修　　士　　論　　文

多言語言語処理モデルの経験的分析
—単語埋め込み計算のための文脈窓および機械翻訳モデルが捉える言語類型論—

Empirical Analysis of Multilingual NLP Models: Context Window for Word Embeddings and Linguistic Typology for Machine Translation

指導教員　　鶴岡 慶雅 教授

東京大学大学院情報理工学系研究科
電子情報学専攻

氏　名　　**48-186416 李凌寒**

提　出　日　　令和二年 01 月 30 日

概 要

自然言語処理では近年ニューラルネットワークに基づくモデル（ニューラルモデル）が台頭している。ニューラルモデルを用いると、入出力のデータさえ用意すれば、従来必要とされていた素性エンジニアリングをすることなく、様々なタスクを従来のモデルを上回る水準で解くことができる。このモデル学習の簡便さや、ニューラルモデルの基礎となっている密な連続値ベクトル表現の柔軟性により、単一のモデルで異なる自然言語を複数扱える多言語自然言語処理モデルの開発が容易となり、近年盛んに研究されている。

しかし、分野の進歩が急速に進む一方で、それら新しいモデルがどのような性質を持つのか、既存の知見とどのような繋がりがあるかは十分に理解されているとは言えず、調査が必要である。本論文では、２種類の多言語自然言語処理モデルについて分析した実験結果についてまとめる。

１つは、複数言語で共通の意味空間を学習する多言語単語埋め込み（Cross-lingual Word Embedding）ついての分析である。この分野では、独立に学習された複数の単語埋め込み空間を共通の空間に写像する mapping-based の手法が主流であるが、これは複数言語の単語埋め込み空間が構造的に類似しており線形写像でマップすることができるという強い仮定に基づいている。単語埋め込み空間の構造は訓練コーパスにおける単語の共起情報に強く依存しているが、実際には文内の単語について文脈窓を定義し、その文脈窓にどのような単語が現れるかの情報によって、単語埋め込みが計算される。この文脈窓と単語埋め込み空間の構造は密接に関連しているにも関わらず、文脈窓の選択が mapping-based の手法で得られた多言語単語埋め込み空間にどのような影響を与えるかを調べた研究は少ない。本研究では、文脈窓と多言語単語埋め込みの関係について理解を深めるために、異なる文脈窓サイズの単語埋め込みを複数言語で訓練し、それらのマッピングの性能を測定する実験を行った。その結果、全体的な傾向として、窓サイズを大きくすればするほど、２つの埋め込み空間はマップしやすくなることが観察された。

もう１つは、複数言語ペア間の翻訳を１つのモデルで行う多言語機械翻訳 (Multilingual Machine Translation) モデルの分析である。今回分析の対象とするモデルは、通常の機械翻訳モデルから構造は変わらずに、入出力のみが多言語に拡張されるというシンプルなものである。シンプルながら、モデルの内部表現の性質、例えば、モデルが異なる言語間の普遍の性質を捉えているかや、各モジュール毎に入出力言語の情報をどの程度保持しているのか、などはあまり調べられてこなかった。今回の実験では、モデルの内部表現から入出力言語の言語類型論的性質を予測する probing タスクを通して、内部表現がどの程度入出力言語の性質を捉えているかを探った。その結果、モデルの内部表現はベースラインを上回る精度で言語の言語類型論的特徴、特に語順に関する特徴を予測でき、また入出力に近い層の内部表現であるほど、その入出力言語の性質を予測しやすいということが分かった。

本研究の結果は異なる言語で共通の文法を捉えることの難しさと可能性を同時に示唆するものとなった。本研究の成果は既存の多言語モデルへの理解を深め、より良いモデル開発への足掛かりとなる。

目 次

図 目 次

表 目 次

第1章　はじめに

1.1　背景

自然言語処理 (natural language processing; NLP) はインターネットが我々の生活に当たり前のものとなった今、現代社会に欠かせない重要な技術になっている。インターネット上で人々がコミュニケーションをとる主な手段はやはり自然言語であり、これを効率的に処理できることのメリットは大きい。例えば、情報検索の分野ではユーザのクエリ、これは往々にして自然言語にて記述されるが、その意味を解釈するために自然言語処理の技術が必要になる。他には、人々が交流する SNS では如何に有害な投稿やスパムを取り締まるかが重要な課題となっているが、言語処理の技術によりそれらを自動的に検知することが可能となる。

NLP の分野では、近年 1 つのモデルで複数の異なる言語を扱う**多言語自然言語処理システム**の研究が盛んに行われている。この技術は世界規模で自然言語処理システムを用いたサービスを提供する場合に非常に重要となる。通常であれば、世界各地の異なる言語を話す人々に向けてサービスを提供する際は、語彙や文法が異なる個別の言語毎に異なるシステムを開発して提供する。しかし、世界の言語は総計 7,000 個あるとされ [1]、話者 100 万人を超える言語だけでも約 250 存在する[1]。この数のシステムを個別に開発、維持・管理するのは非常にコストが大きい。そこで、これらの言語をひとまとめに扱う多言語自然言語処理システムが開発できれば、関わるコストを大幅に削減できるのである。

多言語自然言語処理システム自体は以前から研究されていたが [2]、実用が現実味を帯びてきたのはニューラルネットワークに負うところが大きい。まず、ニューラルネットワークは密な連続値ベクトルを内部表現として持ち、全ての処理は連続値ベクトルと行列の演算によって行われる。この性質により、異なる言語のシステムを扱う際にパラメータを共通のものにしたり、異なるモデル同士の表現やパラメータの間に何かしらの制約を与えるなど、柔軟なモデル構築が出来るようになり、多言語自然言語処理モデルの構築が容易になった。また、ニューラルネットワークに基づくモデルは end-to-end で学習することが出来る。つまり、入出力のデータを用意しさえすれば、特徴量エンジニアリングをすることなく、モデルを学習することができる。従来の機械学習で多言語を扱うとなると、複数言語を入力するために言語に共通する特徴や個別の特徴を考慮するなど煩雑な処理が必要となってしまう。一方、ニューラルネットワークでは end-to-end で学習することが出来るため、入出力データを多言語に拡張するだけで、多言語モデルを学習できてしまう [3, 4, 5]。

しかし、分野の進歩が急速に進む一方で、それら新しいモデルがどのような性質を持つのか、既存の知見とどのような繋がりがあるかは、まだ十分に調べられているとは言い難く、近年の研究では既存モデルの分析が盛んに行われている。本論文では、多言語自然言語処理モデルを分析した 2

[1] https://www.ethnologue.com/

つの実験結果についてまとめる。

1.2 本研究の構成

本論文の以降の構成は以下のようになっている:

第 2 章 単語埋め込み計算のための文脈窓

この章では、多言語単語埋め込みと文脈窓の関係を調査した研究について記述する。ここではまず、現代の自然言語処理の基礎となっている単語埋め込みの技術について導入した後、その多言語への拡張について説明する。そして、単言語埋め込みの学習における文脈窓と、マッピングベースの多言語単語埋め込みの手法の関連性について論じたのち、それらの関係を実験を通じて経験的に分析した結果、及びその示唆するところを述べる。

第 3 章 機械翻訳モデルが捉える言語類型論

この章では、多言語機械翻訳モデルの内部表現を調べた研究について記述する。導入では、ニューラルネットワークに基づく機械翻訳の仕組みについて説明した後、近年の probing タスクの枠組みを用いた分析について紹介する。実験では、多言語機械翻訳モデルの内部表現から入出力言語の言語類型論的特徴を予測するタスクを通じて、モデルが入出力の言語をどれだけ捉えているかを調査した。

第 4 章 おわりに

最後に、本論文のまとめと今後の展望について述べる。

第2章　単語埋め込み計算のための文脈窓

2.1　序論

複数言語で共通の意味空間を学習する多言語単語埋め込み（Cross-lingual Word Embedding）は、様々な多言語 NLP システムの基盤となるという点で重要な要素技術である。多言語単語埋め込みを計算する手法の中でも特に、独立に学習された複数の単語空間を共通の空間に線形写像でマップする写像ベース (mapping-based) の手法が近年盛んに研究されている [6, 7]。写像ベースの手法は、異なる言語の単語埋め込み空間の構造が類似している、もしくは近傍グラフの構造が同型である [8] という強い仮定に基づいている。つまり、写像ベースの手法の性能は、大元の単言語の単語埋め込み空間の構造に強く依存している。

単語埋め込み空間の構造は、訓練コーパスにおける単語の共起情報によって決定される [9, 10]。実際のアルゴリズムでは、文内の単語について文脈窓 (context window) を定義し、その文脈窓にどのような単語が現れるかの情報によって、単語埋め込みが計算される。この文脈窓の選択は、単語埋め込み空間の構造の決定に大きな影響を与える。例えば、前後 1〜2 単語を範囲とする狭い文脈窓で単語埋め込みを訓練するとその埋め込み空間は単語の文法的な性質を捉え、広い文脈窓では単語のトピック的な性質を捉える傾向にあることが知られている [11]。日本語の単語埋め込み空間における近傍単語の例を表 2.2 に示す。このように、文脈窓と単語埋め込み空間の構造は密接に関連しているにも関わらず、文脈窓の選択が多言語の単語埋め込み空間の構造的類似性、ひいてはマッピングの質にどのような影響を与えるかを調べた研究は少ない。

Query word	window size 1	window size 10
言語学	宗教学	比較言語学
	社会学	類型論
	発生学	記号論
	統計学	言語学者
	音声学	生成文法

表 2.1: 異なる窓サイズで訓練された日本語単語埋め込み空間における、上位近傍単語。小さい窓サイズは機能的な類似性 (「-学」) を捉え、大きな窓サイズはトピック的な類似性を捉えている。

本研究では、文脈窓と多言語単語埋め込みの関係について理解を深めるために、異なる文脈窓サイズの単語埋め込みを複数言語で訓練し、それらのマッピングの性能を測定する実験を行った。マッピングの性能は近傍から翻訳に当たる単語を取得する対訳辞書生成のタスクで評価した。その結果、全体的な傾向として、窓サイズを大きくすればするほど、２つの埋め込み空間はマップしや

すくなることが観察された。

2.2 背景：単語埋め込み

2.2.1 単語の意味を捉えたベクトル

近年の NLP システムは機械学習を用いるものが大半であり、モデルへの入力では単語を素性ベクトルとして表現する。単語をベクトルとして表す最も単純な方法は、語彙 W 中の各単語に対して番号を割り振り、その番号の次元が 1、その他の次元が 0 であるような $|W|$ 次元の One-Hot ベクトルを用いる方法である。これを用いると、例えば「コンピュータ」と「計算機」という単語に 0 番と 3 番を割り振ったとすると、それぞれのベクトルは以下のようになる：

$$\mathbf{v}(\text{コンピュータ}) = (1, 0, 0, 0, 0, ..., 0) \tag{2.1}$$

$$\mathbf{v}(\text{計算機}) = (0, 0, 0, 1, 0, ..., 0) \tag{2.2}$$

しかしながら、One-Hot ベクトルには 2 つの大きな問題点がある。 1 つは、語彙サイズ $|W|$ が大きいときに、単語ベクトルも大きくなってしまい、ベクトルをそのまま扱うには非効率だという点である。例えば、日本語版 Wikipedia に含まれる単語の総種類数は数百万にも上り、この次元のベクトルを計算機上で扱うには何かしらの工夫が必要となる。

もう 1 つの問題は、One-Hot ベクトルでは単語の意味を上手く表現出来ない点である。上の例で言えば、「コンピュータ」と「計算機」の One-Hot ベクトルはその 2 つの単語が異なるものである以上の情報を何も持たず、「コンピュータ」と「計算機」が同義語であることなどを表現出来ていない。

以上の問題を解決し、単語の意味を表現したベクトルを得るための技術が**単語埋め込み** (word embedding) である。単語埋め込みは単語の意味を表現した低次元（50 〜 1000）の実数値ベクトルであり、似た意味を持つ単語は埋め込み空間上でのベクトルも近いという性質を持つ（図 2.1）。

このような単語の意味を捉えたベクトルを計算機上で実現するのに鍵となる考えが**分布仮説** (distributional hypothesis) である。この考えは Firth [12] の以下の有名な引用がよく表すところである。

> You shall know a word by the company it keeps.
> ある単語をよく知るには周りを見ればよい。

単語の意味というのは、周りにどのような単語が現れるか、つまり**文脈**によって特徴付けることが出来る。例えば、「私は昨日夕食にマヘウを飲んでンシマを食べた。」という文章を読んだときに、「マヘウ」も「ンシマ」も知らなかったとしても、周りの単語（「夕食」、「飲んで」、「食べた」など）を見ると「フフ」は飲み物で「ンシマ」は食べ物だということはわかるだろう[2]。この分布仮説の考えに基づき、単語埋め込みは言語のコーパスから、単語の共起情報に基づいて計算される。

[2]「マヘウ」はトウモロコシを発酵させて作ったジュース、「ンシマ」はトウモロコシを挽いた粉を湯で練り固めて作った主食で、共にザンビアの郷土料理。

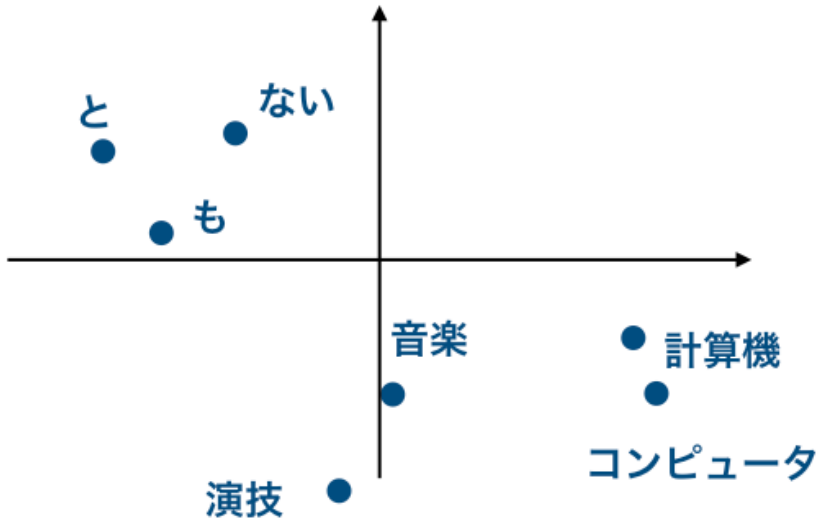

図 2.1: 単語埋め込みを 2 次元平面状に写像した際のイメージ図：「計算機」と「コンピュータ」など似た意味の単語のベクトルは近くに配置される。

2.2.2　カウントベースの手法

分布仮説の考えを単純に実装するには、単語のベクトルとして、周りにどのような単語が現れるかを数えたものをベクトルとすればよい。例えば、単語にわかち書きされた以下の 1 文からなるコーパスを考える。

私　は　昨日　夕食　に　マヘウ　を　飲ん　で　ンシマ　を　食べ　た

この中にある名詞について、前後 2 単語にどのような単語が現れるかを数えた単語共起行列を作ると以下のようになる。

	私	昨日	夕食	マヘウ	ンシマ	飲ん	食べ	は	に	を	で	た
私	0	1	0	0	0	0	0	1	0	0	0	0
昨日	1	1	0	0	0	0	0	1	1	0	0	0
夕食	0	1	0	1	0	0	0	1	1	0	0	0
マヘウ	0	0	1	0	0	1	0	0	1	1	0	0
ンシマ	0	0	0	0	0	1	1	0	0	1	1	0

この行列の行がそれぞれの単語の意味を表すベクトルであるとみなす事ができる。行と列はどちらも単語であるが、概念上区別するために、列毎の単語は「それぞれの単語の周りに現れた単語」として**文脈語**と称して区別することにする。

この単純に単語の頻度を数えたものは、そのまま単語ベクトルとして使用するにはいくつかの好ましくない性質がある。まず、単語の頻度を直接ベクトルの成分に使うのは高頻度語の影響が必要以上に強調されてしまう。例えば、日本語では「は」や「に」などの助詞は他の単語に比べて頻度が高く、大部分の単語と共起することになり、またベクトルの成分も大きくなる。しかし、大部分の単語と共起するということは、各単語の意味を差別化するのにあまり貢献しないということであり、そのような情報がベクトルの成分で大きい値を持つのは好ましくない。そこで、共起行列の成分を単語と文脈語の自己相互情報量 (point-wise mutual information; PMI) に置き換えるのが一般的である。頻度の共起行列を $\mathbf{M}$、自己相互情報量に置き換えた行列を $\mathbf{M}^{PMI}$ とすると以下のように書ける。

$$\mathbf{M}^{PMI}_{i,j} = \log \frac{p(w_i, c_j)}{p(w_i)p(c_j)} \tag{2.3}$$

$$p(w_i, c_j) = \frac{\mathbf{M}_{i,j}}{\sum_{i,j} \mathbf{M}_{i,j}} \tag{2.4}$$

$$p(w_i) = \frac{\sum_i \mathbf{M}_{i,j}}{\sum_{i,j} \mathbf{M}_{i,j}} \tag{2.5}$$

$$p(c_j) = \frac{\sum_j \mathbf{M}_{i,j}}{\sum_{i,j} \mathbf{M}_{i,j}} \tag{2.6}$$

$$\tag{2.7}$$

ここで、 w_i は共起行列の i 行目の単語、c_j は共起行列の j 列目の文脈語を表す。これにより、元の単語と文脈語の頻度を考慮しつつ、特徴的な単語・文脈語の情報を捉えることが可能になる。

次に、この共起行列について次元削減を行う。$\mathbf{M}^{PMI}$ のままでは、単語ベクトルの次元が大きくかつ疎であるので、次元削減により低次元で密なベクトルし、計算機上で扱いやすくする。まず、$\mathbf{M}^{PMI}$ は特異値分解によって以下のように分解される：

$$\mathbf{M}^{PMI} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\mathrm{T}} \tag{2.8}$$

ここで $\mathbf{U}$ と $\mathbf{V}$ は直交行列であり、$\mathbf{\Sigma}$ は特異値を対角成分に持つ行列である。そして、上位 k 個の特異値を選び、単語埋め込み行列 $\mathbf{X}$ を得る。

$$\mathbf{X} = \mathbf{U}_k\mathbf{\Sigma}_k. \tag{2.9}$$

共起行列の次元削減に基づくカウントベースの手法はシンプルかつ有用であるが、次元削減の計算量がボトルネックとなる。特異値分解の計算量は語彙サイズ $|W|$ に対して、3 乗のオーダー $\mathcal{O}(|V|^3)$ の計算量がかかり、大規模コーパスで学習し語彙サイズが数十万を超える場合など、現実的な時間で計算が終わらなくなってしまう。一方で、次に紹介する予測ベースの手法では、単語埋め込みを計算するのにかかる時間を大きく短縮でき、かつカウントベースの手法の性能を上回るようなものが提案されている。

2.2.3 予測ベースの手法

コーパスからを求めた共起行列を元にしたカウントベースの手法と異なり、予測ベースの手法は単語ベクトルをパラメータとして持ち、特定の目的関数を最適化する形でそのパラメータを更新することで、単語ベクトルを得る。代表的なアルゴリズムとして、ここでは Skip-gram [13] を紹介する。

Skip-gram のモデルは訓練コーパス D 中の単語 $w \in W$ と文脈語 $c \in C$ について、単語ベクトル $\mathbf{w} \in \mathbb{R}^d$ と文脈語ベクトル $\mathbf{c} \in \mathbb{R}^d$ をそれぞれ割り当て、これらをパラメータ θ として、以下の目的関数を最大化する。

$$\arg\max_{\theta} \prod_{(w,c)\in D} p(c|w;\theta) \tag{2.10}$$

ここで $(w,c) \in D$ は訓練コーパス D から抽出できる単語・文脈語ペアである。
この目的関数を直感的に理解するためには、次にように考えれば良い：

> ある単語 w に関して、周りにどのような文脈語 c が現れるかの確率分布 $p(c|w;\theta)$ を構成できるようなパラメータ（ベクトル） $\mathbf{w}, \mathbf{c} \in \theta$ は、単語の意味をよく捉えていると言える。

つまり予測ベースの手法でも、分布仮説に基づき周りに現れる文脈語を以って単語の意味をモデル化しているのは変わらない。

通常の Skip-gram では、$p(c|w;\theta)$ は以下のように、内積 $\mathbf{w}\cdot\mathbf{c}$ とソフトマックス関数を用いて計算される。

$$p(c|w;\theta) = \frac{\exp(\mathbf{w}\cdot\mathbf{c})}{\sum_{c'\in C}\exp(\mathbf{w}\cdot\mathbf{c}')} \tag{2.11}$$

ここで、内積 $\mathbf{w}\cdot\mathbf{c}$ は w と c の共起のしやすさを表すスコアだとみなす事ができる。以上の目的関数を用いて、コーパス内の単語・文脈語ペア (w,c) 毎に確率的勾配降下法でパラメータを最適化すれば、所望の単語ベクトル $\mathbf{w}$ が得られる。

しかしながら、上記のモデル化では $\sum_{c'\in C}\exp(\mathbf{w}\cdot\mathbf{c}')$ の項で、非常に計算コストがかかる。文脈語の語彙サイズ $|C|$ が非常に大きいため、全ての単語・文脈語ペアについてこれを計算するのは、大規模コーパスを使用する場合、現実的ではない。

この計算コストの問題を解決する方法の 1 つが負例サンプリング (Negative Sampling) [14] である。これと Skip-gram を組み合わせた手法は Skip-gram with Negative Sampling (SGNS) の名前、または 2013 年に発表された実装の `Word2Vec`[3] の名前で知られ、単語埋め込みの計算をする際の標準手法となっている。

通常の Skip-gram から負例サンプリングを用いた手法に移行するために必要な考えが、予測 (prediction) から弁別 (discrimination) への転換である。通常の Skip-gram では「ある単語の周りにどのような文脈語が現れるか」という観点から $p(c|w;\theta)$ に基づいた目的関数を最適化した。つ

[3] `https://code.google.com/archive/p/word2vec/`

まり、単語ベクトル w はどのような文脈語が周りに現れるかを「予測」しなくてはならなかった。一方で、SGNS では「ある単語・文脈語ペアが訓練コーパスに現れるかどうか」を「弁別」するようにパラメータを最適化する。単語・文脈語ペア (w, c) が訓練コーパスに現れる確率を $p(w, c; \theta)$ とすると、訓練コーパス D を使った目的関数は以下のようになる。

$$\arg\max_{\theta} \prod_{(w,c)\in D} p(w, c; \theta) \tag{2.12}$$

ここで $p(w, c; \theta)$ は内積とシグモイド関数を使ってモデル化される。

$$p(w, c; \theta) = \frac{1}{1 + \exp(-\mathbf{w} \cdot \mathbf{c})} \tag{2.13}$$

しかしながら、式 2.12 の目的関数は望まない自明解を持つ。あらゆる (w, c) について $\mathbf{w} \approx \mathbf{c}$ となるようにし、内積がある程度大きくなるように最適化してしまえば $p(c, w; \theta) = 1$ となり、容易に目的関数を最大化できてしまう。

これを防ぐために負例サンプリングを使う。上の定式化では、訓練コーパスに現れる単語・文脈語ペアという「正例」しか与えられないのが問題であった。訓練コーパスには現れない単語・文脈語ペア、つまり「負例」をランダムに生成する負例コーパス D' を仮定し、目的関数を以下のように書き換える。

$$\arg\max_{\theta} \prod_{(w,c)\in D} p(w, c; \theta) \prod_{(w,c)\in D'} (1 - p(w, c; \theta)) \tag{2.14}$$

実際の計算では、正例 (w, c) 1 つに対して、同様の単語 w に対して負例を k 個 $(w, c_1), (w, c_2), ..., (w, c_k)$ サンプルする。このとき、負例 c_i は文脈語の頻度の $\frac{3}{4}$ 乗に比例した分布からサンプルするのが経験的に良いとされている [14]。つまり、負例コーパスは以下の分布から単語・文脈語ペアを生成するコーパスだとみなすことができる。

$$(w, c) \sim p_{words}(w) \frac{p_{contexts}(c)^{3/4}}{Z} \tag{2.15}$$

ここで $p_{words}(w)$ は訓練コーパス中での単語 w が現れる確率、$p_{contexts}(c)$ は文脈語 c が現れる確率、Z は正規化項である。

2.2.4 文脈窓

以上紹介したカウントベースの手法と予測ベースの手法共に、分布仮説に基づいて、単語 w の周りにどのような文脈語 c が現れるかの情報を用いて単語埋め込みを計算するものであった。文が与えられた際に、それぞれの単語の周りに現れる文脈語の定義を**文脈窓** (context window) と呼ぶ。この文脈窓の定義は、単語埋め込みがどのような意味を捉えるかを決定する要素で、非常に大事なものとなる。

最も一般的な文脈窓は線形文脈窓 (linear context window) であり、これは注目する単語の前後 k 単語を文脈語として抽出する。このパラメータ k を**窓サイズ** (window size) と呼ぶ。

文脈窓の選択は、計算される単語埋め込みの性質に大きな影響を与えることが知られている。例えば、窓サイズを小さく設定したときの単語埋め込みは、単語の文法的な類似性を捉えるようになり、反対に窓サイズを大きくした場合、単語のトピック的な類似性を捉えることが分かっている（図 2.2）。

単語	窓サイズ 1	窓サイズ 10
言語学	宗教学	比較言語学
	社会学	類型論
	発生学	記号論
	統計学	言語学者
	音声学	生成文法
喋る	走り回る	話す
	唄う	カタコト
	暴れる	しゃべり
	持ち歩く	関西弁
	飼う	京都弁

表 2.2: 異なる窓サイズで訓練された日本語単語埋め込み空間における、上位近傍単語。小さい窓サイズは機能的な類似性 (「-学」) を捉え、大きな窓サイズはトピック的な類似性を捉えている。

また、線形窓以外にも、単語の文中での係り受け関係に基づいて文脈を抽出する係り受け文脈窓 (dependency context window) も提案されており、線形文脈窓よりもより文法的な性質を捉えることが示されている [11]。

本研究では、文脈窓の選択と、次に紹介する多言語単語埋め込みにおける 2 つの単語埋め込み空間の構造的類似性との関係を明らかにする。

2.3 背景：写像ベースの多言語単語埋め込み

通常、単語埋め込みは単一言語のコーパスを用いて、その言語に閉じた意味を学習することを試みる。この単語埋め込みを多言語に拡張したものが**多言語単語埋め込み** (multi-lingual word embedding) である。例えば、日本語と英語の多言語単語埋め込み空間上では、同じ意味をもつ「計算機」と *computer* のベクトルも近くなることが期待される。

多言語単語埋め込みの獲得方法は以下の 2 つに大別される：複数言語の単語埋め込みを、同じ意味の単語はベクトルも似たものになるような制約をかけながら学習する同時学習 (joint learning) の手法、また訓練済みの単語埋め込み同士を共通の空間にマップする写像ベース (mapping-based) の手法である [15]。本研究は写像ベースの手法のみを対象とする。

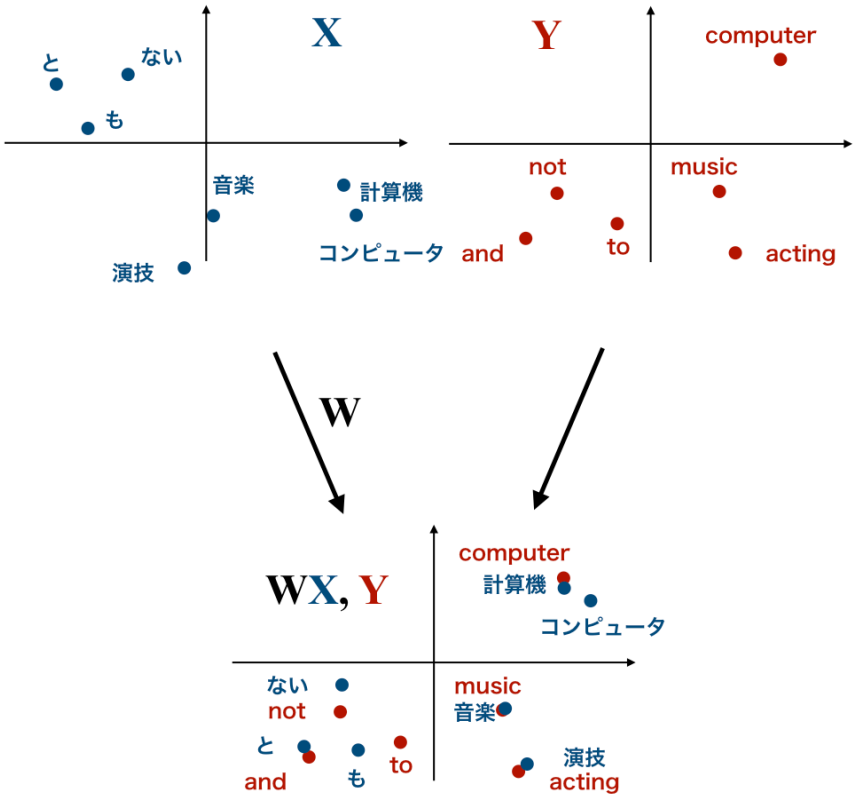

図 2.2: 日本語と英語の単語埋め込みを共通の空間にマップするイメージ図

写像ベースの手法は異なる言語でそれぞれ訓練した単語埋め込みが似たような構造を持つ、という観察 [7] から発展してきた手法である。ソース言語の単語埋め込み $\mathbf{x}$ を ターゲット言語の単語埋め込み空間上に写像する翻訳行列 (translation matrix; alignment matrix) $\mathbf{Z}$ を学習する（図 2.2）。

2.3.1　翻訳行列の学習

翻訳行列 $\mathbf{Z}$ を計算する最も単純な方法は、 2 言語の単語埋め込みと訓練用単語辞書 $(x_i, y_i)_{i=1}^m$ を用いて、次の最小化問題を解くものである [7]。

$$\arg\min_{\mathbf{Z}} \sum_{i=1}^{m} \|\mathbf{Z}\mathbf{x}_i - \mathbf{y}_i\|^2 \tag{2.16}$$

以下、ソース言語とターゲット言語の単語ベクトルをまとめて、それぞれ行列 $\mathbf{X}, \mathbf{Y}$ とし、それぞれの同じ行が辞書上の単語に対応するとして表記する。すると、式 2.16 は以下のようにかける。

$$\arg\min_{\mathbf{Z}} \|\mathbf{ZX} - \mathbf{Y}\|_F^2 \tag{2.17}$$

この最小化問題の解は、X のムーア-ペンローズの擬似逆行列を $\mathbf{X}^+ = \left(\mathbf{X}^\top \mathbf{X}\right)^{-1} \mathbf{X}^\top$ として、$\mathbf{Z} = \mathbf{X}^+ \mathbf{Y}$ で与えられる。

このマッピングの質を向上させるための制約や前処理として様々なものが提案されている。

直交行列の制約

マッピングの品質を上げるためによく使われる手法として、$\mathbf{Z}$ に直交行列となるような制約 $\mathbf{Z}^\top \mathbf{Z} = \mathbf{I}$ を課すというものがある [6, 16]。この制約下での、式 2.17 の解は $\mathbf{Z}^\top \mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$ を $\mathbf{Z}^\top \mathbf{X}$ の特異値分解として、$\mathbf{Z} = \mathbf{V}\mathbf{U}^\top$ で与えられる。

直交制約が必要となる理由として、後述する単語埋め込みのノルム正規化を保存するため [16]、また単言語埋め込み空間内の構造を変えずに保つため [6] と考えられている。

ノルム正規化

式 2.17 の最小化問題では、ノルムが大きい単語ベクトルの影響が大きくなってしまう。この影響を打ち消し、訓練辞書データ内の単語が全て等しく目的関数に寄与するようにするために、ノルム正規化 (length normalization) が使われる [6, 16]。この処理では、各単語ベクトル $\mathbf{x}$ はノルムが 1 になるように正規化される。

$$\mathbf{x} \leftarrow \frac{\mathbf{x}}{|\mathbf{x}|} \tag{2.18}$$

中心化

中心化 (mean centering) の前処理は語彙 $|V|$ 内の全ての単語ベクトルの平均が $\mathbf{0}$ になるように正規化する。

$$\mathbf{x} \leftarrow \mathbf{x} - \frac{1}{|V|} \sum_{i}^{|V|} \mathbf{x}_i \tag{2.19}$$

これは次のような直感を実装したものとなる：「語彙内からランダムに単語を 2 つサンプリングした際に、それらの単語は無関係であることが多い。つまり、それらの単語ベクトルの内積（類似度）を計算した際の期待値は 0 であることが好ましい。」実際に、中心化を施した単語埋め込みの内積の期待値は 0 となる。

2.3.2 対訳辞書生成による評価

多言語単語埋め込みの評価には主に、対訳辞書生成 (bilingual lexicon induction; BLI) のタスクが使われる。このタスクでは、評価用単語辞書 $(x_i, y_i)_{i=1}^m$ を用意し、それぞれの単語ベクトル $\mathbf{x}_i, \mathbf{y}_i$ が多言語埋め込み空間上で近い位置にあるかどうかを評価する。埋め込み同士の距離尺度としてはコサイン類似度を使うのが一般的である。[4]

評価尺度として、ソース単語の翻訳に当たる単語が、ベクトルのコサイン類似度での上位 k 件のターゲット単語内にあるかどうかを精度として算出する top-k [5]や、正解単語の順位の逆数を平均した平均逆順位 (mean reciprocal rank) がある。

2.3.3 単語埋め込み同型の仮定

写像ベースの手法は、以下の単語埋め込み同型の仮定 (isomorphism assumption) を前提としている。

> 異なる言語でそれぞれ訓練した単語埋め込みであっても、空間内での単語間の配置が似ているなど、構造的類似性を持つ。

しかし、この仮定は必ずしも成り立つわけではないということが、近年の研究で示されている。ソース言語とターゲット言語が言語系統的に離れている場合や、ソース言語とターゲット言語のコーパスが互いにドメインの違うものであると、対訳辞書生成のスコアが著しく低下する [8]。

上にあげた要素以外にも様々なものが、2 言語間の単語埋め込みの構造の関係性に影響し得る。その中でも、文脈窓の選択が単語埋め込みの構造に影響を大きく与えることは、2.2.4 節で述べた通りである。文脈窓の選択と写像ベースの多言語単語埋め込みの質との間には、直接的な関係があると予想されるにも関わらずこれまで十分に調査されることはなかった。本研究では、文脈窓と多言語単語埋め込みの関係について理解を深めるために、異なる文脈窓サイズの単語埋め込みを訓練し、それらのマッピングの性能を測定する実験を行った。

2.4 実験設定

本研究での実験は次の 3 段階に分けることができる。

1. ソース言語とターゲット言語のそれぞれの単語埋め込みの訓練
2. 写像ベースでの単語埋め込みのマッピング
3. 得られた多言語単語埋め込みの評価

以下、それぞれの段階での実験設定について説明する。

[4]前述のノルム正規化の前処理を行いかつ翻訳行列が直交行列であれば、多言語単語埋め込みのノルムは全て 1 となり、この場合単語ベクトル同士のコサイン類似度は内積と同値となる。

[5]k の値として、1、5、１０がよく用いられる。

2.4.1 単言語単語埋め込みの訓練

最初に、単言語単語埋め込みを訓練する。このときに異なる文脈窓で訓練したものを用意し、後段でそれをマッピングして得られた多言語単語埋め込みの評価をする。

言語

実験に用いる言語は、言語資源の豊富さと語族の多様性を考慮して、ターゲット言語として英語 (En) 、ソース言語としてフランス語 (Fr)、ドイツ語 (De)、ロシア語 (Ru)、日本語 (Ja) を使用した。

コーパス

単語埋め込みを訓練するためのコーパスには Wikipedia Comparable Corpora[6] を用いた。Comparable コーパスを用いた理由は、各言語である程度のデータ量が確保でき、かつ多言語単語埋め込みが学習しやすい設定にすることで文脈窓の影響を強調できるためである。ソース言語には 100 万文、ターゲット言語には 500 万文用いた。

ソース言語とターゲット言語のコーパスのドメインが異なる場合、埋め込み空間の構造も異なってしまうことが知られている [8]。今回 comparable コーパスで得られた傾向が、ドメイン違いのコーパスでも成り立つがどうかを調べるために、ソース言語で異なるドメイン（ニュース）のコーパス[7]も用いて埋め込みを訓練した。

文脈窓

文脈窓の影響を調査するために、線形文脈窓のサイズを 1, 2, 3, 4, 5, 7, 10, 15, 20 の間で変化させた。

線形文脈窓の他に、係り受け関係に基づいた文脈窓 [17] から訓練した埋め込みについても調べた。当初の予想では、係り受け関係は、線形文脈窓と異なり、言語毎に異なる語順の影響を受けにくいため、より言語普遍な性質を持つ単語埋め込みが得られ、結果マッピングもしやすくなると考えた。しかし予備実験の結果、係り受け文脈窓の性能は、同様の訓練単語・文脈語ペア数を持つ線形文脈窓と同様の性能を示し、これといった特徴的な傾向はみられなかった。具体的に言えば、今回用いた係り受け文脈窓から抽出できた単語・文脈語ペア数は、線形文脈窓サイズ 3 のものと同程度数であったが、マッピングの性能も両者の間で同程度であった。係り受け文脈窓を使うことにより生じる線形文脈窓との違いは今回観察されなかったため、以下の分析では線形文脈窓についてのみ論じる。

[6] https://linguatools.org/tools/corpora/wikipedia-comparable-corpora/
[7] https://wortschatz.uni-leipzig.de/en/download

アルゴリズムの実装

単語埋め込みの計算には Skip-gram with Negative Sampling [14] の手法を用いたが、窓サイズの影響について論じる際は、その実装に注意しなくてはならない。オリジナルの C 言語によって実装された `Word2Vec`[8] や、python 実装である `Gensim`[9] は dynamic window の仕組みを採用しており、各トークンに対する実際の窓サイズは 1 から設定された窓サイズの間から一様にサンプリングされる [13]。また、上に挙げた実装では高頻度のトークンを subsampling によって取り除くことで訓練を効率化しているが、この高頻度トークンの除去は単語・文脈語ペアを抽出する前に行われるため、実質文脈窓サイズを増やすこととなる（“dirty” subsampling と呼ばれる [18]）。こうした dynamic window と dirty sub-sampling の仕組みは、実質の文脈窓サイズを変化させるため、窓サイズが埋め込みに与える影響を曖昧にしてしまう可能性がある。従って、本実験では `word2vecf`[10] に基づいて実験を行う。 `word2vecf` は入力として直接、単語・文脈語ペアを与えることが出来るので、固定した窓サイズから単語・文脈語ペアを抽出した後に、sub-sampling を行った。

ハイパーパラメータ

単語埋め込み訓練時のハイパーパラメータを表 2.3 に示す。

	ソース言語 (Fr, De, Ru, Ja, 100 万文)	ターゲット言語 (En, 500 万文)
単語埋め込みの次元数	300	
負例の数	15	
subsampling 率	0.001	
語彙に含まれる単語の最小頻度	10	15
エポック数	10	5

表 2.3: 単語埋め込み訓練時の設定

2.4.2 単語埋め込みのマッピング

単言語埋め込みを訓練した後は、写像ベースの手法で 2 言語の埋め込み空間を揃えた。空間を揃えるための行列 $\mathbf{Z}$ は 2 言語の単語埋め込みと単語辞書から計算される。ここでは、単語辞書を $(x_i, y_i)_{i=1}^m$ として、最小化問題 $\arg\min_{\mathbf{Z}} \sum_{i=1}^m \|\mathbf{Z}\mathbf{x}_i - \mathbf{y}_i\|^2$ を解く一般的な手法を用いた [7]。このとき、マッピングの品質を上げるために、$\mathbf{Z}$ には直交行列になるような制約を課し、単語埋め込みにはマッピング前にノルム正規化と中心化の前処理を適用した [6]。

[8] `https://code.google.com/archive/p/word2vec/`
[9] `https://radimrehurek.com/gensim/`
[10] `https://bitbucket.org/yoavgo/word2vecf/src/default/`

訓練と評価に用いた単語辞書は Google Translate [11] から構築した。ターゲット言語である英語の語彙中にある単語を全てソース言語に翻訳し、ソース言語にない単語は取り除いた。これをソース言語から英語への逆方向についても行い、それぞれで得られた翻訳単語対の和集合を取り、対象の言語対の辞書を得た。この中から訓練データとして 5,000 対、評価データとして 2,000 対の単語翻訳ペアを各言語対でランダムに抽出した。

プログラムの乱数シードを変え、全ての設定について 3 つずつ多言語埋め込みを訓練した。以下の結果では、それらの平均を標準偏差と共に示す。

2.4.3 多言語単語埋め込みの評価

文脈窓サイズを変えて得られた多言語単語埋め込みを対訳辞書生成のタスクで評価する。対訳辞書生成は評価用辞書内の翻訳単語ペアについて、ソース言語の単語埋め込みから、多言語単埋め込み空間におけるコサイン類似度に基づく近傍探索によってターゲット単語を取得するタスクである。ここでは、評価尺度としては平均逆順位 (mean reciprocal rank; MRR) を用いた。N 単語ペアと、それらの近傍探索での順位 rank_i が与えられたとき、平均逆順位スコアは以下のように計算される。

$$\frac{1}{N}\sum_{i=1}^{N}\frac{1}{\mathrm{rank}_i} \tag{2.20}$$

対訳辞書生成は簡便に多言語単語埋め込みを評価できるメリットがある一方で、そのスコアは下流タスクでの性能と必ずしも相関しないということが指摘されている [19]。したがって、文脈窓の影響についてさらなる知見を得るために、言語横断の下流タスクを用いても評価した。言語横断の下流タスクでは、ある言語の訓練データで学習したモデルが、異なる言語の評価データについてどのくらいの性能を発揮するか、というのを多言語埋め込みを用いて評価する。具体的には、単語埋め込み列を入力として受け取るモデルを $\mathcal{M}(\theta)$ とする。多言語単語埋め込みを言語横断の下流タスクで評価する際には、ソース言語の訓練データを $\mathcal{D}_{src}$ として、そこからの入力系列データを多言語単語埋め込みに変換し、モデル $\mathcal{M}(\theta)$ を訓練する。この時、モデルパラメータ θ は更新されるが、単語埋め込みは更新されない。その後、訓練済みモデルをターゲット言語の評価データ $\mathcal{D}_{tgt}$ で評価する。この時の入力は訓練時と同様に多言語単語埋め込みなので、ソース言語とターゲット言語で共通の意味空間が出来ていれば、評価時と訓練時の言語が異なっていても、ある程度の精度が期待できる。

今回の実験では、以下の 3 タスクを用いて評価を行った。

極性分析

極性分析は、与えられた文書の極性（ポジティブな内容か、ネガティブな内容か）を判定するタスクである。本実験では Webis-CLS-10 コーパス[12] [20] をデータセットとして用いた。これは

[11] `https://translate.google.com/` (October 2019)
[12] `https://webis.de/data/webis-cls-10.html`

Amazon の商品レビュー文とその評価（1 から 5 の 5 段階評価）から成るデータセットで、英語、ドイツ語、フランス語、日本語（今回の実験の 1 つであるロシア語は含まれていない）のデータが含まれる。今回は、極性分析をポジティブかネガティブかを判定する 2 値分類タスクとし、レビュー評価が 1-2 のものはネガティブ、4-5 のものはポジティブとし、評価が 3 のものは使用しなかった。データセットの統計を表 2.4 に示す。データこのタスクに用いるモデルは、文書分類でよく用いられる畳み込みにに基づいた分類器を用いた [21]。

訓練	開発	テスト
6,000	6,000	6,000

表 2.4: Webis-CLS-10 コーパス（極性分析）のデータ内訳

文書分類

文書分類は、与えられた文書のトピックを分類するタスクである。本実験では、MLDoc[13] [22] をデータセットとして用いた。これは、8 つの言語を対象に、ロイターの新聞記事をまとめたものになる。タスクとしては 4 クラス分類となり、以下のラベルが定義されている：`Corporate/Industrial`, `Economics`, `Government/Social`, `Markets`。データセットの統計を表 2.5 に示す。このタスクに用いるモデルも、極性分析同様に畳み込みにに基づいた分類器を用いた。

訓練	開発	テスト
10,000	1,000	4,000

表 2.5: MLDoc コーパス（文書分類）のデータ内訳

係り受け解析

係り受け解析は、単語分割された文を入力として、各単語間の係り受け関係を解析するタスクである。今回の実験では、言語非依存な係り受け関係を定義し、それに基づいたアノテーションを公開している Universal Dependencies [14] のデータセットを用いた。訓練データとして英語の UD English EWT dataset[15] [23]、その他の言語の評価データセットとして PUD treebanks を用いた。データセットの統計を表 2.6 に示す。モデルは Dozat ら [24] の LSTM に基づくグラフベースの係り受け解析器を使用した。評価スコアとして Labeled Attachment Score (LAS) を用いた。これは係り先と関係ラベルが共に正しいトークンの割合を表す。

[13] `https://github.com/facebookresearch/MLDoc`
[14] `https://universaldependencies.org/`
[15] `https://universaldependencies.org/treebanks/en_ewt/index.html`

訓練	開発	テスト
12,543	2,002	1,000

表 2.6: 係り受け解析タスクのデータ内訳

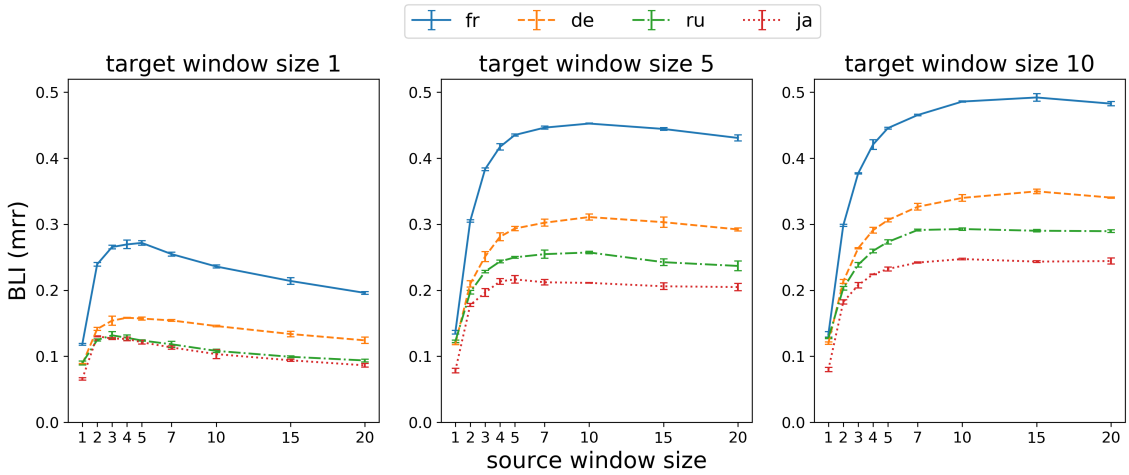

図 2.3: Comparable な設定における対訳辞書生成のスコア。ターゲット言語の窓サイズは固定し、ソース言語の窓サイズが変化している。

2.5 実験結果

2.5.1 対訳辞書生成による評価

ターゲット言語の文脈窓を固定した設定

はじめに、ソース言語とターゲット言語のコーパスが comparable であり、ターゲット言語の文脈窓サイズが固定された場合の実験結果を図 2.3 に示す。この設定は、言語資源が豊富なターゲット言語の単語埋め込みが、訓練済みモデルとして公開されているという一般的な状況を模したものになる。

結果から、ソース言語の窓サイズが小さいときはスコアも低くなることが分かる。これは今回のデータ量において、この窓サイズでは、十分な数の単語・文脈語ペアが得られず、質の良い単語埋め込みが訓練できなかったからだと考えられる。また、ターゲット言語の窓サイズに対して、一番スコアが高くなるソース言語の窓サイズは、ターゲット言語の窓サイズの周辺にあることが観察される。全体的には、ソース言語とターゲット言語の窓サイズをどちらも増やすことで、対訳辞書生成のスコアも上がる傾向にあると分かる。

この結果は一見すると Søgaard ら [8] の結果と一致しないように見える。Søgaard らは英語とスペイン語の単語埋め込みを `fasttext` [25] のアルゴリズムを用いて、窓サイズ 2 の設定で学習し、それらからを教師なしマッピングの手法 [26] を用いて多言語単語埋め込みを獲得した。彼ら

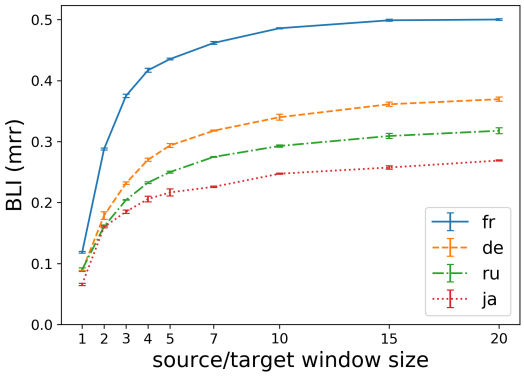

図 2.4: Comparable な設定における BLI スコア。

はこれに加えて、スペイン語の単語埋め込みの窓サイズを 10 に設定したものも学習し、マッピングの性能を比べたが、評価用単語辞書の top-1 の精度は 81.89 から 81.28 へとごく僅かに低下した。一方で、我々の実験結果では窓サイズを上げるとスコアは有意に上がる傾向にある。この観察結果のずれは実験設定の違いに起因するものだと考えられる。まず、`fasttext` の実装は dynamic window の仕組みを採用しており、窓サイズの影響を曖昧にしている可能性がある。また、彼らは単語埋め込みの訓練コーパスとして Wikipedia 全文を用いており、我々のものより遥かに大きい。最後に、`fasttext` の学習アルゴリズムは単語の文字レベルの情報を考慮して単語埋め込みを学習する。英語とスペイン語は表層が似ている単語を多く持つため、この文字レベル情報を用いることの影響は大きいと考えられる。

ソース/ターゲット言語の文脈窓を変化させる設定

ソース言語とターゲット言語の窓サイズを同じに揃え、どちらも変化させた時の設定の結果を図 2.4 に示す。

ソース言語とターゲット言語の窓サイズを増やすことにより、BLI のスコアも一貫して向上することが観察される。大きい窓サイズは、単語のトピック的な性質を捉えることを考えると、これはよりトピック的な性質を捉えた単語埋め込みの方が異なる言語間でマップしやすいからだと考えられる。トピックは、言語固有の文法などと異なり、コーパスが comparable である限り言語普遍だと考えられる。従って、トピックをより捉えた単語埋め込みは異なる言語間でマップがしやすいのも妥当なことだと思われる。

このトピック的な単語埋め込みはマップがしやすいという仮説は、単語の品詞毎の BLI の結果からも支持されると考えられる。直感的に、名詞は他の品詞に比べてトピックをよく表すことが多いと考えられるため、よりトピック的な性質を持った埋め込みが与えられているはずである。従って、名詞の BLI のスコアは単語埋め込みの窓サイズと特に強い相関を示すことが期待される。図 2.5 に各品詞のスコアとスピアマンの相関係数を示す[16]。全ての言語において、実際に名詞が相関係数 0.99 以上と最も強い相関を示している。

[16]各単語の品詞は、和田ら [27] にならい、Brown Corpus におけるその単語の最も頻度の高い品詞を用いた。

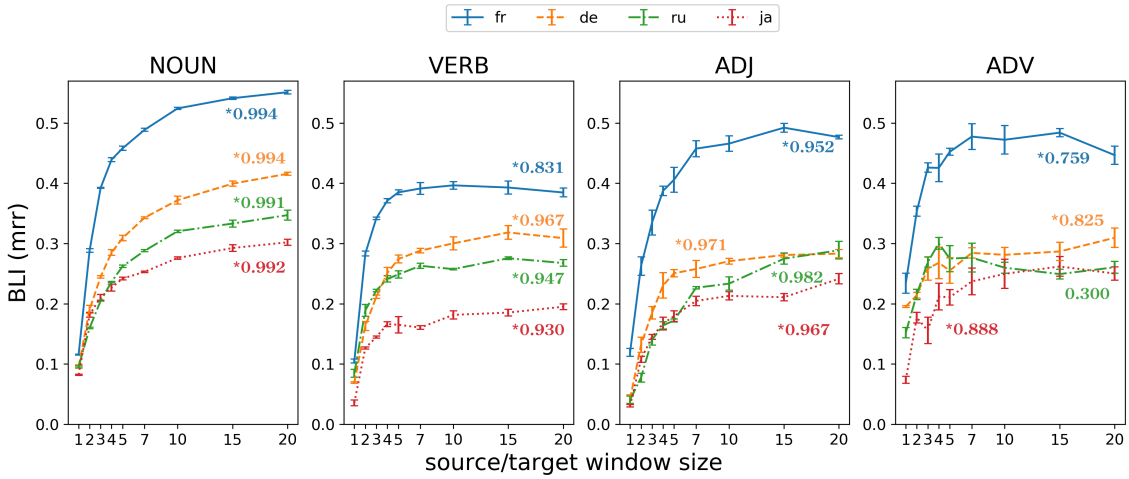

図 2.5: Comparable な設定における、各品詞の BLI スコア。グラフ上の数値はスコアと窓サイズのスピアマンの順位相関係数。統計的に有意な相関はアスタリスクで示される ($p < 0.05$)。

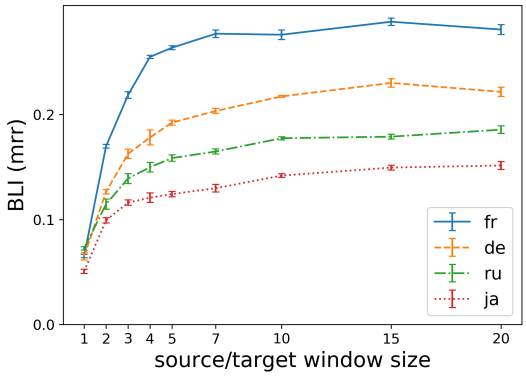

図 2.6: 異なるドメインの設定における BLI スコア。

異なるドメインの設定

これまでの結果は、ソース言語とターゲット言語のコーパスが comparable であるという比較的理想的な条件下で行った実験の結果であった。コーパスが comparable であるとき、2 つのコーパスは同じトピックを持つので、トピック的な単語埋め込みがマップしやすいのは当然だと思われる。この傾向が、異なるドメインのコーパスを用いた時にも見られるかどうかを調べるために、ソース言語のコーパスに異なるドメイン（ニュース）のコーパスを用いた結果を図 2.6 に示す。

まず、comparable の設定（図 2.4）に比べて、全体的に 0.1 〜 0.2 ポイント低いスコアを示している。これは、先行研究の、ドメインが一致していることが埋め込み空間の類似性に重要であるという観察 [8] と合致する。

次に、BLI の性能と窓サイズの関係については、窓サイズを大きくすると BLI のスコアも上がるという、comparable の設定と同じ傾向が観察される。このことは、たとえソース言語とターゲット言語でコーパスのドメインが異なっていても、窓サイズを大きくすることで単語埋め込みはドメ

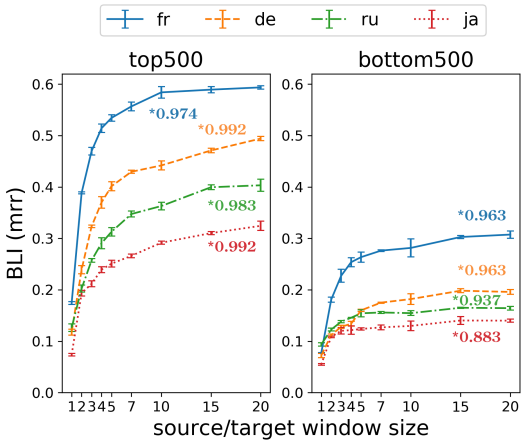

図 2.7: Comparable な設定における高頻度語と低頻度語の BLI スコア。

イン普遍なトピックを捉えマッピングがしやすくなるということを示唆している。

単語頻度による分析

窓サイズを広げることでどのような単語のマップがしやすくなるのかについて、さらなる知見を得るために、評価用の単語辞書から頻度が上位 500 の単語と下位 500 の単語を抜き出して、それらについてのスコアを評価した。Comparable の設定における結果を図 2.7 に示す。

高頻度語 (top500) のスコアが低頻度語 (bottom500) のものより低いのは、既存のマッピングの手法は低頻度語に弱いという、先行研究の観察と合致する [28, 29]。

窓サイズとの関係については、高頻度語と低頻度語のどちらも大きい窓サイズにするにつれてスコアが上がっているが、日本語 (Ja) とロシア語 (Ru) については上昇の傾向は弱い。

一方、ドメインが異なるコーパスでの結果（図 2.8）からは、低頻度語は、特にフランス語 (Fr) とロシア語 (Ru) で顕著であるが、窓サイズを大きくするとスコアが下がることが観察される。高頻度語は一貫して、窓サイズに比例してスコアが上昇している。

これは、ドメイン違いのコーパスでは単語埋め込みを学習する時に、高頻度語は多くの訓練事例である単語・文脈語ペアと関連づけられることで、大きい窓サイズを広げても意味上関係の無い文脈語によるノイズの影響を受けにくく、一方で低頻度語は限られた訓練事例の中で窓サイズを広げてしまうと、ノイズやドメインの違いが増幅され、結果マップがしにくくなるからだと考えられる。

2.5.2 下流タスクによる評価

3 つの下流タスク、極性分析、文書分類、係り受け解析による単語埋め込みの評価の結果を表 2.9 に示す。この結果では、ソース言語とターゲット言語の文脈窓は共に同じ値に設定し変化させている。

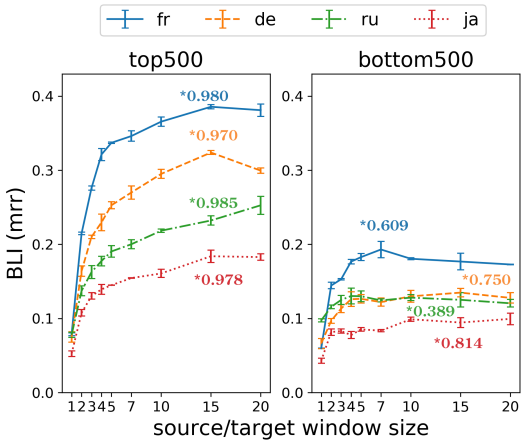

図 2.8: 異なるドメインの設定における高頻度語と低頻度語の BLI スコア。

極性分析と文書分類のタスクについて、どちらも英語の転移元タスクでは 3 から 5 の文脈窓が一番良い性能を発揮している。しかし、一方で、転移先のタスクではより大きい窓サイズが良いスコアを示す傾向にある。唯一の例外は日本語の文書分類タスクであり、文脈窓とスコアの相関は 0.158 と統計的に有意な相関を示していないが、これは日本語と英語は言語的な違いが大きいため、埋め込みをマップするのが難しいからだと考えられる。実際、文書分類における日本語のスコアは、他の言語に比べて一番低いものとなっている。

係り受け解析に関しては、英語の転移元タスクでは文脈窓が小さければ小さいほど良い、という傾向を示している。これは小さい文脈窓は、単語の文法的な側面を捉えるという観察に沿う結果となっている [11]。しかしながら、この転移元タスクでの傾向は、転移先タスクでは見られない。転移元タスクで一番良い性能を発揮していた窓サイズ 1 は、転移先タスクでは最も悪いスコアを示している。転移先タスクでは、強い相関は見られないものの、より大きい窓サイズが良い性能を示す傾向にある。これは恐らく、小さい窓サイズは単語の文法的な性質を捉えているとはいえ、対訳辞書生成のタスクの結果でも示されていたように単語のマッピングがしづらいためだと思われる。言語によって文法構造は異なるということを考えても、文法的な単語埋め込みはマッピングが難しいことが推測される。この結果は統語的なタスクにおける、言語間転移学習の難しさを示唆するものだと言えるだろう。

まとめると、下流タスクによる評価からわかる全体的な傾向は、転移元タスクにおける最適な窓サイズが必ずしも転移先のタスクでも最適であるとは限らないことである。実践的には、転移元と転移先の言語の単語埋め込みをうまくマッピングしやすい大きい窓サイズを選ぶのが良い、ということが言える。

2.6 結論

文脈窓の選択と多言語言語埋め込みの構造的類似性との間には、明示的に繋がりがあると考えられるにも関わらず、先行研究では十分に調査されて来なかった。この研究では、線形文脈窓の窓サ

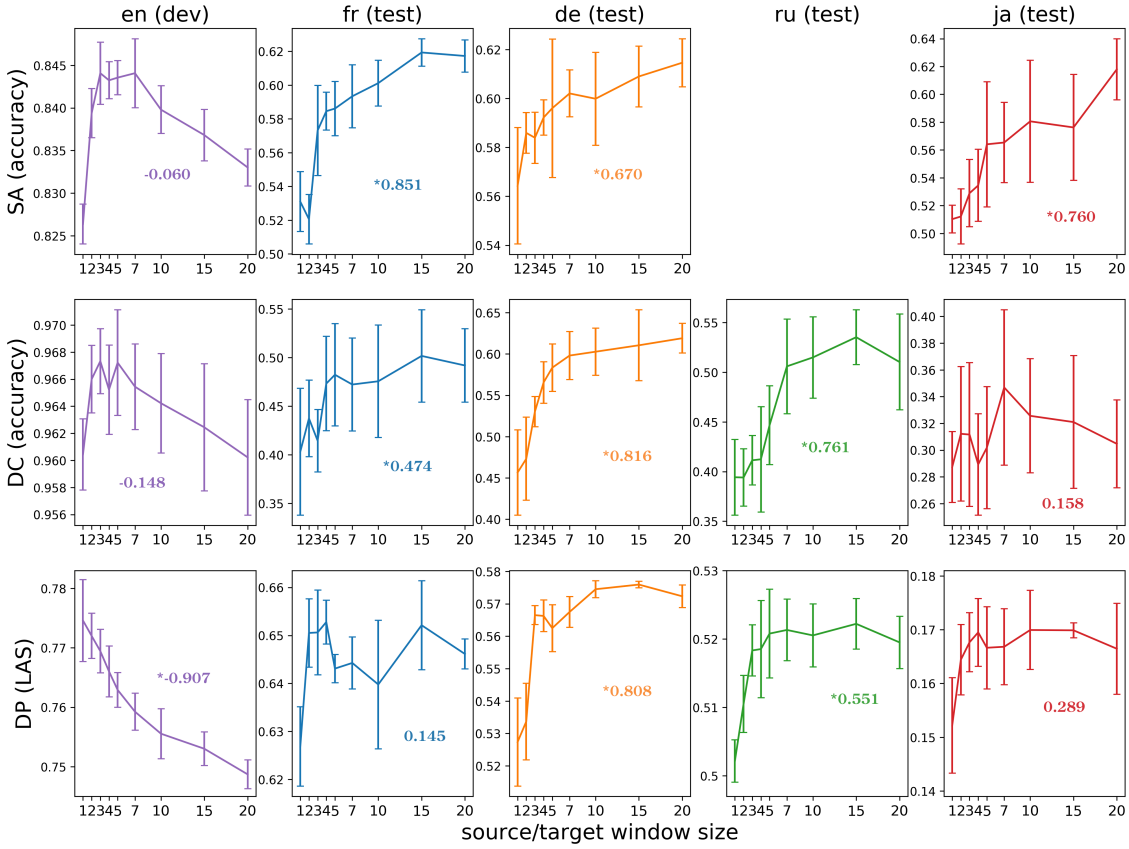

図 2.9: 下流タスクにおける多言語単語埋め込みの評価。それぞれのラベルは、SA: 極性分析、 DC: 文書分類、DP: 係り受け解析、を意味する。グラフ上の数値はスコアと窓サイズのスピアマンの順位相関係数。統計的に有意な相関はアスタリスクで示される ($p < 0.05$)。

イズと多言語言語埋め込みの関係について、入念に設計された実験を通じて知見を深めた。得られた知見を以下にまとめる。

- ソース言語にもターゲット言語にも大きな文脈窓を適用することで、それらの単語埋め込みをマッピングしやすくなる。単語の中でも特に名詞がうまくマッピングされ、またこの傾向はソース言語とターゲット言語のコーパスのドメインが異なる場合でも同様に観察される。
- 多言語単語埋め込みを言語横断の下流タスクで評価した場合、転移元タスクで一番性能を発揮する文脈窓は、必ずしも転移先でも良い性能を示すとは限らない。特に係り受け解析でこの傾向が顕著であり、転移元タスクでは最も小さい窓サイズ 1 が一番良い結果を示すが、転移先では最も低い性能を示す。これは、小さい窓サイズでは、単語埋め込み空間のマッピングが上手くいかず、その影響が大きく出ているのだと考えられる。

今後の研究課題としては、異なる単語によって異なる文脈窓を定義して単語埋め込みを学習する

モデルを考案するといったものがある。今回の実験結果では、名詞と形容詞は大きな窓サイズによりマッピングの精度が上がり、動詞と副詞についてはその傾向は薄かった。また、例えば内容語と機能語の区別を考えてみても、内容語はトピックが意味を決定するのに大きな手がかりとなり大きい窓サイズが有用であるが、機能語に関しては文法的な振る舞いが大事になり小さい窓サイズが向いている、という直感から、単語毎に適切な文脈窓を与えることは妥当なことであると思われる。

また、小さい窓サイズで訓練すると文法的な性質を持った単語埋め込みが得られるが、それらの複数言語間でのマッピングがしにくいという観察から、如何にして異なる言語間で共通の文法的性質を捉えた埋め込み空間を獲得するかということを考えるのも興味深い。

第3章　機械翻訳モデルが捉える言語類型論

3.1　序論

伝統的な機械翻訳と異なり、ニューラルネットワークに基づく機械翻訳 (Neural Machine Translation; NMT) は、1 つのモデルで複数言語ペアの翻訳を行う多言語機械翻訳 [3, 30] が容易に実現できる。多言語機械翻訳モデルは、各言語ペアについてそれぞれモデルを開発・訓練するのに比べて、容易に開発・維持管理ができる。これは、世界規模の Web サービスを提供する企業などにとって非常にメリットがある。また、今回の分析の対象となる多言語機械翻訳モデルは、通常の 2 言語機械翻訳モデルからモデル構造を変えることなく入出力データが多言語ペアに拡張されるだけであり、既存の機械翻訳用のフレームワークが流用できる。

このように、ニューラル多言語機械翻訳は近年注目を浴びている分野であるが、モデルが複数の言語に関する情報をどのように捉えているかというのはあまり明らかになっていない。先行研究では、ニューラル機械翻訳モデルの内部表現からは様々な言語学的情報が復元できることが知られている [31, 32, 33, 34, 35]。しかし、複数言語を扱う多言語機械翻訳モデルが、複数言語に共通する言語学的性質または異なる性質をどの程度捉えているのかは明らかになっていない。

本研究の目的は、多言語機械翻訳モデルが言語の普遍性と多様性をどの程度捉えているか、を明らかにすることである。具体的には、以下の問いに答えることを目的とする。

- モデルの各モジュールはどれだけ言語類型論的情報を含んでいるか？
- 言語類型論的素性の中でも、どのような素性を機械翻訳モデルはよく捉えているのか？

これらの問いを probing タスク [36, 37] の枠組みに則り明らかにする。Probing タスクでは、ニューラルモデルの内部表現を抽出し、その表現をから調べたい言語学的性質のラベルを予測する分類器を訓練する。そして、その分類器を評価データで評価することによって、入力とした内部表現がどれだけ、対象の言語学的性質を捉えているかを判定する。

実験の結果、機械翻訳モデルに内部表現は、入力に近い層のものは入力言語の類型論的特徴をよく捉え、出力に近い層のものは出力言語の類型論的特徴をよく捉えることが分かった。また、言語の類型論的特徴の中でも、語順に関するものをよく捉えることが分かった。本研究から得られる知見は、現在の多言語機械翻訳モデルの持つ限界を明らかにし、より良いモデルを考案する足がかりになる。

3.2 背景：ニューラル機械翻訳

機械翻訳システムの入力となる言語をソース言語と呼び、出力をターゲット言語と呼ぶ。機械翻訳では、ターゲット言語の文章を生成しなくてはならないが、自然言語の生成の仕組みは**言語モデル**の枠組みで説明されることが多い。本節ではニューラル機械翻訳システムを条件付き言語モデルとしてみなし、まずはその導入として言語モデルの説明をする。

3.2.1 言語モデル

言語モデルはその名の通り、人間の使用する自然言語を抽象的にモデル化したものである。言語を生成するためには、人間言語のどのような側面をモデル化したら良いだろうか？例えば、次の文章の続きを考えてみるとする。

今日はとても良い...

流暢な日本語話者であれば、上の文章に続くものが容易に思い浮かぶはずである（「... 天気です。」「... 日だった。」など）。ここから、文を生成するための能力は文脈が与えられた時に次に続く単語を予測する能力である、と考えることができる。これを踏まえた、本論での言語モデルの定義は以下のようなものになる。

ある言語における、文脈 c を与えられたときの、単語 $w \in V$ の出現確率 $p(w|c)$。

N-gram 言語モデル

最も単純な形の言語モデルは、文脈 c を直前の n 単語とする、N-gram 言語モデルである（式 3.1）。

$$p(w_t|w_{t-1}, ..., w_{t-n}) \tag{3.1}$$

この言語モデルを推定するためには、訓練コーパス内の単語列の頻度を数え上げれば良い。文の生成には、文頭から再帰的にモデルの確率にしたがってサンプリング、または貪欲に単語を選択していく。

しかし、この N-gram 言語モデルには重大な問題がある。データ・スパースネスの問題である。文脈の長さ n を大きくすればするほど、長い単位のイディオムや慣用的な言い回しを捉えることができ、生成文の流暢さに寄与する一方で、文脈の単語列の組み合わせ（$|V|^n$ の大きさ）が爆発的に増加してしまう。人間の産出する文章は非常に多様なので、その一部に過ぎない訓練コーパスでその言語における文法的な単語列を全てカバーできるわけではないのである。

この問題を緩和するために、訓練データ内に現れない文脈と単語の組にもわずかに確率を与えるスムージングの手法 [38] や、同様の意味を持つ単語を同じものとしてクラスタリングして扱う手法 [39]、単語列の確率が 0 の場合に、より小さい N-gram の確率を用いる back-off の手法 [40] が提案されてきた。しかし、これらはより良い性能を発揮するニューラルネットワークに基づいたモデルに取って代わられることとなる。

ニューラル N-gram 言語モデル

N-gram 言語モデルの根本的な問題は、単語を離散的な記号としてしか表現していなかったことに起因する。これでは、「犬」と「猫」は全く別の単語として扱われ、どちらも生物であることや、人間の愛玩動物として人気なものであることなどの意味的な共通点が捉えられない。また「走る」と「食べる」では、どちらも動詞であり、文中では主語をとり節を作ることが出来るといった文法的な共通点も見逃されてしまう。人間はこういった単語の共通点を認識しているために、新しい単語に数回触れただけで、それを幾千通りもの文脈で用いることが出来るのである。

ニューラル N-gram 言語モデルは、単語を連続値ベクトル $\mathbf{w} \in \mathbb{R}^d$ で表現する[17] ことで、単語間の関係を捉え高性能な言語モデルを実現する。単語の出現確率 $p(w_t|w_{t-1}, ..., w_{t-n})$ を求めるために、順伝播型ニューラルネットワーク (Feedforward Neural Network) のモデルに基づいて計算を行う。

まず、文脈語の単語ベクトル $\mathbf{w}_{t-1}, ..., \mathbf{w}_{t-n}$ の単語ベクトルから、ネットワークの隠れ層ベクトル $\mathbf{h}$ を以下のように計算する。

$$h = \tanh\left(\mathbf{b}_h + \sum_i \mathbf{H}_i \mathbf{w}_i\right) \tag{3.2}$$

ここで、$\mathbf{b}_h \in \mathbb{R}^{d_h}, \mathbf{H}_j \in \mathbb{R}^{d_h \times d}$ は d_h を隠れ層の次元数とするパラメータである。

隠れ層ベクトル $\mathbf{h}$ から、単語の確率分布スコア $\mathbf{s} \in \mathbb{R}^{|V|}$ を以下のように計算する。

$$\mathbf{s} = \mathbf{W}\mathbf{h} \tag{3.3}$$

ここで、$\mathbf{W} \in \mathbb{R}^{|V| \times d_h}$ はパラメータである。最終的な確率分布は、$\mathbf{s}$ にソフトマックス関数を適用することで得られる。

$$p(w_t|w_{t-1}, ..., w_{t-n}) = \frac{\exp(s_t)}{\sum_j^{|V|} \exp(s_j)} \tag{3.4}$$

RNN 言語モデル

上記のニューラル N-gram 言語モデルは、単純な N-gram 言語モデルに比べて、効率的に言語の N-gram 分布を捉えられるようになった。しかし、モデルの捉えられる依存関係は、依然として n 単語以内の距離の単語間に限られる。これは、実際の言語は、単語間に長距離の依存関係が存在するということを考えると好ましくない。

例えば、1 つには文法的な依存関係がある。以下の英語の文では、主語の *kangaroos* に依存して、動詞の *seem* の活用形が決定される。

> The **kangaroos** that are destroying the fences and eating the pasture **seem** to have escaped from mountain fire.

[17] この単語の連続値ベクトルは単語埋め込み (word embedding) とも呼ばれる（節 2.2.1 参照）。

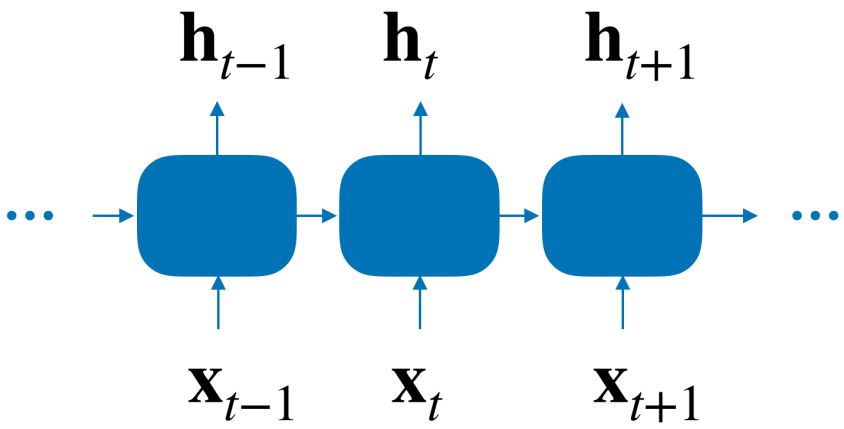

図 3.1: RNN の系列計算の模式図。

このような文法的な関係はどれだけ離れた単語間でも成り立ちうるものであるため、言語モデルは任意長の系列内の依存関係を捉えられることが望ましい。

また、語の選択選好 (selectional preferences) [41] も考慮する必要がある。選択選好とは、文法的に関係にある単語（目的語など）にどのような意味を持つ単語が現れやすいかということであり、言ってしまえば我々の持つ常識である。例えば、以下の 2 つの日本語文を比べてみると、前者の方が圧倒的に日本語に現れやすい文章であることがわかる。

> 私は**米**をおとbetter遠く離れた旅先で久しぶりに**食べた**。
> 私は**自転車**をおとという遠く離れた旅先で久しぶりに**食べた**。

これは我々の知識の中に「食べた」の目的語に現れやすい単語の知識を持っているためである。

その他には、どのような単語が現れやすいかを予測するためには、文のトピックやジャンルを考慮すれば、前の段落や文章の頭など遠く離れた単語も手がかりとなる。

こういった、長距離の単語の依存関係を捉えられるのが再帰型ニューラルネットワーク (recurrent neural network; RNN) に基づいた言語モデルである。RNN は入力を単語ベクトル列 $\mathbf{x}_1, ..., \mathbf{x}_L$ とすると、それぞれに対する隠れ層ベクトル $\mathbf{h}_t$ を以下のように計算していく。

$$\mathbf{h}_t = \begin{cases} \tanh\left(W_{xh}\mathbf{x}_t + W_{hh}\mathbf{h}_{t-1} + \mathbf{b}_h\right) & t \geq 1 \\ 0 & \text{otherwise} \end{cases} \tag{3.5}$$

$\mathbf{h}_0$ は隠れ層ベクトルの初期状態であり、普通は $\mathbf{0}$ ベクトルに初期化する。各タイムステップの $\mathbf{h}_t$ は前の時刻の隠れ層ベクトル $\mathbf{h}_{t-1}$ に依存しているため、時刻 t の隠れ層ベクトルは、再帰的に時刻 $t-1$ 以前の全ての系列の情報を含んでいることになる。計算の模式図を図 3.1 に示す。各

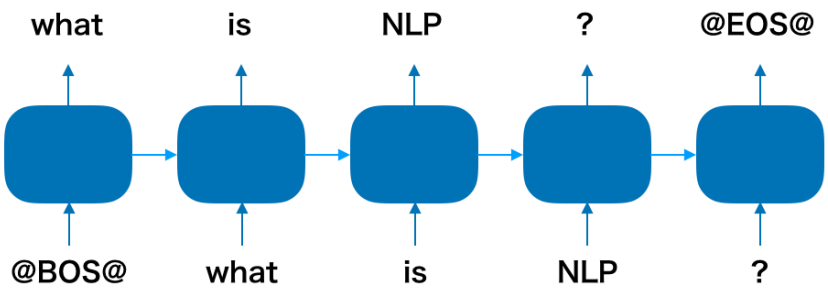

図 3.2: RNN 言語モデル。

単語の出力確率は、ニューラル N-gram 言語モデル同様に、隠れ層ベクトルを語彙サイズ分の次元を持つベクトルになるように写像し、ソフトマックス関数を適用する。

文頭の単語を予測するためには、文の開始を表す BOS (beginning of sentence) トークン を特殊トークンとして語彙に加え入力する。また、単語の予測を打ち切るために、文末を表す EOS (end of sentence) トークンも用いる。RNN 言語モデルを用いた文生成の模式図を図 3.2 に示す。

LSTM 言語モデル

RNN 言語モデルの学習には誤差逆伝播法 (back-propagation) が用いられる。しかし、長い系列を入力とした際に、誤差が過去の入力トークンまで伝播していかない勾配消失の問題が起こる。また、RNN では隠れ状態ベクトル $\mathbf{h}_t$ が系列の履歴を保持するのと、それ自体が次の単語を予測する 2 つの役割を担っており、これは必ずしも最適でない可能性がある。

これを解決するのが長・短期記憶ネットワーク (long short-term memory; LSTM) を用いた、言語モデルである。LSTM では時系列毎の状態を表すベクトルとして隠れ状態ベクトル $\mathbf{h}_t$ に加え、記憶セルベクトル $\mathbf{c}_t$ を持つ。LSTM では、入力ゲート $\mathbf{i}_t$、出力ゲート $\mathbf{o}_t$、および忘却ゲート $\mathbf{c}_t$ のユニットを持ち、これらで系列の情報をコントロールする。隠れ状態 $\mathbf{h}_t$ は以下のように計算される（図 3.3）。

$$\mathbf{u}_t = \tanh\left(\mathbf{W}_{xu}\mathbf{x}_t + \mathbf{W}_{hu}h_{t-1} + \mathbf{b}_u\right) \tag{3.6}$$

$$\mathbf{i}_t = \sigma\left(\mathbf{W}_{xi}\mathbf{x}_t + \mathbf{W}_{hi}h_{t-1} + \mathbf{b}_i\right) \tag{3.7}$$

$$\mathbf{o}_t = \sigma\left(\mathbf{W}_{xo}\mathbf{x}_t + \mathbf{W}_{ho}h_{t-1} + \mathbf{b}_o\right) \tag{3.8}$$

$$\mathbf{c}_t = \mathbf{i}_t \odot \mathbf{u}_t + \mathbf{c}_{t-1} \tag{3.9}$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh\left(\mathbf{c}_t\right) \tag{3.10}$$

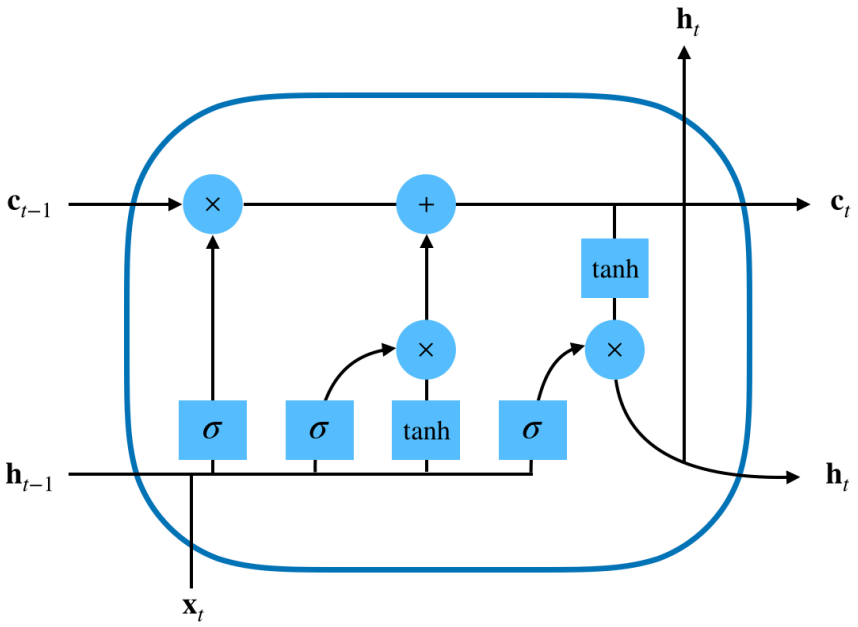

図 3.3: LSTM の内部計算の模式図。

以上に述べた RNN またはその発展系である LSTM は図 3.4 のように層を重ねることで、表現力を高めることができる。

3.2.2　エンコーダ・デコーダモデル

機械翻訳は条件付き言語モデルだとみなすことができる。つまり、ソース言語の入力文を条件として追加した言語モデルである（式 3.11）。

$$p(w_t|w_{t-1}, ..., w_{t-n}, c_{src}) \tag{3.11}$$

これは RNN に基づいて、シンプルに実装することができる。ソース言語の単語列を $x_1, ..., x_m$、ターゲット言語の単語列を $y_1, ..., y_n$ とする。この時、ソース言語の単語ベクトル列を RNN に入力し、最後の単語についての隠れ層ベクトル $\mathbf{h}_m$ を得る。この処理を担当する RNN を**エンコーダ**と呼ぶ。エンコーダからの表現は、ソース言語の入力文の条件に関する情報 c_{src} を担うものだと考えることが出来る。

次に、別の RNN を用意し、この隠れ層をエンコーダからの出力 $\mathbf{h}_m$ で初期化し、その後は RNN 言語モデル同様に、文の開始を表す特殊トークンを入力して文を生成していく。この文を生成するモジュールを**デコーダ**と呼ぶ（図 3.5）。

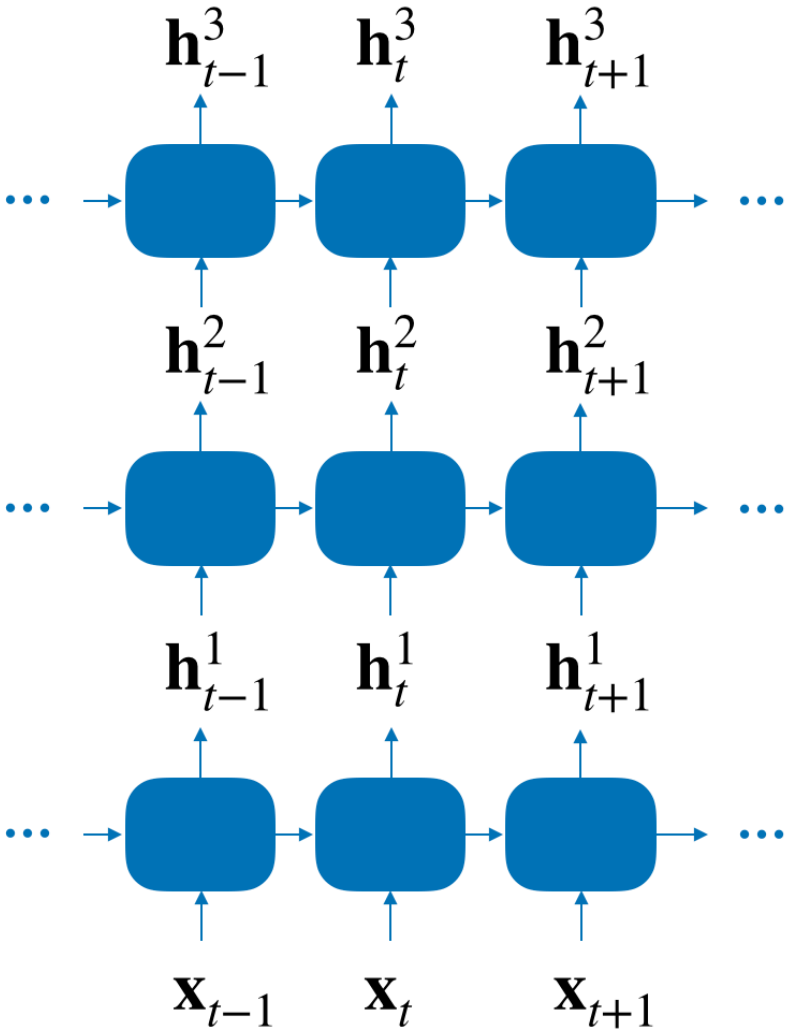

図 3.4: 複数層からなる RNN モデル。

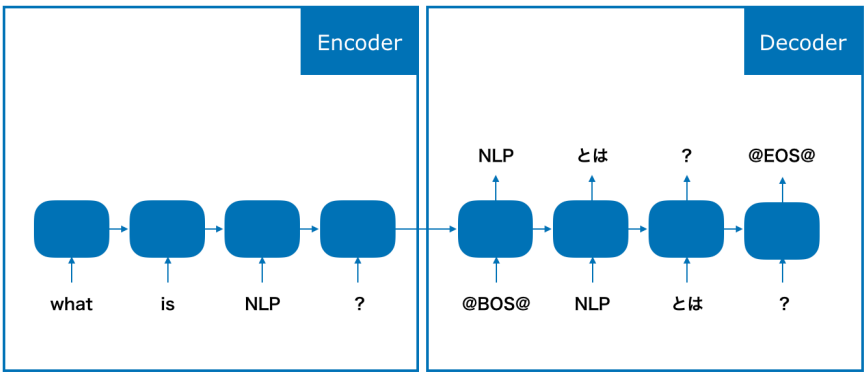

図 3.5: エンコーダ・デコーダモデル。

注意機構

エンコーダ・デコーダモデルはシンプルかつ強力なモデルであるが、ここにさらに翻訳に特化した機構を加えて、性能を著しく向上させたのが注意機構 (attention mechanism) である。翻訳では、入力系列と出力系列にトークン間の対応関係、つまり単語の翻訳関係があることが多い。この対応関係を注意機構は明示的に学習することが出来る。

注意機構の計算は、エンコーダの出力系列 $\mathbf{h}_1^{enc}, ..., \mathbf{h}_m^{enc}$ とデコーダからの出力 $\mathbf{h}_j^{dec}$ について計算される。まず、デコーダの出力 $\mathbf{h}_j^{dec}$ が、エンコーダの出力系列内のトークン $\mathbf{h}_i^{enc}$ にどのくらい関係するかのスコア a_{ij} を計算する。このスコアの計算には、順伝播型ニューラルネットワークを用いるもの [42] や、双線形モデルを使うもの [43] などがあるが、ここでは最も単純な内積を用いた場合を例とする（式 3.12）。

$$a_{ij} = \mathbf{h}_i^{enc} \cdot \mathbf{h}_j^{dec} \tag{3.12}$$

このスコアを出力の各系列について計算した後、ソフトマックス関数を用いて合計が 1 の重みとし、エンコーダ出力系列の重み和を計算する。

$$\hat{a}_{ij} = \frac{\exp(a_{ij})}{\sum_i \exp(a_{ij})} \tag{3.13}$$

$$\mathbf{h}_j^{attn} = \sum_i \hat{a}_{ij} \mathbf{h}_i^{enc} \tag{3.14}$$

最後に、この注意機構から出力 $\mathbf{h}_j^{attn}$ から j 番目の単語を予測する。

3.3 背景：Probing タスク

ニューラルネットワークに基づくモデルにより自然言語処理は大きな飛躍を見せた一方で、従来のパイプラインで逐次処理していく伝統的な言語処理にはあった解釈性が失われてしまった。例えば、伝統的な機械翻訳の仕組みでは、構文解析、単語のアラインメントの学習、単語の並び替えなど、様々な処理が独立したモジュールで実現されているため、翻訳の仕組みが人間にとって直感的に分かりやすく、また翻訳エラーがどの段階のミスによるものなのかが特定出来た。一方で、ニューラル機械翻訳での処理は個別具体的なモジュールに分かれているわけではなく、入力から出力まで全てニューラルネットワークによる end-to-end の仕組みで翻訳される。その内部処理は連続値ベクトルと行列の演算によって行われ、人間が演算の意味するところを解釈することは難しい。

近年ではニューラルネットワークの内部動作について **probing タスク**を通じて理解を深める試みが為されている。Probing タスクとは、ニューラルモデルから出力されたベクトル、または内部のベクトル表現から、入出力に関する言語学的な特徴を予測できるかどうかを調べるタスクである。一般的には、分類タスクとして以下のように定式化できる。分析の対象としたいベクトル表現を $\mathbf{s} \in \mathbb{R}^d$ とする。これは文の表現や、文中のトークンに対応する表現であることが多い。そのベクトル表現に含まれているかどうかを調べたい言語学特徴のラベルの集合を $\mathcal{L}$ とする。これは例えば文であれば文長を表すラベルであったり、トークンであれば単語の品詞の集合などがあり得る。これらのラベルをベクトル表現から予測する分類器 $C(\mathbf{s}) : \mathbb{R}^d \longmapsto \mathcal{L}$ を訓練し、評価データでの精度で、モデルからの表現がどれだけ言語学的な特徴を捉えているかを調べる。この分類器にはロジスティック回帰や、多層パーセプトロンなどが使われる。

機械翻訳モデルについては、その応用としての重要性から、様々な研究において probing タスクによる分析の対象とされてきた。今までの結果から、ニューラル機械翻訳モデルのエンコーダの内部表現は入力言語の形態論 [31, 32]、統語論 [33]、意味論 [34, 35] の情報を捉えていることが分かっている。本研究では、多言語機械翻訳モデルを対象に、その内部表現が異なる複数の言語の類型論的性質を捉えているかどうかを調査する。

3.4 実験設定

本研究での実験は、多言語機械翻訳モデルの訓練と、その内部表現の probing の 2 段階に分けることができる。

3.4.1 多言語機械翻訳モデルの訓練

コーパス

今回は多言語機械翻訳モデルが、言語の類型論的知識をどれだけ捉えているかを調べる実験であるため、訓練コーパスは以下の条件を満たすものが望ましい。

- なるべく多くの言語についてのデータが存在する。
- それら言語についてそれぞれが同一内容の文の翻訳になっている。

- 全ての言語について、データが同じ文数だけ存在する。

全ての言語について同一内容であることを保証することで、言語間のコーパスに記述される文の意味論的な違いを無くし、言語間で公平な条件でモデルを訓練できる。また、使用するデータ数が言語によって偏ってしまうと、モデルがデータ数の多い言語の性質を捉える方向に偏る恐れがある。以上の点を考慮して、訓練コーパスには Bible コーパス [44] を用いた。Bible コーパスは 100 言語の聖書の翻訳から成るコーパスである。しかし、全ての言語について聖書の全文が含まれているわけではないため、ほぼ全文の翻訳が存在する 58 の言語について共通の文章を抽出し実験に使用した。訓練、開発、テストデータのデータ数[18]の内訳を表 3.1 に示す。

訓練	開発	テスト
23,555	455	455

表 3.1: 聖書コーパスのデータスプリットの内訳。

開発データは翻訳モデル訓練の際の訓練早期打ち切り (early stopping) のために用い、テストデータは後段の probing タスクのために用いた。

多言語機械翻訳モデル

今回、機械翻訳モデルとして、アテンション付き LSTM エンコーダ・デコーダモデル [43] を分析の対象とした。単語埋め込みと隠れ層のサイズは 512、エンコーダ・デコーダの層の数は 3 、ドロップアウト率は 0.1 とした。訓練は Adam を最適化器として用いた。

モデルは、Jonson ら [3] と同様に、基本的な機械翻訳のアーキテクチャは 2 言語間翻訳と変えずに、入出力を多言語に拡張したものとした。英語以外の 57 の言語をソース言語としターゲット言語の英語へ翻訳する多対一 (many-to-one) の設定と、英語から 57 言語に翻訳する一対多 (one-to-many) の 2 種類のモデルを訓練した。入力文にソース言語を明示するタグは付加しない。また一対多の翻訳の設定では、出力文の先頭には、 BOS トークンの代わりに出力言語を指定する特殊トークンを付加した。

入力文は、言語非依存トークナイザーである `sentencepice` [45] での単語分割を行った。ソース言語については、全ての 57 のソース言語の文章から 32,000 のサイズを持つ語彙を学習し、ターゲット言語は英語のコーパスから 8,000 のサイズを持つ語彙を学習した。

3.4.2 Probing タスク

今回の probing タスクの目的は、機械翻訳モデルが翻訳を行う際に、ソース言語の類型論的性質を捉えているかどうかである。したがって、probing タスクの入力は、モデルにソース文を入力した時の各層の内部表現である。エンコーダ・デコーダの各層はトークン毎のベクトルとなっており、これを文のベクトルとしてまとめるために max-pooling を行った [37, 46]。

[18]Bible コーパスでは、通常のパラレルコーパスと異なり文単位でなく、聖書内の節単位で翻訳されている。したがって、1 つの翻訳ペアが複数文から成る場合もある。

言語類型論データベース

Probing タスクで分類器が予測する類型論素性は、言語類型論データベースの URIEL [47] から抽出した。URIEL は様々な言語学データベースから抽出した言語の類型論データをまとめたものである。使用したデータには 103 個の統語的素性と、28 の音韻的素性、158 の音素素性は全て 2 値素性として含まれる。今回は機械翻訳のタスクを通じて直接的に特徴を捉えることが出来ると考えられる統語的素性を対象に用い、また正負ラベルが余りにも偏っている素性（正または負ラベルが 10 以下のもの）は取り除いた。結果、実験で用いた統語的素性は 57 個となった。統語的素性の例を表 3.2 に示す。

素性		日本語	英語
SVO	SVO の基本語順を持つ	−	+
OBJECT_BEFORE_VERB	基本語順において、目的語が動詞の前に置かれる	+	−
PLURAL_WORD	複数形の単語を持つ	−	+
POLARQ_MARK_FINAL	yes/no 疑問文を表す分詞が文末に置かれる	+	−
NUMERAL_BEFORE_NOUN	数詞が名詞の前に置かれる	+	+

表 3.2: URIEL に含まれる統語的素性の例。

訓練・評価データ

Probing タスクの訓練・評価データは Bible コーパスのテストデータの 455 データから抽出される。今回、訓練データと評価データへの分割は言語毎に行われる。つまり、訓練データと評価データに共通して含まれるソース言語はない。これは、訓練データと評価データに同じ言語が存在すると、分類器は類型論的性質を予測するために「どの言語であるか」の情報を使えてしまうためである。タスクの制度は 10 分割交差検定で評価した。

ベースライン

Probing タスクの精度から、モデルが言語類型論の知識を獲得できているかどうかを解釈するために、ベースラインの精度と比較する必要がある。一般的なベースラインとしては、多数決 (Majority Vote) ベースライン（訓練データに最頻出のラベルを、評価データに対しても、予測として出力する）やランダムモデルベースライン（モデルの重みをランダムに初期化し、その出力で評価する）などが考えられる。今回は、多数決ベースラインに加え、より強力なベースラインとして Bag-of-Tokens 素性ベースラインを用いる。

Bag-of-Tokens 素性ベースラインは、文の入力のトークン列の ID に当たる次元に 1 を立て、その他の次元は 0 としたベクトルを入力とし、言語類型論的素性を予測する分類器を学習して得られるベースラインである。このベースラインの背後には、同じトークンを持つ言語は類型論的性質も似る傾向にあり、分類器はトークンと類型論的素性を結びつけるだけである程度の精度を達成できるのではないかという直感がある。例えば、ヨーロッパの諸言語はアルファベットを文字

	enc_emb	enc_1	enc_2	enc_3	dec_emb	dec_1	dec_2	dec_3
Majority vote (MV)	69.89%							
Bag-of-Tokens (BoT)	74.43%							
RNN (many-to-one)	79.10%	78.43%	75.21%	74.32%	69.79%	69.92%	69.90%	69.84%
RNN (one-to-many)	-	-	-	-	77.77%	79.09%	80.07%	80.98%

表 3.3: 類型論素性予測のタスクのモデル各層の内部表現の精度。値は各類型論素性予測についての 2 値分類タスクの平均。*enc* はエンコーダ、*dec* はデコーダ、*emb* は埋め込み層を表す。

として使うが、入力トークンがアルファベットから成るという情報だけから、ヨーロッパの言語に多い類型論的性質をある程度予測できてしまう。今回の研究では、多言語機械翻訳モデルが言語類型論とトークンの表層的な相関関係以上のものを捉えているかどうかを調べるために、この Bag-of-Tokens ベースラインとも比較する。

3.5 実験結果

このセクションは、多言語機械翻訳モデルがどの程度言語の文法についての普遍性を捉えられているのかを、probing タスクの結果から示すことを目的とする。表 3.3 にモデルの各層の類型論素性予測のタスクの精度をベースラインと共に示す。表の精度は 56 の類型論素性予測の 2 値分類タスクでのスコアの平均である。

3.5.1 各層に含まれる情報

多対一モデルのエンコーダの各層に含まれる情報

先行研究では、機械翻訳モデルの各層に含まれている情報は異なることが示されている。Belinkov ら [31] の実験結果では、機械翻訳モデルのエンコーダの各層のベクトルから入力トークンの品詞を予測するタスクにおいて、エンコーダの第 1 層目のスコアが最も高く、それ以降は層が上がるにつれてスコアが低くなる傾向が観察されている。これは直感的には次のように解釈出来る。

> エンコーダの上の層では単語の表層的な情報が失われ、代わりに抽象的な意味の情報がより含まれている。また、品詞は個別の単語で決まるだけでなく、文の中で他の単語とどのような関係にあるかということによっても決まるため、文全体の情報が考慮されていない単語埋め込み層の表現よりも、エンコーダの第一層目の表現の方が品詞情報をうまく捉えられている。

実験の前には、この傾向が今回の類型論予測のタスクについても成立すると考えていた。つまり、エンコーダの上の層に行くにつれて、言語個別の情報は捨象され抽象的な意味を捉えるのでタスクの精度は低くなっていく。また、単語埋め込み層の表現は語順などの情報が含まれておらず言語の類型論を予測するための情報が足りずエンコーダの第 1 層などと比べるとタスクの精度は低く

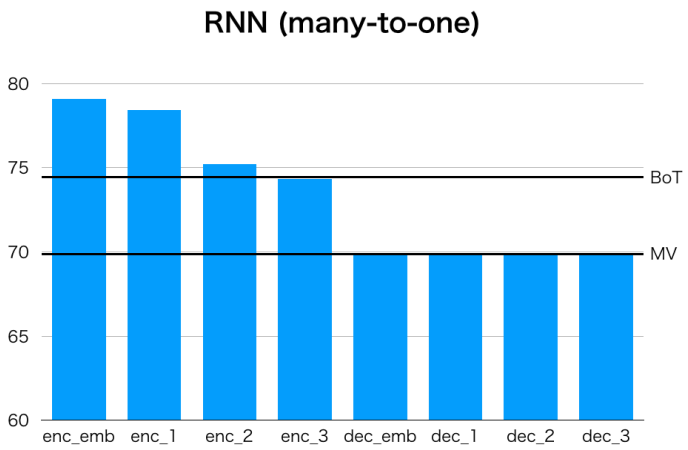

図 3.6: 多対一 RNN 多言語機械翻訳モデルの各層の類型論予測タスクの性能。

なると予測していた。しかし、この予想とは若干異なる傾向を実験結果は示している。表 3.3 から多対一のモデルのスコアをまとめたグラフを図 3.6 に示す。

実際の結果と予測を比較すると、エンコーダの上の層ではスコアが落ちるという点では合致しているが、単語埋め込み層のスコアが一番高いという点では予測に反している。これは、モデルのエンコーダはトークンの埋め込み表現に言語の基本語順などの情報を付加するような演算をしていないことを示唆している。

多対一モデルのデコーダの各層に含まれる情報

エンコーダとデコーダのスコアを比較すると、エンコーダ側の表現はどれも BoT ベースラインを超えるか同様のスコアを示しているが、デコーダ側の表現は多数決ベースラインを上回る結果を見せていない。これは、デコーダがターゲット言語の文を出力するときの内部表現は、入力言語の情報を全く含んでいないことを意味している。出力言語のデコードは入力言語に寄らず同じ挙動で行えることが効率性の観点から言えば好ましく、今回はエンコーダ・デコーダモデルはこれを達成できていると言える。

一対多モデルのデコーダの各層に含まれる情報

次に一対多モデルにおいて、デコーダの内部表現がどの程度出力言語の類型論的素性が予測できるかの結果を図 3.7 にまとめる。今回のモデルでは出力言語はデコーダの最初の入力トークンで指定し、エンコーダの表現には出力言語の情報は一切含まれていないため、デコーダの表現からの結果のみ示す。

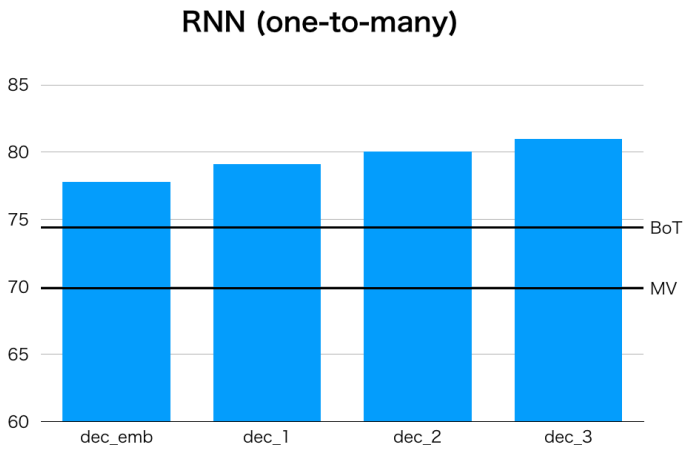

図 3.7: 一対多 RNN 多言語機械翻訳モデルの各層の類型論予測タスクの性能。

結果をみると、デコーダの層が上がるにつれて類型論予測のタスクの性能が上がっている。これは、多対一モデルの結果と合わせると、全体的に単語の表層に近い位置にある表現ほどスコアが高くなる傾向にあると言える。

3.5.2 定性的分析

どのような類型論的素性が予測しやすいのかという傾向を調べるために、モデルととベースラインとのスコアの差順に類型論的素性を並べ、上位 5 件と下位 5 件のものを抽出した。多対一 RNN 多言語機械翻訳モデルから一番平均スコアの高いエンコーダ単語埋め込み層からの結果を表 3.4 に示す。上位を語順に関する素性が占めており、下位は特定の文法事項が存在するか否かに関する素性が占めている。

この傾向は層が上がっても、同様かどうかを調べるために、エンコーダ第 3 層目からの結果を表 3.5 に示す。上位を語順に関する素性が独占し、下位はそれ以外という傾向は単語埋め込み層の結果と変わらない。

一対多モデルにおいて、デコーダの表現からターゲット言語の類型論的素性を予測するタスクにおいても、エンコーダ同様に、語順に関する素性が予測しやすいという傾向が見られた（表 3.6、）。

以上をまとめると、多言語機械翻訳モデルの内部表現は言語の類型論の中でも語順に関する特徴をよく捉えているということが言える。しかし注意すべきは、この結果は文の表現から言語の類型論的特徴を予測した際のものだ、ということである。基本語順といった特徴は様々な文にそのまま表れていることが多いが、そうでないもの、例えば `INDEFINITE_WORD`（不定冠詞を含むか）や

Feature	BoT	Model	Gain
SOV	66.61	87.31	20.70
OBJECT_BEFORE_VERB	61.95	82.62	20.67
SVO	69.35	87.91	18.56
NEGATIVE_WORD_BEFORE_OBJECT	63.33	80.48	17.15
OBJECT_AFTER_VERB	80.94	96.07	15.12
POLARQ_MARK_INITIAL	78.08	74.69	-3.39
FUTURE_AFFIX	76.27	72.34	-3.92
POLARQ_WORD	72.80	68.00	-4.81
PERFECTIVE_VS_IMPERFECTIVE_MARK	62.36	57.04	-5.32
INDEFINITE_WORD	70.35	64.02	-6.33

表 3.4: 多対一 RNN 多言語機械翻訳モデルのエンコーダ単語埋め込み層から抽出された素性の類型論的素性予測タスクのベースラインからのスコア差の上位／下位 5 件。

Feature	BoT	Model	Gain
OBJECT_BEFORE_VERB	61.95	77.34	15.39
NEGATIVE_WORD_BEFORE_OBJECT	63.33	77.28	13.95
NEGATIVE_WORD_ADJACENT_BEFORE_VERB	55.55	69.43	13.88
SOV	66.61	79.06	12.45
POSSESSOR_AFTER_NOUN	69.48	80.67	11.19
ADJECTIVE_AFTER_NOUN	70.67	58.36	-12.32
PROSUBJECT_WORD	70.69	57.63	-13.07
CASE_SUFFIX	71.75	58.01	-13.75
PROSUBJECT_AFFIX	69.57	55.74	-13.83
TEND_DEPMARK	71.11	55.19	-15.93

表 3.5: 多対一 RNN 多言語機械翻訳モデルのエンコーダ第 3 層から抽出された素性の類型論的素性予測タスクのベースラインからのスコア差の上位／下位 5 件。

Feature	BoT	Model	Gain
OBJECT_BEFORE_VERB	61.95	61.48	24.66
SOV	66.61	64.70	24.30
ADPOSITION_AFTER_NOUN	69.22	66.65	21.58
POSSESSOR_AFTER_NOUN	69.48	68.78	20.63
POSSESSOR_BEFORE_NOUN	68.92	58.94	18.17
OBJECT_HEADMARK	79.12	82.93	-3.81
NEGATIVE_WORD_AFTER_VERB	86.58	77.54	-3.84
POLARQ_MARK_INITIAL	78.08	79.55	-4.79
PROSUBJECT_WORD	70.69	54.94	-6.37
INDEFINITE_WORD	70.35	50.07	-12.39

表 3.6: 一対多 RNN 多言語機械翻訳モデルのデコーダ第 3 層目から抽出された素性の類型論的素性予測タスクのベースラインからのスコア差の上位／下位 5 件。

Feature	BoT	Model	Gain
SOV	66.61	87.82	21.21
OBJECT_BEFORE_VERB	61.95	79.13	17.18
ADPOSITION_AFTER_NOUN	69.22	85.63	16.42
OXV	79.08	92.56	13.48
POSSESSOR_AFTER_NOUN	69.48	82.58	13.10
COMITATIVE_VS_INSTRUMENTAL_MARK	73.74	65.8	-7.94
POLARQ_MARK_INITIAL	78.08	69.13	-8.95
INDEFINITE_WORD	70.35	61.22	-9.13
OBJECT_HEADMARK	79.12	68.31	-10.81
NEGATIVE_WORD_AFTER_VERB	86.58	72.42	-14.15

表 3.7: 一対多 RNN 多言語機械翻訳モデルのデコーダ単語埋め込み層から抽出された素性の類型論的素性予測タスクのベースラインからのスコア差の上位／下位 5 件。

POLARQ_MARK_INITIAL（yes/no 疑問文を表す目印が文頭に来るか）といった特徴は、その言語の中でも一部の文にしか表れない特徴である。したがって、今回の文の表現から言語自体の特徴を予測するという実験の枠組みでは、そもそも確かめられていない可能性がある。理想的には、文毎にどのような通言語的な文法的特徴が含まれているか注釈がついたデータを用いるのが良く、例えば言語間共通の枠組みで様々な言語の文について統語的な注釈をした Universal Dependencies のデータなどを使った実験などが考えられるが、言語毎のデータ量や種類の偏りや、注釈のカバレッジの問題などがある。より良い実験の枠組みの考案は今後の研究の課題としたい。

3.6 結論

本研究では多言語機械翻訳モデルの内部表現が、異なる言語の言語類型論についてどの程度の情報を含んでいるかということを、probing タスクの枠組みから調査した。その結果、以下のような観察結果が得られた。

- モデルの層毎に含まれる情報の傾向を調べると、ソース言語の類型論は入力に一番近いエンコーダの単語埋め込み層での表現が最もよく捉えており、反対にターゲット言語の情報は出力に一番近いデコーダの一番上の層がよく捉えている、という傾向が見られた。
- 予測する類型論的素性毎に見ると、語順に関する素性がそうでないものに比べ予測しやすいことが分かった。

今後の研究課題としては、各言語の類型論的素性が翻訳の品質にどの程度影響を与えるのか、ということを調査する方向が考えられる。より具体的に言えば、 1 つには、ソース言語とターゲット言語の類型論的特徴が翻訳品質に与える影響がある。例えば、英語とのフランス語は類型論的に似ているために、英語と日本語の翻訳に比べて機械翻訳の品質は高い。また、入出力言語の個別の類型論的特徴による影響もある。特殊な活用を持つ言語（名詞の性の活用や、有生性による活用など）などのへの翻訳を考えるのも、既存の翻訳システムの課題・改善点を発見するのに有用だと思われる。

第4章　おわりに

本論文では、自然言語処理の分野で特に重要な位置を占める技術である単語埋め込みと機械翻訳について、それらの多言語拡張モデルについて分析を行った。

単語埋め込みと文脈窓の関係に関して調べた結果からは、トピック的な意味を捉えた埋め込み空間は異なる言語間でも共有でき、文法的な性質を捉えたものは難しいということが分かった。既存のマッピングベースでは文法的な性質を捉えた単語埋め込みの知識を言語間でうまく転移できないことも明らかになった。多言語機械翻訳モデルが入出力言語の類型論的特徴をどの程度捉えているか調べた研究では、モデルの内部表現は言語類型論をベースラインを上回る結果で捉えていることがわかり、特に語順に関する類型論的特徴をよく捉えていることが分かった。これらの結果はそれぞれ、異なる言語間で文法を共有することの難しさと可能性を示していると言える。

以上を踏まえた今後の展望としては、複数言語で共通するもの（単語の意味、意味役割、一般的な文法構造など）と大きく異なるもの（細かい文法事項）を考慮したモデル化を検討することが考えられる。モデルが多言語間で知識を共有するのを促進しよりパラメータ効率の良いモデルが出来る可能性がある他、共通部分と言語特有のモジュールを分けて構成することでモデルの解釈性も向上させることが期待出来る。これを実現する方法としては、言語普遍の意味を捉えた訓練信号として画像を用いることや、Universal Dependencies といった言語普遍の文法モデルを利用する、などの方向が考えられる。

参考文献

[1] Emily Bender. *Linguistic Fundamentals for Natural Language Processing: 100 Essentials from Morphology and Syntax*. Morgan & Claypool Publishers, 2013.

[2] Daniel Bikel and Imed Zitouni. *Multilingual Natural Language Processing Applications: From Theory to Practice*. IBM Press, 2012.

[3] Melvin Johnson, Mike Schuster, Quoc V Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viegas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. Google's Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation. *Transactions of the Association for Computational Linguistics*, Vol. 5, pp. 339–351, 2017.

[4] Phoebe Mulcaire, Jungo Kasai, and Noah A. Smith. Polyglot contextual representations improve crosslingual transfer. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 3912–3918, Minneapolis, Minnesota, 2019. Association for Computational Linguistics.

[5] Telmo Pires, Eva Schlinger, and Dan Garrette. How multilingual is multilingual BERT? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4996–5001, Florence, Italy, 2019. Association for Computational Linguistics.

[6] Mikel Artetxe, Gorka Labaka, and Eneko Agirre. Learning principled bilingual mappings of word embeddings while preserving monolingual invariance. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 2289–2294, Austin, Texas, 2016. Association for Computational Linguistics.

[7] Tomas Mikolov, Quoc V. Le, and Ilya Sutskever. Exploiting similarities among languages for machine translation. *arXiv.org*, 2013.

[8] Anders Søgaard, Sebastian Ruder, and Ivan Vulić. On the Limitations of Unsupervised Bilingual Dictionary Induction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 778–788, Melbourne, Australia, 2018. Association for Computational Linguistics.

[9] Peter D. Turney and Patrick Pantel. From Frequency to Meaning: Vector Space Models of Semantics. *Journal of Artificial Intelligence Research*, Vol. 37, No. 1, pp. 141–188, 2010.

[10] Marco Baroni, Georgiana Dinu, and Germán Kruszewski. Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 238–247, Baltimore, Maryland, 2014. Association for Computational Linguistics.

[11] Omer Levy and Yoav Goldberg. Dependency-Based Word Embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 302–308, Baltimore, Maryland, 2014. Association for Computational Linguistics.

[12] John Rupert Firth. A synopsis of linguistic theory. In *Studies in Linguistic Analysis*, pp. 1930–1955. Oxford, 1957.

[13] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient Estimation of Word Representations in Vector Space. In *Proceedings of the International Conference on Learning Representations*, Arizona, USA, 2013.

[14] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pp. 3111–3119. Curran Associates, Inc., 2013.

[15] Sebastian Ruder, Ivan Vulić, and Anders Søgaard. A Survey of Cross-lingual Word Embedding Models. *Journal of Artificial Intelligence Research*, Vol. 65, No. 1, pp. 569–630, 2019.

[16] Chao Xing, Dong Wang, Chao Liu, and Yiye Lin. Normalized Word Embedding and Orthogonal Transform for Bilingual Word Translation. In *Human Language Technologies: The 2015 Annual Conference of the North American Chapter of the ACL*, pp. 1006–1011, Stroudsburg, PA, USA, 2015. Association for Computational Linguistics.

[17] Bofang Li, Tao Liu, Zhe Zhao, Buzhou Tang, Aleksandr Drozd, Anna Rogers, and Xiaoyong Du. Investigating Different Syntactic Context Types and Context Representations for Learning Word Embeddings. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 2421–2431, Copenhagen, Denmark, 2017. Association for Computational Linguistics.

[18] Omer Levy, Yoav Goldberg, and Ido Dagan. Improving Distributional Similarity with Lessons Learned from Word Embeddings. *Transactions of the Association for Computational Linguistics*, Vol. 3, pp. 211–225, 2015.

[19] Goran Glavaš, Robert Litschko, Sebastian Ruder, and Ivan Vulić. How to (Properly) Evaluate Cross-Lingual Word Embeddings: On Strong Baselines, Comparative Analyses, and

Some Misconceptions. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 710–721, Florence, Italy, 2019. Association for Computational Linguistics.

[20] Peter Prettenhofer and Benno Stein. Cross-Language Text Classification Using Structural Correspondence Learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pp. 1118–1127, Uppsala, Sweden, 2010. Association for Computational Linguistics.

[21] Yoon Kim. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1746–1751, Doha, Qatar, 2014. Association for Computational Linguistics.

[22] Holger Schwenk and Xian Li. A Corpus for Multilingual Document Classification in Eight Languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, 2018. European Language Resources Association (ELRA).

[23] Natalia Silveira, Timothy Dozat, Marie-Catherine de Marneffe, Samuel Bowman, Miriam Connor, John Bauer, and Chris Manning. A Gold Standard Dependency Corpus for English. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pp. 2897–2904, Reykjavik, Iceland, 2014. European Language Resources Association (ELRA).

[24] Timothy Dozat and Christopher D Manning. Deep Biafine Attention for Neural Dependency Parsing. In *Proceedings of the International Conference on Learning Representations*, 2017.

[25] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics*, Vol. 5, pp. 135–146, 2017.

[26] Guillaume Lample, Alexis Conneau, Marc Aurelio Ranzato, Ludovic Denoyer, and Herve Jegou. Word Translation without Parallel Data. In *Proceedings of the International Conference on Learning Representations*, Vancouver, Canada, 2018.

[27] Takashi Wada, Tomoharu Iwata, and Yuji Matsumoto. Unsupervised multilingual word embedding with limited resources using neural language models. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 3113–3124, Florence, Italy, 2019. Association for Computational Linguistics.

[28] Fabienne Braune, Viktor Hangya, Tobias Eder, and Alexander Fraser. Evaluating bilingual word embeddings on the long tail. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pp. 188–193, New Orleans, Louisiana, 2018. Association for Computational Linguistics.

[29] Paula Czarnowska, Sebastian Ruder, Edouard Grave, Ryan Cotterell, and Ann Copestake. Don't Forget the Long Tail! A Comprehensive Analysis of Morphological Generalization in Bilingual Lexicon Induction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 974–983, Hong Kong, China, 2019. Association for Computational Linguistics.

[30] Thanh-Le Ha, Jan Niehues, and Alex Waibel. Toward Multilingual Neural Machine Translation with Universal Encoder and Decoder. *arXiv.org*, 2016.

[31] Yonatan Belinkov, Nadir Durrani, Fahim Dalvi, Hassan Sajjad, and James Glass. What do Neural Machine Translation Models Learn about Morphology? In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2017.

[32] Arianna Bisazza and Clara Tump. The Lazy Encoder: A Fine-Grained Analysis of the Role of Morphology in Neural Machine Translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2018.

[33] Xing Shi, Inkit Padhi, and Kevin Knight. Does String-Based Neural MT Learn Source Syntax? In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2016.

[34] Yonatan Belinkov, Lluís Màrquez, Hassan Sajjad, Nadir Durrani, Fahim Dalvi, and James Glass. Evaluating Layers of Representation in Neural Machine Translation on Part-of-Speech and Semantic Tagging Tasks. In *Proceedings of the International Joint Conference on Natural Language Processing*, 2017.

[35] Adam Poliak, Yonatan Belinkov, James Glass, and Benjamin Van Durme. On the Evaluation of Semantic Phenomena in Neural Machine Translation Using Natural Language Inference. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2018.

[36] Yossi Adi, Einat Kermany, Yonatan Belinkov, Ofer Lavi, and Yoav Goldberg. Fine-grained Analysis of Sentence Embeddings Using Auxiliary Predicition Tasks. In *Proceedings of the International Conference on Learning Representations*, 2017.

[37] Alexis Conneau, German Kruszewski, Guillaume Lample, Loic Barrault, and Marco Baroni. What you can cram into a single vector: Probing sentence embeddings for linguistic properties. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2018.

[38] Stanley F. Chen and Joshua Goodman. An empirical study of smoothing techniques for language modeling. In *34th Annual Meeting of the Association for Computational Linguistics*, pp. 310–318, Santa Cruz, California, USA, 1996. Association for Computational Linguistics.

[39] Reinhard Kneser and Jochen Peters. Semantic clustering for adaptive language modeling. In *1997 IEEE International Conference on Acoustics, Speech, and Signal Processing*, Vol. 2, pp. 779–782, 1997.

[40] Slava M. Katz. Estimation of probabilities from sparse data for the language model component of a speech recognizer. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. 35, No. 3, pp. 400–401, 1987.

[41] Philip Resnik. Selectional preference and sense disambiguation. In *Tagging Text with Lexical Semantics: Why, What, and How?*, 1997.

[42] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *Proceedings of the International Conference on Learning Representations*, 2015.

[43] Thang Luong, Hieu Pham, and Christopher D. Manning. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 1412–1421, Lisbon, Portugal, 2015. Association for Computational Linguistics.

[44] Christos Christodouloupoulos and Mark Steedman. A massively parallel corpus: the Bible in 100 languages. *Language Resources and Evaluation*, Vol. 49, No. 2, pp. 375–395, 2015.

[45] Taku Kudo. Subword Regularization: Improving Neural Network Translation Models with Multiple Subword Candidates. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2018.

[46] Alexis Conneau, Douwe Kiela, Holger Schwenk, Lo ıc Barrault, and Antoine Bordes. Supervised Learning of Universal Sentence Representations from Natural Language Inference Data. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2017.

[47] Patrick Littell, David Mortensen, Ke Lin, Katherine Kairis, Carlisle Turner, and Lori Levin. URIEL and lang2vec: Representing languages as typological, geographical, and phylogenetic vectors. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics*, 2017.

発表文献

本研究に関する発表文献

査読付き会議論文

1. What do Multilingual Neural Machine Translation Models Learn about Typology?. Ryokan Ri, and Yoshimasa Tsuruoka. The First Workshop on Typology for Polyglot NLP (TyP-NLP 2019). https://typology-and-nlp.github.io/2019/assets/2019/papers/6.pdf

その他の発表文献

査読付き会議論文

1. Designing the Business Conversation Corpus. Matīss Rikters, Ryokan Ri, Tong Li and Toshiaki Nakazawa. The 6th Workshop on Asian Translation (WAT 2019). https://www.aclweb.org/anthology/D19-5204.pdf

査読なし会議

1. 文脈情報を考慮した日英ニューラル機械翻訳. 李 凌寒, 中澤 敏明, 鶴岡 慶雅. 言語処理学会 第 25 回年次大会 (2019). http://www.anlp.jp/proceedings/annual_meeting/2019/pdf_dir/A2-2.pdf

謝辞

本研究を進めるにあたって、大変多くの方にお世話になりました。

指導教員である鶴岡慶雅教授には、修士の 2 年間に渡って様々な方面でお世話になりました。研究をこれから本格的に始めるぞ、という時期には右も左も分からず手当たり次第に考えたものをミーティングで話したり、先生に相談しに参りましたが、その度に率直な意見やアドバイスをくださって非常に参考になりました。だんだん先生の反応で研究ネタの筋が良いのか悪いのか、判定できるようになってきた気がします。また、先生が研究室のために計算機環境を整えてくれたおかげで気兼ねせずに実験を回せ、とても快適な研究生活を送ることが出来ました。論文を書いたときは、締め切り直前にも関わらず添削して下さり非常に有り難かったです。数々のご指導ありがとうございました、感謝を申し上げます。

特任講師の中澤敏明先生には、私が NICT プロジェクトに参加したことを始めとして、色んなところに連れて行ってくださったり、多くの人と知り合う貴重な機会を提供して頂きました。おかげでこの 2 年間研究室の中にとどまらず、外の世界も伺い知ることが出来ました。

もう卒業してしまわれた 1 つ上の先輩である河村圭悟さんと水谷陽太さんには色んなことを教わりました。python プログラムの書き方、深層学習計算におけるテクニック、生協の使い方、研究室付近の美味しいお店などなど、どれもとても役に立っています。またミーティングでの研究に対する積極的な姿勢には多くの見習う点がありました。

同じ分野の大先輩である橋本和真さんと江里口瑛子さんとは、幸運ながらも研究室に在籍する期間が短い間ではありましたが重なったため、いろいろ相談させていただくことが出来て大変嬉しかったです。研究室に遊びに来てくださった際や、学会などでも近況の話を聞いて刺激を受けました。

特別研究員の Matīss Rikters さんとは初めての人と論文を執筆をすることを通じて多くのことを学ばせていただきました。また、国際学会にもよく一緒に参加しましたが、移動中でもたくさんおしゃべりをしてくれて楽しく過ごすことが出来ました。もう少し私も英語を流暢に話せるよう頑張ります。

同期の皆さんとはたまにラウンジでご飯を食べたり、他愛のないおしゃべりをしたりと、日常生活の面で大変お世話になりました。特に安井豪くんとは、モデルの細かい実装の話や研究についての議論を通じて、非常に刺激的な時間を過ごすことが出来ました。宮崎広夢くんとはよくコーヒーを囲みながらおしゃべりして楽しかったです。卒論生の西川荘介くんは研究テーマが非常に近かったのもあり、よく興味深い研究の話を聞かせてくれました。

作業する際はよく大学近くのカフェ・ベローチェに行っていました。カフェ・ベローチェは安価なコーヒーと快適な作業空間を提供してくれました。思い返すと卒論執筆の際にも実家の近くにあるカフェ・ベローチェに缶詰になっていましたので、中々長い付き合いです。おそらく博論もカフェ・ベローチェで書くと思います。

最後に、私の考えを尊重してくれ自由に好きなことをやらせてくれているに留まらず、生活面でのサポートをしてくれている家族にはいくら感謝してもしきれません。今の私があるのはあなたたちのおかげです。ありがとうございます。