法と同様にして計算される。また、ゲート機構を使用して画像の文脈ベクトルの大きさを制御する。ゲート機構では前の隠れ状態 $\hat{s}_{j-1}$ を使用してスケール変数 β_j を計算し、画像の文脈ベクトルへの注意の大きさを決定する。

$$z_{j,i}^v = \boldsymbol{v}_v \tanh(\boldsymbol{U}_\alpha^v \boldsymbol{s}_j + \boldsymbol{W}_\alpha^v \boldsymbol{v}_{r,i}) \tag{6}$$

$$\alpha_{j,i}^v = \frac{\exp(z_{j,i}^v)}{\sum_{k=1}^N \exp(z_{j,k}^v)} \tag{7}$$

$$\beta_j = \sigma(\boldsymbol{W}_s \hat{\boldsymbol{s}}_{j-1} + \boldsymbol{b}_s) \tag{8}$$

[†]我々の実験では $\alpha = 0.1$ を用いた。

$$\boldsymbol{c}_j^v = \beta_j \sum_{i=1}^{N} \alpha_{j,i}^v \boldsymbol{v}_{r,i} \tag{9}$$

$\boldsymbol{v}_v$、$\boldsymbol{U}_\alpha^v$、$\boldsymbol{W}_\alpha^v$、$\boldsymbol{W}_s$ および $\boldsymbol{b}_s$ はモデルのパラメータである。

最終的な隠れ状態 $\hat{\boldsymbol{s}}_j$ 仮の隠れ状態 $\boldsymbol{s}_j$、文の文脈ベクトル $\boldsymbol{c}_j^t$ および画像の文脈ベクトル $\boldsymbol{c}_j^v$ から計算される。

$$\boldsymbol{z}_j = \sigma_z(\boldsymbol{W}_z^t \boldsymbol{c}_j^t + \boldsymbol{W}_z^v \boldsymbol{c}_j^v + \boldsymbol{W}_z \hat{\boldsymbol{s}}_j) \tag{10}$$
$$\boldsymbol{r}_j = \sigma_r(\boldsymbol{W}_r^t \boldsymbol{c}_j^t + \boldsymbol{W}_r^v \boldsymbol{c}_j^v + \boldsymbol{W}_r \hat{\boldsymbol{s}}_j) \tag{11}$$
$$\boldsymbol{s}'_j = \tanh(\boldsymbol{W}_z^t \boldsymbol{c}_j^t + \boldsymbol{W}_z^v \boldsymbol{c}_j^v + \boldsymbol{r}_j \odot (\boldsymbol{U} \hat{\boldsymbol{s}}_j)) \tag{12}$$
$$\hat{\boldsymbol{s}}_j = (1 - \boldsymbol{z}_j) \odot \boldsymbol{s}'_j + \boldsymbol{z}_j \odot \boldsymbol{s}_j \tag{13}$$

σ_z と σ_r はシグモイド関数を活性化関数とするフィードフォワード層である。$\boldsymbol{W}_z^t$、$\boldsymbol{W}_z^v$、$\boldsymbol{W}_z$、$\boldsymbol{W}_r^t$、$\boldsymbol{W}_r^v$、$\boldsymbol{W}_r$、$\boldsymbol{W}_z^t$、$\boldsymbol{W}_z^v$ および $\boldsymbol{U}$ はモデルパラメータである。

システム出力は隠れ状態 $\hat{\boldsymbol{s}}_j$、前のシステム出力 $\hat{y}_{j-1}$、文の文脈ベクトル $\boldsymbol{c}_j^t$、画像の文脈ベクトル $\boldsymbol{c}_j^v$ から計算される。

$$\boldsymbol{o}_j = \tanh(\boldsymbol{L}^s \hat{\boldsymbol{s}}_j + \boldsymbol{L}^w \boldsymbol{e}_{dec}(\hat{y}_{j-1}) + \boldsymbol{L}^t \boldsymbol{c}_j^t + \boldsymbol{L}^i \boldsymbol{c}_j^v) \tag{14}$$
$$p(w|\hat{y}_{<j}) = \mathrm{softmax}(\boldsymbol{o}_j) \tag{15}$$
$$\hat{y}_j = \underset{w \in \mathcal{V}}{\mathrm{argmax}} \{p(w|\hat{y}_{<j})\} \tag{16}$$

$\boldsymbol{L}^s$、$\boldsymbol{L}^w$、$\boldsymbol{L}^t$ および $\boldsymbol{L}^i$ はモデルパラメータである。

■損失関数　損失関数には cross entropy 損失関数（式 21）を使用する。

$$J = -\sum_{j=1}^{M} \log(p(y_j|\hat{y}_{<j})) \tag{17}$$

4.3.4 Visual Attention Grounding NMT

Visual Attention Grounding NMT（VAG-NMT）[3] は画像の特徴量をエンコーダ・デコーダに組み込むのと同時に、マルチタスク学習も使用する。補助タスクとして潜在共有空間の学習を行い、エンコーダを共有する。

■アーキテクチャ　VAG-NMT の共有エンコーダは Bahadanau ら [2] を拡張したものである。モデルは文の分散表現 $\boldsymbol{t}$ をエンコーダの隠れ状態 $\boldsymbol{h}$ と画像の局所特徴量 $\boldsymbol{v}$ を使い計算する。

$$z_i = \tanh(\boldsymbol{W}_v \boldsymbol{v}_r) \cdot \tanh(\boldsymbol{W}_h \boldsymbol{h}_i) \tag{18}$$

$$\beta_i = \frac{\exp(z_i)}{\sum_{k=1}^{N} \exp(z_k)} \tag{19}$$

$$\boldsymbol{t} = \sum_{i=1}^{N} \beta_i \boldsymbol{h}_i \tag{20}$$

$\boldsymbol{W}_v$ と $\boldsymbol{W}_h$ はモデルパラメータである。

MT デコーダの構造は Bahadanau ら [2] と同様であるが、文の分散表現 $\boldsymbol{t}$ を使用してデコーダの隠れ状態を初期化する。

$$\boldsymbol{h}_0^{\mathrm{dec}} = \tanh(\boldsymbol{W}_{init}(\frac{1}{2}\boldsymbol{t} + \frac{1}{2N}\sum_{i}^{N} \boldsymbol{h}_i)) \tag{21}$$

$\boldsymbol{W}_{init}$ はモデルパラメータである。

潜在共有空間では、文の分散表現 $\boldsymbol{t}$ と画像の全域特徴量 $\boldsymbol{v}_g$ が近くなるように写像される。

$$\hat{\boldsymbol{t}} = \tanh(\boldsymbol{W}_t \boldsymbol{t} + \boldsymbol{b}_t) \tag{22}$$

$$\hat{\boldsymbol{v}} = \tanh(\boldsymbol{W}_v \boldsymbol{v}_g + \boldsymbol{b}_v) \tag{23}$$

$\boldsymbol{W}_t$、$\boldsymbol{b}_t$、$\boldsymbol{W}_v$ および $\boldsymbol{b}_v$ はモデルパラメータである。

■損失関数　VAG-NET の損失関数は 2 つのタスクの損失関数の線形補間（式 3）で与えられる。MT タスクの損失関数 $J_{\mathrm{T}}(\theta, \phi_{\mathrm{T}})$ は cross entropy 損失関数（式 21）で与えられる。

潜在共有空間タスクの損失関数 $J_{\mathrm{V}}(\theta, \phi_{\mathrm{V}})$ は negative sampling を使用する max margin 損失関数で与えられる。

$$\begin{aligned} & J_{\mathrm{V}}(\theta, \phi_{\mathrm{V}}) \\ & = \sum_{k} \max_{p \neq k} \{0, \gamma - d(\hat{\boldsymbol{t}}_k, \hat{\boldsymbol{v}}_k) + d(\hat{\boldsymbol{t}}_k, \hat{\boldsymbol{v}}_p)\} \\ & + \sum_{p} \max_{k \neq p} \{0, \gamma - d(\hat{\boldsymbol{t}}_p, \hat{\boldsymbol{v}}_p) + d(\hat{\boldsymbol{t}}_k, \hat{\boldsymbol{v}}_p)\} \end{aligned} \tag{24}$$

d はコサイン類似度、k と p はそれぞれ文と画像の番号である。$\boldsymbol{t}_{k\neq p}$ は negative sample で同じバッチ内の他のデータから選択する。γ はマージンで潜在共有空間のばらつきを調整する変数である[‡]。

4.4 モデルの評価手法

機械翻訳の標準的な評価指標（BLEU、METEOR）を用いた評価に加え、MMT では画像に対するモデルの挙動を評価する。

4.4.1 敵対的評価

敵対的評価（Adversarial evaluation）[38] はモデルが画像を認識しながら翻訳を行っているかどうかを評価する指標である。この評価では image awareness を以下の手順で計算する。

1. 正しく対応する文と画像をモデルに入力し、評価指標（BLEU または METEOR）を計算する（Congruent）。
2. 正しく対応してない文と画像をモデルに入力し、評価指標を計算する（Incongruent）。
3. Congruent と Incongruent の評価指標の差を計算する。

モデルが画像を認識して翻訳を行っている場合、評価指標の差は大きくなる。逆にモデルが画像を無視して翻訳を行っている場合、評価指標の差は小さくなる。Elliott ら [38] はすべての MMT モデルが必ずしも画像を認識して使用しているわけではないことを示した。

4.4.2 Input degradation

Input degradation [39] は入力文に特定の制約を適用することで、入力文が不完全なときのモデルの挙動を評価する。これまでの MMT の先行研究での画像などの

[‡]我々の実験では $\gamma = 0.1$ を用いた。

MMT モデル	En–De		En–Fr		En–Cs		En–Ja
	BLEU	Meteor	BLEU	Meteor	BLEU	Meteor	BLUE
NMT	40.39	57.82	60.13	74.86	**31.28**	**30.82**	**38.96**
dec–init	**40.70**	**57.91**	59.87	**74.88**	31.01	30.49	38.86
IMAG+	40.43	57.73	**60.20**	74.83	31.07	30.60	38.71
DA-NMT	39.70	57.30	59.73	74.60	30.75	30.52	38.47
VAG-NMT	39.52	56.97	59.37	74.30	30.88	30.13	38.20

表 4.2: 各 MMT モデルの評価データにおける性能

補助的な情報の効果は限定的であることが多い。これは、機械翻訳においては、目的言語の翻訳を生成するときに入力文だけで十分であることが多いからであり、入力文が完全な状況下では MMT モデルの reasoning 性能を評価することが難しい。Caglayan ら [39] は入力文を 3 つの制約（名詞マスク、色マスク、長さマスク）を適用することで画像の認識が要求される状況下での MMT モデルの挙動を評価した。

4.5 まとめ

本章ではマルチモーダルニューラル機械翻訳で使用されるデータセット、モデル、および評価手法について概説した。

予備実験として、表 4.2 に本章で紹介した MMT モデルを、BLEU および Meteor を使用して評価データで評価したときの性能を示した。English–German 翻訳においては、dec–init モデルが最も翻訳性能が良かったが、他の言語対においては、MMT と文のみを使用する NMT では顕著な差は見られなかった。

第 5 章　大規模単言語コーパスを利用したサブワード分割

サブワード分割は文中の単語を更に小さい単位であるサブワードに分割する手法であり、NMT モデルの翻訳性能を向上させることが知られている [40, 41]。統計的機械翻訳に比べ語彙数に制限がある NMT モデルにおいて、サブワード分割を利用することにより、未知語や低頻度語を効果的に NMT モデルで扱うことができるようになる。

本章ではサブワード分割で最もよく使用される BytePair Encoding（BPE）[40] を使用する。しかし、低資源領域において、BPE の設定は NMT モデルの翻訳性能に大きく影響することが知られている [42]。これは、低資源領域においては、適切なサブワードを学習できないことに起因すると考えられる。例えば、“universal” という単語は本来 “uni+versal” と分割されることが期待されるが、低資源である場合、“uni” というサブワードの出現が少ないため、より頻出である “un“を使用してサブワード分割が行われ、“un+iversal” となり、適切にサブワード分割できない。この問題を軽減させるため、本章では、大規模な単言語コーパスを利用して学習したサブワードを利用することで、より適切なサブワード分割が行える手法を提案する。

5.1　大規模単言語コーパスから学習したサブワードの適用

本章では、大規模な単言語コーパスから事前学習したサブワードを用いて、対訳コーパスをサブワード分割する。その後、分割された対訳コーパスを用いて MMT モデルを訓練する。

■サブワードの学習　提案手法ではまず、Sennrich ら [40] の BPE を用いて、大規模な単言語コーパスからサブワードを学習する。対訳コーパスに対して BPE を行う場合、原言語および目的言語の両方に適用することができるサブワード（joint BPE）を学習することが一般的であるが、単言語コーパスからサブワードを学習するときは単言語のみに適用できるサブワードを学習する。

言語	行数	トークン数	タイプ数
英語	45.5M	2.9B	7.9M
ドイツ語	18.6M	936M	8.5M
チェコ語	4.3M	144M	2.8M
日本語	10.1M	602M	2.6M

表 5.1: 各言語の Wikipedia コーパスに含まれるデータ行数とトークン数

■サブワードの適用　MMT モデルの訓練に使用するマルチモーダル対訳コーパスは、言語ごとに学習されたサブワードを使用してサブワード分割する。実験では、原言語文のみをサブワード分割する場合、目的言語文のみをサブワード分割する場合、原言語文および目的言語文をサブワード分割する場合の 3 つの分割方法について検討する。MMT モデルはサブワード分割されたデータを使用して訓練・評価・テストする。

5.2　実験設定

5.2.1　サブワードの学習

BPE に使用するサブワードを学習する大規模な単言語コーパスには Wikipedia コーパス*を使用する。具体的には、WikiExtractor†を用いて Wikipedia のタイトルと本文を抽出したのち、Moses スクリプト‡を使い lower-case、tokenize、および記号の normalize の前処理を行った。日本語データについては MeCab（辞書には IPADIC を使用）を使用して単語分割を行った後、Moses スクリプトを使用して同様の前処理を行った。表 5.1 は Wikipedia コーパスの統計情報である。

サブワードの学習に使用するマージ数は 30,000 である。また、サブワードは言

*https://dumps.wikimedia.org/。英語・ドイツ語・日本語は 2020 年 7 月 20 日、チェコ語は 2020 年 12 月 20 日のものを使用した。
†https://github.com/attardi/wikiextractor
‡https://github.com/moses-smt/mosesdecoder

語ごとに学習を行い、2 つ以上の言語で使用できる joint BPE は学習しない。

5.2.2 データセット

実験では Multi30K データセットを使用して MT モデルを訓練・評価・テストを行った。評価指標には BLEU を使用した。評価する言語対は入力言語に英語、出力言語にドイツ語・チェコ語・日本語とした。

画像の特徴量の抽出には Multi30k で提供されているスクリプト feature-extractor を使用した[§]。これにより、事前学習された ResNet-50 [33] により画像がエンコードされたのち、ネットワークの pool5 層にある隠れ状態（2048 次元）が特徴量として抽出される。

5.2.3 モデル

本章では画像を使用しない NMT と、decoder initialization（dec–init）、IMAGINATION（IMAG+）、doubly-attentive NMT（DA-NMT）、および VAG-NMT の 4 つの MMT モデルに対して、提案手法を適用した。また、ベースラインとして、サブワードを使用しないモデルと、Multi30K で学習したサブワードを用いてサブワード分割した対訳コーパスを使い訓練したモデルを用いる。Multi30K でサブワードを学習するとき、マージ数は 10,000 を使用した。

エンコーダの埋め込み層は 300 次元で、隠れ層は双方向 GRU で 640 次元である。MT システムのデコーダの埋め込み層は 300 次元で、隠れ層は 320 次元である。また、マルチタスク学習を用いる MMT モデルの潜在共有空間は 2048 次元である。モデルの訓練には Adam を使用し、初期学習率は 0.0004 である。ドロップアウトの確率は 0.4 にした.

入力文・参照文は Moses を使い lower-case、tokenize、および normalize を行った。語彙サイズに上限は設けず、全ての単語を語彙として使用する。

[§]https://github.com/multi30k/dataset

		Model	W–W	W–B_m	B_m–W	B_m–B_m	W–B_w	B_w–W	B_w–B_w
英語 → ドイツ語	test2016	NMT	38.53	38.94	37.97	39.03	38.78	38.49	**39.33**
		dec–init	38.38	38.93	38.21	38.64	39.12	37.92	**39.73**
		IMAG+	38.09	39.11	38.20	39.06	39.09	38.32	**39.22**
		DA-NMT	37.79	38.17	37.88	37.93	38.23	37.52	**38.78**
		VAG-NMT	37.80	38.80	37.35	38.07	**39.06**	37.73	38.60
	test2017	NMT	31.37	**32.36**	30.38	31.67	32.23	31.45	32.21
		dec–init	31.28	31.59	30.57	31.58	**31.97**	31.54	32.25
		IMAG+	31.16	**32.75**	30.92	32.14	32.06	31.03	32.40
		DA-NMT	30.81	**32.12**	30.55	31.10	31.71	30.64	31.81
		VAG-NMT	30.41	31.56	29.83	31.48	31.15	30.26	**31.88**
英語 → チェコ語	test2016	NMT	32.31	32.28	32.23	**32.76**	32.41	32.02	31.65
		dec–init	31.91	32.17	31.58	32.09	32.01	**32.43**	31.86
		IMAG+	**33.17**	32.58	31.63	32.52	31.91	31.84	32.62
		DA-NMT	31.71	**32.04**	31.35	31.94	31.79	31.45	31.29
		VAG-NMT	31.34	**31.83**	30.39	31.46	31.60	31.73	31.30
英語 → 日本語	test2016	NMT	38.26	38.54	38.32	38.21	**38.85**	38.68	38.79
		dec–init	38.72	38.77	38.09	38.72	**38.86**	38.40	38.67
		IMAG+	**38.80**	38.54	38.31	38.37	38.50	38.26	38.78
		DA-NMT	38.11	38.18	38.03	37.33	37.95	38.15	**38.55**
		VAG-NMT	37.94	**38.26**	37.49	37.74	38.15	37.94	38.10
Average			35.31	**35.43**	34.63	35.09	**35.40**	34.91	**35.94**

表 5.2: サブワード分割別の MMT モデルの性能。“NMT” は画像を用いない NMT モデルを表す。各列はそれぞれ入力側の分割方法と出力側の分割方法を示し、“W” はサブワード分割無し、B_m は `Multi30K` を使用したサブワード分割、B_w は `Wikipedia` を使用したサブワード分割を表す。

言語対		分割なし	Multi30K	Wikipedia
英語 → ドイツ語	原言語	9,796	5,155	8,480
	目的言語	18,048	7,083	10,010
英語 → チェコ語	原言語	9,796	4,448	8,480
	目的言語	22,239	7,184	11,443
英語 → 日本語	原言語	9,796	5,021	8,480
	目的言語	12,817	7,487	10,280

表 5.3: サブワード分割別の語彙のサイズ。

5.2.4 結果

表 5.2 は Multi30K データセットで実験を 3 回行った結果で、test2016 と test2017 での BLEU の結果を示す。単語レベルの MMT モデル（W–W）に比べ、目的言語側に Multi30K で学習した BPE を適用した場合（W–B_m）、目的言語側に Wikipedia で学習した BPE を適用した場合（W–B_w）、入力言語・目的言語側に Wikipedia で学習した BPE を適用した場合（B_w–B_w）で多くの MMT モデルの翻訳性能が向上する（それぞれ +0.12 BLEU、+0.09 BLEU、+0.63 BLEU）ことを確認した。

5.3 考察

実験の結果、Wikipedia コーパスで学習したサブワードを用いた BPE が、MMT モデルの性能を向上させることがわかった。

目的言語側のサブワード分割に比べ、原言語側へのサブワード分割は効果が低い。これは、原言語である英語の語彙は Multi30K の翻訳を行うには十分であり、サブワード分割を適用することにより過剰に単語の分割が行われためだと考えられる。表 5.3 はそれぞれのサブワード分割手法を適用した場合のモデルの語彙数である。Wikipedia のサブワードを使用したモデル（B_w–*）は Multi30K を使用したモデル（B_m–*）に比べ、多くの単語レベルの語彙を残しており、出力言語のサブ

サブワード分割	日本語参照文
なし	ボストン テリア が 、 白い 柵 の 前 で 、...
Multi30K	ボ@@ スト@@ ン テリア が 、 白い 柵 の 前 で 、...
Wikipedia	ボストン テリア が 、 白い 柵 の 前 で 、...
なし	ウィンタージャケット を 着 て ヘルメット を ...
Multi30K	ウィ@@ ン@@ ター@@ ジャケット を 着 て ヘルメット を ...
Wikipedia	ウィンター@@ ジャケット を 着 て ヘルメット を ...

表 5.4: 異なるコーパスで学習したサブワードを用いたサブワード分割の例。“@@”を含むトークンはサブワード分割された単語である。

ワード分割方法を問わず、平均的な性能も向上している（+0.57 BLEU）。

また、大規模な単言語コーパスを使用することで適切な単語分割が行われることを確認するため、Multi30K を使用した場合と Wikipedia を使用した場合でサブワード分割が異なる文を表 5.4 に示した。この例のように、Wikipedia を用いたサブワード分割は、Multi30K を用いたサブワード分割よりも、適切に分割できることがわかる。

5.4 まとめ

本章では、大規模な単言語コーパスからサブワードを学習し、低資源なマルチモーダル対訳コーパスを適切にサブワード分割する手法を提案した。実験を通して、既存の MMT モデルに対してサブワード分割を行う場合、Wikipedia コーパスから学習したサブワードを使用することが、翻訳性能を向上させることを示した。

第 6 章　バイアスを消去された単語分散表現を利用するマルチモーダル機械翻訳モデル

事前学習された単語分散表現は多くの自然言語処理において、ニューラルモデルの重要な要素だと考えられている。NMT においては、事前学習された単語分散表現は低資源領域において有用であることが示されている [43]。その中で、エンコーダ及びデコーダの重みを FastText [19] で事前学習された単語分散表現で初期化している。著者らは低資源な言語対においてモデルの性能が向上することを示した。

しかし、ハブネス問題（バイアス、3.2）は事前学習された単語分散表現の効用を低下させることが知られている [9]。Rios Gonzales ら [9] は人手で意味や語義のラベルをアノテーションすることでハブネス問題の影響を軽減させたが、時間やコストが掛かる。この研究では、他の自然言語処理タスクで利用されている 3 つのバイアス消去手法（3.2）を MMT に適用し、その性能を評価した。翻訳実験を通して、事前学習された単語分散表現が MMT でも有効であることを示した。また、バイアス消去を行うことで更に翻訳性能を向上させることを示した。

6.1　事前学習した単語分散表現を利用した MMT モデル

事前学習した単語分散表現は MMT モデルの単語埋め込み層の初期化に使用する。初期化された単語埋め込み層の重みは、MMT モデルの訓練を通して更新する。

■未知語の単語分散表現　NMT における未知語には 2 種類ある。1 つは単語分散表現の訓練コーパスには含まれるが、MT モデルの語彙に含まれない単語（Out-of-vocabulary: OOV）であり、一般に NMT モデルでは特殊トークン（例えば、`unk`）で表される単語である。他方は MT モデルの語彙には含まれるが、単語分散表現の訓練コーパスに含まれない単語（Out-of-embedding: OOE）である。FastText は OOE 単語に対して文字 n-gram を使用して分散表現を計算するため、OOE は word2vec 及び GloVe のみで発生する。OOV および OOE 単語の分散表現には、語彙に含まれないが、事前学習された単語分散表現に含まれる単語の分散表現の平均を使用する。これらの単語分散表現も他の単語同様、MT モデルの訓練を通して

更新される。

■サブワード分割　また、本章ではサブワード分割を利用する MMT モデルに、事前学習した単語分散表現を組み込み手法を提案する。単語分散表現は単語ごとに学習されるものとして提案されたが、サブワードに対しても学習することができる。例えば、FastText は単語分散表現を計算するとき、単語の分散表現に加えて、その単語を構成する分散表現も考慮している。5 章で示したとおり、目的言語側をサブワード分割することが MMT モデルの性能を向上させることが分かっており、大規模単言語コーパスからサブワードの分散表現を学習することで、性能を更に向上させることが期待できる。サブワード分割された単語分散表現は以下の手順で学習を行う。

- 単言語コーパスからサブワードを学習する。
- 学習したサブワードを使用して単言語コーパスをサブワード分割する。
- サブワード分割された単言語コーパスを使用して、単語分散表現を学習する。

得られた単語分散表現は MMT モデルの単語埋め込み層の初期化に使用する。なお、本章は対訳コーパスを使用するサブワード分割は検討しない。これは、一般に単語分散表現は汎用性を高めるために、最終的に適用するタスクに依存しない形で作成されるためである。

6.2　実験

6.2.1　単語分散表現

本章では word2vec、GloVe、および FastText で訓練した単語分散表現の効用を評価する。既に公開されている事前学習された単語分散表現はいずれも使用できる言語が限定的か、学習に使用したコーパスが異なっている。そこで、本章では `Wikipedia` コーパス*を使用して、単語分散表現の訓練を行った。具体的には、

*https://dumps.wikimedia.org/。英語・ドイツ語・日本語は 2020 年 7 月 20 日、チェコ語は 2020 年 12 月 20 日のものを使用した。

言語	行数	サブワード分割なし トークン数	タイプ数	サブワード分割あり トークン数	タイプ数
英語	45.5M	2,590M	7.9M	2,917M	57.0K
ドイツ語	18.6M	936M	8.5M	1,173M	44.7K
チェコ語	4.3M	144M	2.8M	191M	39.0K
日本語	10.1M	602M	2.6M	656M	59.5K

表 6.1: 各言語の `Wikipedia` コーパスに含まれるデータ行数とトークン数。

`WikiExtractor`[†]を用いて `Wikipedia` のタイトルと本文を抽出したのち、Moses スクリプト[‡]を使い lower-case、tokenize、および記号の normalize の前処理を行った。日本語データについては MeCab（辞書には IPADIC を使用）を使用して単語分割を行った後、Moses スクリプトを使用して同様の前処理を行った。単語分散表現の訓練に使用するハイパーパラメータは先行研究と同様のものを使用しており、すべての言語で同一のものを使用する。サブワード分割には Byte Pair Encoding（BPE）[40] を使用し、マージ数は 30,000 とした。表 6.1 に前処理済み Wikipedia コーパスの統計情報を示す。

全ての単語分散表現は 300 次元で訓練した。訓練に使用したパラメータは以下の通りで、ここにないパラメータは既定値を使用した。word2vec は CBoW を使用して、窓枠 10、負例 10、最小出現回数 10、イテレーション 3 で訓練した。GloVe は窓枠 10、最小出現回数 10 で訓練した。FastText は 5-gram までのサブワード、窓枠 5、負例 10 で訓練した。

バイアスの消去には Localized Centering（LC）[§]、All-but-the-Top（AbtT）[¶]、および Autoencoder（AE）を使用する手法を使用する。

[†]https://github.com/attardi/wikiextractor
[‡]https://github.com/moses-smt/mosesdecoder
[§]実験では $k = 10$ を使用した
[¶]実験では $D = 3$ を使用した

言語	BPE	タイプ数	トークン数	OOV	OOE
英語	なし	9,796	380,214	7,898K	79
	あり	8,480	403,575	48K	0
ドイツ語	なし	18,048	365,536	8,448K	1,196
	あり	10,010	438,681	35K	0
チェコ語	なし	22,239	297,896	2,745K	1,417
	あり	11,443	399,214	28K	1
日本語	なし	12,817	597,449	2,623K	496
	あり	10,280	635,378	49K	0

表 6.2: Multi30K の訓練データ（各言語 29,000 文）に含まれる単語のタイプ数、トークン数、および Wikipedia コーパスを使用した場合の OOV および OOE のタイプ数。

6.2.2 データセット

MMT モデルの訓練・評価・テストには Multi30K を使用した。評価する言語対は、入力言語に英語、出力言語にドイツ語・チェコ語・日本語とした。表 6.2 は Multi30K の語彙についての統計情報である。

画像の特徴量の抽出には Multi30k データセットで提供されているスクリプト `feature-extractor` を使用した[‖]。これにより、事前学習された ResNet-50 [33] により画像がエンコードされたのち、ネットワークの pool5 層にある隠れ状態（2048 次元）が特徴量として抽出される。

6.2.3 モデル

本章では画像を使用しない NMT と、decoder initialization (dec–init)、IMAGINATION (IMAG+)、doubly-attentive NMT (DA-NMT)、および VAG-NMT の

[‖] https://github.com/multi30k/dataset

分割	初期化	NMT	dec–init	IMAG+	DA-NMT	VAG-NMT
W–W	Random	38.53	38.38	38.09	37.79	37.80
	word2vec	**38.79**	38.48	**38.19**	38.48	**38.20**
	GloVe	38.12	**38.61**	37.75	36.51	37.26
	FastText	35.53	19.85	26.86	32.89	37.88
W–B_w	Random	38.78	39.12	39.09	38.07	39.06
	word2vec	39.14	**39.35**	**39.71**	**39.29**	**39.49**
	GloVe	**39.48**	38.87	39.39	38.06	38.45
	FastText	8.36	7.60	6.22	23.70	38.75
B_w–B_w	Random	39.33	39.73	39.22	38.78	38.60
	word2vec	**39.98**	**39.93**	39.16	**39.42**	**39.13**
	GloVe	39.45	38.60	**39.35**	38.53	38.74
	FastText	6.73	6.07	9.61	36.94	38.69

表 6.3: 単語分散表現で初期化したモデルの `test2016` データセットに対する結果 (英語 → ドイツ語)。

4 つの MMT モデルに対して提案手法を適用した。モデルの評価指標には BLEU [15] を用いた。

エンコーダの埋め込み層は 300 次元で、隠れ層は双方向 GRU で 640 次元である。MT システムのデコーダの埋め込み層は 300 次元で、隠れ層は 320 次元である。また、マルチタスク学習を用いる MMT モデルの潜在共有空間は 2048 次元である。モデルの訓練には Adam を使用し、初期学習率は 0.0004 である。ドロップアウトの確率は 0.4 にした.

分割	初期化	NMT	dec–init	IMAG+	DA-NMT	VAG-NMT
W–W	Random	32.31	**31.91**	**33.17**	**31.99**	31.34
	word2vec	31.03	31.31	30.97	31.16	32.11
	GloVe	**32.68**	31.89	32.19	31.75	31.12
	FastTest	14.92	32.17	12.56	29.84	**31.22**
W–B_w	Random	32.41	32.01	31.91	31.79	31.60
	word2vec	31.79	32.06	30.80	31.50	**32.14**
	GloVe	**32.60**	**32.91**	**32.46**	**32.95**	32.08
	FastText	6.31	7.76	5.47	8.25	31.66
B_w–B_w	Random	31.65	31.86	32.62	31.29	31.30
	word2vec	31.50	32.04	32.01	31.31	31.14
	GloVe	**32.81**	**32.76**	**32.78**	**32.74**	**32.07**
	FastTest	9.53	8.71	6.15	5.31	31.12

表 6.4: 単語分散表現で初期化したモデルの `test2016` データセットに対する結果（英語 → チェコ語）。

6.3 結果

6.3.1 事前学習した単語分散表現

表 6.3 は英語–ドイツ語翻訳における、ランダムに単語埋め込み層を初期化したモデル（Random）および事前学習した単語分散表現を使って初期化したモデルの性能である。ランダムで初期化する場合に比べると、いずれの MMT モデルにおいても、事前学習した単語分散表現を用いて初期化したほうが、モデルの翻訳性能が向上することがわかった。特に、word2vec を用いて学習した単語分散表現はほとんどのモデルで性能改善に役立つ。一方で FastText は MMT モデルの性能を大きく低下させることが分かった。

一方で、英チェコ翻訳（表 6.4）では GloVe で事前学習した単語分散表現を用いるモデルが最も良い性能を示しており、word2vec を用いるモデルは多くの場合で

分割	初期化	NMT	dec–init	IMAG+	DA-NMT	VAG-NMT
W–W	Random	38.26	**38.72**	**38.80**	38.11	37.94
	word2vec	38.43	38.58	38.77	**38.12**	38.14
	GloVe	**38.81**	38.44	38.18	37.52	**38.34**
	fastText	38.19	38.52	37.08	36.45	37.37
W–B_w	Random	**38.85**	**38.86**	38.50	37.95	38.15
	word2vec	38.60	37.97	**38.83**	**38.12**	38.19
	GloVe	38.36	38.52	38.63	37.34	**38.64**
	FastTest	38.50	39.09	38.05	37.03	37.68
B_w–B_w	Random	38.79	38.67	**38.78**	**38.55**	38.10
	word2vec	38.59	**38.99**	38.65	38.33	38.31
	GloVe	**38.84**	38.34	38.08	37.45	**38.74**
	FastText	37.74	38.00	32.39	36.50	36.82

表 6.5: 単語分散表現で初期化したモデルの `test2016` データセットに対する結果（英語 → 日本語）。

性能を落としている。また、英日翻訳（表 6.5）ではこれまでの 2 つの言語対に比べ、ランダムで初期化したモデルが最も良い性能を示すことが多いことが分かった。

6.3.2 バイアスを消去した単語分散表現

表 6.6 は NMT、decoder initialization、および IMAGINATION について、バイアスを消去した単語分散表現を使いモデルを初期化したときの性能である。Doubly-attentive NMT および VAG-NMT は他のモデルに比べ性能が低いため、除外した。

ほとんどのモデルにおいて、バイアスを消去した単語分散表現を用いることでモデルの性能が向上することが分かった。特に、半数以上のモデルでは GloVe の単語分散表現を用いることで、最も良い性能のモデルを訓練することができる。例えば、

	初期化	NMT			dec–init			IMAGINATION		
		W–W	W–B_w	B_w–B_w	W–W	W–B_w	B_w–B_w	W–W	W–B_w	B_w–B_w
英語 → ドイツ語	Random	38.53	38.78	39.33	38.38	39.12	39.73	38.09	39.09	39.22
	word2vec	38.79	39.14	39.98	38.48	39.35	39.93	38.19	39.71	39.16
	w/ LC	37.68	39.68	39.24	38.64	38.64	39.89	38.46	38.71	39.75
	w/ AbtT	38.55	**39.96**	39.47	38.56	**40.05**	39.96	38.72	39.56	39.02
	w/ AE	38.51	39.05	39.46	38.58	39.88	**40.25**	38.94	**39.94**	38.94
	GloVe	38.12	39.48	39.45	38.61	38.87	38.60	37.75	39.39	39.35
	w/ LC	38.14	39.21	39.61	38.21	39.36	39.32	38.45	39.45	39.64
	w/ AbtT	**38.84**	39.80	**40.21**	**38.65**	39.50	39.45	**39.57**	38.32	**40.25**
	w/ AE	38.55	39.13	39.94	38.27	39.46	39.28	38.06	39.73	39.49
英語 → チェコ語	Random	32.31	32.41	31.65	31.91	32.01	31.86	**33.17**	31.91	32.62
	word2vec	31.03	31.79	31.50	31.31	32.06	32.04	30.97	30.80	32.01
	w/ LC	32.66	31.97	32.30	32.47	**32.99**	32.31	32.34	31.87	32.83
	w/ AbtT	32.22	32.76	32.88	32.42	32.64	31.78	32.64	32.80	32.39
	w/ AE	32.49	32.23	33.18	31.77	32.98	32.75	31.56	32.46	32.15
	GloVe	**32.68**	32.60	32.81	31.89	32.91	32.76	32.19	32.46	32.78
	w/ LC	31.99	32.64	32.43	32.18	32.87	32.60	31.86	**33.08**	32.36
	w/ AbtT	32.40	32.60	32.77	**33.05**	32.66	**33.16**	33.16	32.37	32.79
	w/ AE	32.37	**33.73**	**33.38**	32.67	32.04	32.34	32.86	32.84	**33.04**
英語 → 日本語	Random	38.26	38.85	38.79	38.72	38.86	38.67	38.80	38.50	38.78
	word2vec	38.43	38.60	38.59	38.58	37.97	38.99	38.77	38.83	38.65
	w/ LC	38.90	38.93	38.38	38.29	38.67	38.56	**39.17**	39.20	38.71
	w/ AbtT	38.48	**39.23**	**39.18**	38.73	39.07	38.92	38.98	38.84	38.53
	w/ AE	38.04	38.65	38.61	38.90	**39.18**	38.88	38.46	38.50	38.55
	GloVe	38.81	38.36	38.84	38.44	38.52	38.34	38.18	38.63	38.08
	w/ LC	38.45	38.27	38.44	38.44	38.79	39.02	38.79	38.83	38.87
	w/ AbtT	**39.14**	38.96	38.88	38.59	39.04	38.26	38.89	39.12	**38.99**
	w/ AE	38.40	38.76	38.87	**38.94**	38.74	**39.06**	38.74	**39.22**	38.83

表 6.6: バイアスを消去した単語分散表現で初期化したモデルの `test2016` データセットに対する結果。“w/ LC”、“w/ AbtT”、および “w/ AE” はそれぞれ、LC、AbtT、AE を使用してバイアスを消去したときの結果を示す。

英語–ドイツ語翻訳においては、IMAGINATION モデルで Wikipedia で学習したサブワードで BPE を適用し、GloVe で学習した単語分散表現を使い単語埋め込み層を初期化したとき、モデルの性能が向上し（+0.13 BLEU）、All-but-the-Top でバイアスを消去することで更に性能を向上させる（+1.03 BLEU）。

6.4 議論

6.4.1 単語分散表現

本章では、word2vec や GloVe で事前学習した単語分散表現が全ての言語対において、NMT モデルの性能を向上させることが分かった。一方で、FastText は性能の向上に寄与しなかった。FastText を使用した単語分散表現の学習では、単語に加えサブワードの分散表現も学習するため、単語の分散表現のみを学習する word2vec および GloVe の場合に比べ、より多くのデータが必要である。Qi ら [43] は Wikipedia コーパスと Common Crawl コーパスを使い学習した FastText の単語分散表現を用いて NMT モデルの性能を向上させたが、単語分散表現の訓練で使用されたコーパスに含まれる単語は本章で使用した単語のおよそ 50 倍である。

6.4.2 バイアス消去

All-but-the-Top および Autoencoder を使用する手法は多くのモデルの性能を向上させることが分かった。バイアスを消去しない場合、事前学習した単語分散表現を用いたモデルが、ランダムに初期化されたモデルよりも、性能で劣る場合が見受けられる。特に英日翻訳では半数以上のモデルでランダムに初期化したほうが良い結果となった。しかし、バイアスの消去を行うことで、すべてのモデルにおいて、事前学習した単語分散表現を用いるモデルのほうが良い性能を達成することができた。このことから、事前学習した単語分散表現を用いるときは、バイアスを消去することが好ましいと言える。

一方で、Localized Centering はモデルの性能向上に限定的にしか貢献しなかった。Localized Centering では、元の単語分散表現から近傍単語の平均を差し引くため、バイアスを消去したあとの単語分散表現の大きさは小さくなる。Hara ら [10]

はバイアス消去後の単語分散表現をそのまま使用したのに対して、MT モデルでは追加の訓練を行うため、その過程で単語間の関係が失われ、性能が向上しなかったと考えられる。

6.5 まとめ

本章では大規模単言語コーパスで事前学習した単語分散表現を MMT モデルで使用する手法を提案した。実験を通して、word2vec や GloVe で学習した単語分散表現がモデルの性能を向上させることが分かった。また、All-but-the-Top や Autoencoder を使用してバイアスを消去することで更に改善させることができることを示した。

第 7 章　単語分散表現を予測するマルチモーダル機械翻訳モデル

マルチモーダル機械翻訳に利用できるデータセットは小さいため、外部リソースを利用する研究も行われている。このアプローチでは、画像を含まない対訳コーパス [34, 7] や逆翻訳で作成した疑似コーパス [44] を追加のリソースとして利用する手法が提案されている。

リソースの乏しい言語対でのニューラル機械翻訳において、事前学習された単語分散表現をエンコーダに組み込むことで性能が向上するが、デコーダに組み込んでも性能が向上しないことが確認されている [43]。Kumar ら [45] は、事前学習した単語分散表現をデコーダで予測する手法を提案し、従来のニューラル機械翻訳と同等以上の性能を達成し低頻度語の翻訳精度を向上させており、単語分散表現をより効果的に利用しているといえる。

本章では、単語分散表現を積極的に活用する Kumar らの手法をマルチモーダル機械翻訳に導入し、事前学習された単語分散表現の有効性を示した。

7.1　単語分散表現の予測によるマルチモーダル機械翻訳

本章では、マルチモーダル機械翻訳に Kumar ら [45] の手法を導入し、事前学習された単語分散表現を組み込む。単語分散表現の効果を確認しやすくするため、ベースとなるモデルには、シンプルなマルチタスク学習型のモデルである IMAGINATION [34] を使用する（4.3.2 を参照）。

機械翻訳の基本的な構造は、Bahdanau ら [2] と同じであるが、デコーダの出力層で単語の生成確率ではなく、単語分散表現を予測し、事前学習された単語分散表現から最も近い単語をシステム出力とする点が異なる [45]。

$$\hat{\boldsymbol{e}}_{\boldsymbol{j}} = \tanh(\boldsymbol{W}_{\mathrm{o}}\boldsymbol{s}_{\boldsymbol{j}} + \boldsymbol{b}_{\mathrm{o}}) \tag{1}$$

$$\hat{y}_j = \underset{w \in \mathcal{V}}{\operatorname{argmin}} \{d(\hat{\boldsymbol{e}}_j, \mathrm{e}(w))\} \tag{2}$$

$\boldsymbol{s}_{\boldsymbol{j}}$、$\hat{\boldsymbol{e}}_{\boldsymbol{j}}$、$\hat{y}_j$ はそれぞれ各タイムステップ j におけるデコーダの隠れ状態、単語分散表現予測、システム出力で、$\boldsymbol{W}_{\mathrm{o}}$ および $\boldsymbol{b}_{\mathrm{o}}$ は出力層のパラメータである。また、$\mathcal{V}$

は出力言語側の語彙集合、w は出力言語の語彙集合に含まれる単語、$\mathrm{e}(w_k)$ は w_k に対応する事前学習された単語分散表現、$d(\boldsymbol{e_a}, \boldsymbol{e_b})$ は 2 つの単語分散表現 $\boldsymbol{e_a}$ と $\boldsymbol{e_b}$ の間の距離を表し、本章では距離関数にコサイン類似度を使用する。

機械翻訳の損失関数には Lazaridou ら [46] が提案する Margin-based Ranking Loss を使用する。

$$J_T(\theta, \phi_{\mathrm{T}}) = \sum_{j}^{M} \max\{0, \gamma + d(\hat{\boldsymbol{e}}_j, \mathrm{e}(w_j^-)) - d(\hat{\boldsymbol{e}}_j, \mathrm{e}(y_j))\} \tag{3}$$

$$w_j^- = \underset{w \in \mathcal{V}}{\mathrm{argmax}}\{d(\hat{\boldsymbol{e}}_{\boldsymbol{j}}, \mathrm{e}(w)) - d(\hat{\boldsymbol{e}}_{\boldsymbol{j}}, \mathrm{e}(y_j)) \tag{4}$$

M は出力文の長さ、γ はマージンである*。w_j^- は負例であり、予測した単語分散表現と近く、正解の単語分散表現から遠いものが 1 つ選ばれる。

事前学習された単語分散表現はエンコーダ埋め込み層とデコーダ埋め込み層の初期化、およびデコーダの出力層に使用する。エンコーダの埋め込み層は初期化ののち、学習データを使用して追加の学習を行うが、デコーダの埋め込み層と出力層はパラメータを固定し、学習を行わない。

7.2 実験

7.2.1 データセット

実験では Multi30k データセットを使用して学習、検証、評価した。評価する言語対は、入力言語にフランス語、出力言語に英語とした。評価指標には BLEU と METEOR を使用した。

画像の特徴量の抽出には Multi30k で提供されているスクリプト feature-extractor を使用した†。これにより、事前学習された ResNet-50 [33] により画像がエンコードされたのち、ネットワークの pool5 層にある隠れ状態（2048 次元）が特徴量として抽出される。

*実験では $\gamma = 0.5$ を使用した

†https://github.com/multi30k/dataset

7.2.2 モデル

共有エンコーダの入力層は 128 次元で、隠れ層は双方向 GRU で 256 次元である。機械翻訳システムのデコーダの入力および出力層で使用する単語分散表現は 300 次元で、隠れ層は 256 次元である。また、潜在共有空間は 2048 次元である。

入力文と参照文は Multi30k データセットのスクリプト task1-tokenize.sh を使用し前処理を行った。語彙サイズはすべての入力言語および出力言語で 10,000 とした。

損失関数の補間係数には $\lambda = 0.5$ を使用した。最適化手法には Adam を使用し、学習率は 0.0004 である。勾配は 1.0 でクリッピングし、ドロップアウト率は 0.3 に設定した。

7.2.3 単語分散表現

モデルに使用する単語分散表現の事前学習には FastText を使用する。Wikipedia および Common Crawl から学習した単語分散表現はオンラインで公開されている[‡][47]。これらの単語分散表現は skip-gram [17] を使って学習されており、次元は 300 である。

未知語の単語分散表現には、事前学習に使用したコーパスに含まれるがモデルの学習データに含まれない単語分散表現の平均を使用する。また、事前学習された単語分散表現は All-but-the-Top を使用して前処理する。

7.3 結果

表 7.1 は Multi30k データセットで実験を 3 回行った結果である。dev、test にそれぞれ検証と評価データセットでの結果を示す。NMT は画像を使わず単語の生成確率を予測する機械翻訳システム、IMAG+ は IMAGINATION [34] を再実装したマルチモーダル機械翻訳システムの結果である。提案手法は NMT と IMAGINATION にくらべ、BLEU スコアでそれぞれ +1.90 と +1.72 の改善を確

[‡]https://fasttext.cc/

	dev	test	
Model	BLEU	BLEU	METEOR
NMT	50.83	51.00±.37	42.65±.12
IMAG+	51.03	51.18±.16	42.80±.19
Proposed	52.20	52.90±.07	43.70±.11

表 7.1: Multi30k の評価

認した。

7.4 考察

事前学習された単語分散表現が、機械翻訳システムへ与える影響を確認するため、表 7.2 では、埋め込み層をランダムで初期化した場合、および、デコーダの埋め込み層を固定しない場合の結果を示した。random は一様分布、fasttext は FastText で事前学習した単語分散表現で初期化した。Fixed はデコーダ埋め込み層のマッピング行列を固定するかどうかを表す。

デコーダ埋め込み層を固定しない場合、IMAGINATION と同等以下の性能となる。このことから、単語分散表現の予測に基づくマルチモーダル機械翻訳システムにおいては、デコーダの埋め込み層を事前学習された単語分散表現に固定することが重要である。

また、すべての埋め込み層を一様分布で初期化する場合と比べ、デコーダ側を FastText で初期化した場合は BLEU スコアで +1.84 の改善が見られ、エンコーダ側を FastText で初期化した場合の改善（+0.55 ポイント）と比べ、大きな改善が確認できる。これは、マルチタスク学習を行うことにより、同じ学習データを使用したとしても、エンコーダ側はデコーダ側に比べ、十分に学習することができ、事前学習された単語分散表現の効果が低下するためである。

次に、事前学習された単語分散表現が、システム出力の傾向に与える影響を確認するため、図 7.1 では、単語の出現頻度ごとの F 値を示した。ここでいう F 値と

エンコーダ	デコーダ	Fixed	BLEU	METEOR
fasttext	fasttext	Yes	52.90	43.70
random	fasttext	Yes	52.06	43.23
fasttext	random	No	50.77	42.44
random	random	No	50.22	41.97
fasttext	fasttext	No	50.29	42.25
random	fasttext	No	49.69	41.68

表 7.2: 埋め込み層別の単語分散表現の効果

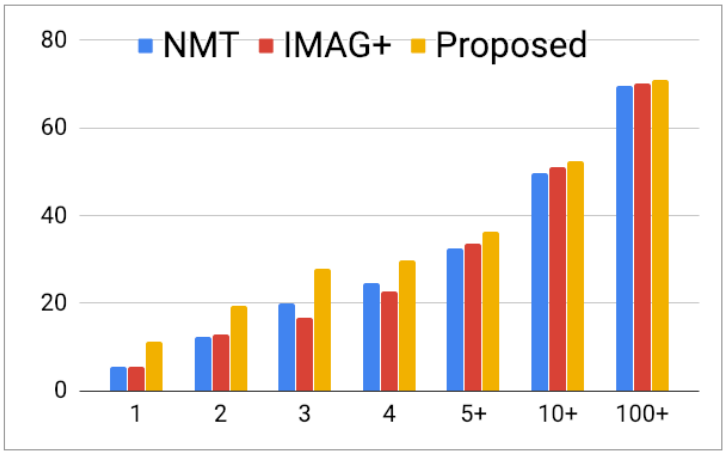

図 7.1: 単語の出現頻度ごとの F 値

は、単語が出現した参照文のインデックスを正解ラベル、単語が出現したシステム出力のインデックスを予測として計算したものである。提案手法はベースラインと比較し、低頻度語で顕著な改善が見られた一方、高頻度語になると改善の効果が限定的であることが分かった。

7.5　まとめ

本章では、事前学習された単語分散表現を効果的に利用する手法を、マルチモーダル機械翻訳に導入し、事前学習された単語分散表現の利用がマルチモーダル機械翻訳の性能を改善することとともに、事前学習された単語分散表現がデコーダの改善に有効であることを示した。

第 8 章　おわりに

本研究では、大規模に利用可能な単言語コーパスをマルチモーダル機械翻訳モデルに組み込むための手法を調査した。特に単語分割と単語分散表現に着目し、事前学習されたサブワードの利用、事前学習された単語分散表現の利用、および単語分散表現を予測する MMT モデルを提案した。Multi30K を用いた翻訳実験を通して、複数の言語において提案手法が MMT モデルの翻訳精度を向上させることを確認した。今後は、マルチモーダル機械翻訳に最適な単語分散表現を得られる単言語コーパスの研究や画像の情報を事前学習された単語分散表現に統合することを検討していきたい。

発表リスト

国際会議

1. Zizheng Zhang, Tosho Hirasawa, Wei Houjing, Masahiro Kaneko, Mamoru Komachi. **Translation of New Named Entities from English to Chinese.** The 7th Workshop of Asian Translation. December, 2020.
2. Hiroto Tamura, Tosho Hirasawa, Masahiro Kaneko, Mamoru Komachi. **TMU Japanese-English Multimodal Machine Translation System for WAT 2020.** The 7th Workshop on Asian Translation. December, 2020.
3. Hwichan Kim, Tosho Hirasawa, Mamoru Komachi. **Korean-to-Japanese Neural Machine Translation System using Hanja Information.** The 7th Workshop on Asian Translation. December, 2020.
4. Aizha Imankulova* and Masahiro Kaneko* and Tosho Hirasawa* and Mamoru Komachi. **Towards Multimodal Simultaneous Neural Machine Translation.** The Fifth Conference on Machine Translation. November, 2020. (*Equal contribution)
5. Masashi Takaku, Tosho Hirasawa, Mamoru Komachi, Kanako Komiya. **Neural Machine Translation from Historical Japanese to Contemporary Japanese Using Diachronically Domain-Adapted Word Embeddings.** The 34th Pacific Asia Conference on Language, Information and Computation. October, 2020.

6. Masahiro Kaneko, Aizhan Imankulova, Tosho Hirasawa and Mamoru Komachi. **English-to-Japanese Diverse Translation by Combining Forward and Backward Outputs.** The Fourth Workshop on Neural Generation and Translation. July, 2020.
7. Tosho Hirasawa*, Zhishen Yang*, Mamoru Komachi, Naoaki Okazaki. **Keyframe Segmentation and Positional Encoding for Video-guided Machine Translation Challenge 2020.** The First Workshop on Advances in Language and Vision Research: Video-guided Machine Translation (VMT) Challenge. July, 2020. (*Equal contribution)
8. Hwichan Kim, Tosho Hirasawa and Mamoru Komachi. **Zero-shot North Korean to English Neural Machine Translation by Character Tokenization and Phoneme Decomposition.** The 58th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop, July, 2020.
9. Tosho Hirasawa and Mamoru Komachi. **Debiasing Word Embeddings Improves Multimodal Machine Translation.** Machine Translation Summit XVII Volume 1: Research Track. August, 2019.
10. Tosho Hirasawa, Hayahide Yamagishi, Yukio Matsumura and Mamoru Komachi. **Multimodal Machine Translation with Embedding Prediction.** The 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop. June, 2019.

国内会議

1. 金輝燦, 平澤寅庄, 小町守. 韓国語対訳データを利用した文字分割と音素分解による朝鮮語ニューラル機械翻訳. 言語処理学会第 26 回年次大会, 2020.
2. 高久雅史, 平澤寅庄, 小町守, 古宮嘉那子. 通時的な領域適応を行った単語分散表現を利用した古文から現代文へのニューラル機械翻訳. 言語処理学会第 26 回年次大会, 2020.
3. 平澤寅庄, 山岸駿秀, 松村雪桜, 小町守. 事前学習した単語分散表現を利用し

たマルチモーダル機械翻訳. 言語処理学会第 25 回年次大会. 2019.

謝辞

研究活動に際して丁寧に指導をして下さるとともに、研究する上で快適な環境を与えて下さった小町守准教授に深く感謝します。研究生活を通して、学会発表や他大学や研究機関との共同研究等、大変多くの経験をすることができました。研究生時代にメンターとして指導して下さった山岸さん、松村さんには多くのことを教えていただき、本当に感謝しています。また、金子さんや Aizhan さんは日頃から研究についての相談や議論に付き合っていただき、感謝の念に耐えません。そして、研究生活をともに過ごし様々な相談に乗って頂いた研究室の学生の皆さん、ありがとうございます。最後に、副査を引き受けて下さり多くのアドバイスを頂きました山口亨教授と高間康史教授に感謝します。

参考文献

[1] I. Sutskever, O. Vinyals, and Q.V. Le, “Sequence to sequence learning with neural networks,” Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, pp.3104–3112, 2014.

[2] D. Bahdanau, K. Cho, and Y. Bengio, “Neural machine translation by jointly learning to align and translate,” Proceedings of 3rd International Conference on Learning Representations, 2015.

[3] M. Zhou, R. Cheng, Y.J. Lee, and Z. Yu, “A visual attention grounding neural model for multimodal machine translation,” Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pp.3643–3653, 2018.

[4] O. Caglayan, W. Aransa, A. Bardet, M. García-Martínez, F. Bougares, L. Barrault, M. Masana, L. Herranz, and J. van deWeijer, “LIUM-CVC submissions for WMT17 multimodal translation task,” Proceedings of the Second Conference on Machine Translation, pp.432–439, 2017.

[5] I. Calixto, Q. Liu, and N. Campbell, “Doubly-attentive decoder for multi-modal neural machine translation,” Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, Volume 1: Long Papers, pp.1913–1924, 2017.

[6] P. Koehn and R. Knowles, “Six challenges for neural machine translation,” Proceedings of the First Workshop on Neural Machine Translation, pp.28–39, 2017.

[7] S. Grönroos, B. Huet, M. Kurimo, J. Laaksonen, B. Mérialdo, P. Pham, M. Sjöberg, U. Sulubacak, J. Tiedemann, R. Troncy, and R. Vázquez, “The memad submission to the WMT18 multimodal translation task,” Proceedings of the Third Conference on Machine Translation: Shared Task Papers, pp.603–611, 2018.

[8] R. Sennrich and B. Zhang, “Revisiting low-resource neural machine translation: A case study,” Proceedings of the 57th Conference of the Association for Computational Linguistics, Volume 1: Long Papers, pp.211–221, 2019.

[9] A. Rios Gonzales, L. Mascarell, and R. Sennrich, “Improving Word Sense Disambiguation in Neural Machine Translation with Sense Embeddings,” Proceedings of the Second Conference on Machine Translation, pp.11–19, 2017.

[10] K. Hara, I. Suzuki, M. Shimbo, K. Kobayashi, K. Fukumizu, and M. Radovanovic, “Localized centering: Reducing hubness in large-sample data,” Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, pp.2645–2651, 2015.

[11] J. Mu and P. Viswanath, “All-but-the-top: Simple and effective postprocessing for word representations,” Proceedings of 6th International Conference on Learning

Representations, 2018.

[12] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” Neural Comput., vol.9, no.8, pp.1735–1780, 1997.

[13] K. Cho, B. van Merriënboer, D. Bahdanau, and Y. Bengio, “On the Properties of Neural Machine Translation: Encoder–Decoder Approaches,” Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation, pp.103–111, Oct. 2014.

[14] S. Bengio, O. Vinyals, N. Jaitly, and N. Shazeer, “Scheduled sampling for sequence prediction with recurrent neural networks,” Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, pp.1171–1179, 2015.

[15] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, “BLEU: a method for automatic evaluation of machine translation,” Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, pp.311–318, 2002.

[16] M. Denkowski and A. Lavie, “Meteor Universal: Language specific translation evaluation for any target language,” Proceedings of the Ninth Workshop on Statistical Machine Translation, pp.376–380, 2014.

[17] T. Mikolov, I. Sutskever, K. Chen, G.S. Corrado, and J. Dean, “Distributed representations of words and phrases and their compositionality,” Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013, pp.3111–3119, 2013.

[18] J. Pennington, R. Socher, and C.D. Manning, “GloVe: Global Vectors for Word Representation,” Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, pp.1532–1543, 2014.

[19] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, “Enriching word vectors with subword information,” Trans. Assoc. Comput. Linguistics, vol.5, pp.135–146, 2017.

[20] M. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, “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), pp.2227–2237, 2018.

[21] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pp.4171–4186, 2019.

[22] G. Dinu and M. Baroni, “Improving zero-shot learning by mitigating the hubness problem,” Proceedings of 3rd International Conference on Learning Representations: Workshop Track Proceedings, 2015.

[23] M. Faruqui, Y. Tsvetkov, P. Rastogi, and C. Dyer, “Problems with evaluation of word embeddings using word similarity tasks,” Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP, pp.30–35, 2016.

[24] M. Kaneko and D. Bollegala, “Autoencoding improves pre-trained word embeddings,” Proceedings of the 28th International Conference on Computational Linguistics, pp.1699–1713, 2020.

[25] D. Elliott, S. Frank, K. Sima’an, and L. Specia, “Multi30k: Multilingual English-German image descriptions,” Proceedings of the 5th Workshop on Vision and Language, pp.70–74, 2016.

[26] P. Young, A. Lai, M. Hodosh, and J. Hockenmaier, “From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions,” Transactions of the Association for Computational Linguistics, vol.2, pp.67–78, 2014.

[27] H. Nakayama, A. Tamura, and T. Ninomiya, “A visually-grounded parallel corpus with phrase-to-region linking,” Proceedings of The 12th Language Resources and Evaluation Conference, pp.4204–4210, 2020.

[28] B.A. Plummer, L. Wang, C.M. Cervantes, J.C. Caicedo, J. Hockenmaier, and S. Lazebnik, “Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models,” Int. J. Comput. Vis., vol.123, no.1, pp.74–93, 2017.

[29] S. Frank, D. Elliott, and L. Specia, “Assessing multilingual multimodal image description: Studies of native speaker preferences and translator choices,” Natural Language Engineering, vol.24, no.3, pp.393–413, 2018.

[30] D. Elliott, S. Frank, L. Barrault, F. Bougares, and L. Specia, “Findings of the second shared task on multimodal machine translation and multilingual image description,” Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Task Papers, pp.215–233, 2017.

[31] L. Barrault, F. Bougares, L. Specia, C. Lala, D. Elliott, and S. Frank, “Findings of the third shared task on multimodal machine translation,” Proceedings of the Third Conference on Machine Translation: Shared Task Papers, pp.304–323, 2018.

[32] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and F. Li, “ImageNet: A large-scale hierarchical image database,” Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp.248–255, 2009.

[33] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, pp.770–778, 2016.

[34] D. Elliott and À. Kádár, "Imagination improves multimodal translation," Proceedings of the Eighth International Joint Conference on Natural Language Processing, Volume 1: Long Papers, pp.130–141, 2017.

[35] S. Ren, K. He, R.B. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, pp.91–99, 2015.

[36] P.-Y. Huang, F. Liu, S.-R. Shiang, J. Oh, and C. Dyer, "Attention-based multimodal neural machine translation," Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers, pp.639–645, 2016.

[37] Y. Zhao, M. Komachi, T. Kajiwara, and C. Chu, "Double attention-based multimodal neural machine translation with semantic image regions," Proceedings of the 22nd Annual Conference of the European Association for Machine Translation, pp.105–114, 2020.

[38] D. Elliott, "Adversarial evaluation of multimodal machine translation," Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pp.2974–2978, 2018.

[39] O. Caglayan, P. Madhyastha, L. Specia, and L. Barrault, "Probing the need for visual context in multimodal machine translation," Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pp.4159–4170, 2019.

[40] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, Volume 1: Long Papers, pp.1715–1725, 2016.

[41] T. Kudo, "Subword regularization: Improving neural network translation models with multiple subword candidates," Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, Volume 1: Long Papers, pp.66–75, 2018.

[42] R. Sennrich and B. Zhang, "Revisiting low-resource neural machine translation: A case study," Proceedings of the 57th Conference of the Association for Computational Linguistics, Volume 1: Long Papers, pp.211–221, 2019.

[43] Y. Qi, D.S. Sachan, M. Felix, S. Padmanabhan, and G. Neubig, "When and why are pre-trained word embeddings useful for neural machine translation?,"

Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), pp.529–535, 2018.

[44] J. Helcl, J. Libovický, and D. Varis, "CUNI system for the WMT18 multimodal translation task," Proceedings of the Third Conference on Machine Translation: Shared Task Papers, pp.616–623, 2018.

[45] S. Kumar and Y. Tsvetkov, "Von Mises-Fisher Loss for Training Sequence to Sequence Models with Continuous Outputs," Proceedings of 7th International Conference on Learning Representations, 2019.

[46] A. Lazaridou, G. Dinu, and M. Baroni, "Hubness and pollution: Delving into cross-space mapping for zero-shot learning," Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, Volume 1: Long Papers, pp.270–280, 2015.

[47] E. Grave, P. Bojanowski, P. Gupta, A. Joulin, and T. Mikolov, "Learning Word Vectors for 157 Languages," Proceedings of the Eleventh International Conference on Language Resources and Evaluation, 2018.