学修番号 15890521

修士論文

レシピ文書の日英機械翻訳

佐藤　貴之

2017 年 2 月 1 日

首都大学東京大学院
システムデザイン研究科 情報通信システム学域

佐藤　貴之

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）

石川 博 教授　（副指導教員）

レシピ文書の日英機械翻訳*

佐藤　貴之

概要

近年，インターネット上で取得可能なレシピが増加している．例えば，日本の料理レシピサービスであるクックパッドでは 250 万品以上のレシピが取得できる（数字は 2016 年 9 月のもの）．同様に，アメリカの料理レシピサービスである Yummly でも 100 万品以上のレシピが取得できる．取得可能なレシピが増加するにつれ，これらに関する研究も増加している．これまでに研究されてきたトピックとしては，例えば，レシピの解析や検索，要約，推薦などがある．レシピは文法が単純であるが，レシピ特有の単語や表現によって，その解析が難しいという問題がある．そのため，専用の辞書構築や特有のアノテーションなども研究されている．レシピに関する研究が増加する中，これまでに研究されていないトピックとして機械翻訳がある．食文化の多様化とともにレシピの需要も国際的に拡大しており，特に日本食は健康にも良いとされることから，日本以外でも需要が大きい．機械翻訳で日本語のレシピを他言語に翻訳することで，多くの人がそれらを利用できるようになると思われる．そこで，本研究ではレシピ翻訳の現状と課題を確認するため，機械翻訳でレシピを翻訳し，その誤りを分析する．翻訳対象には，16,280 レシピから構成される日英対訳コーパスを使用する．各レシピは，クックパッドに投稿された日本語レシピを英語に翻訳したものであり，タイトル，材料，手順のフィールドから構成され，それぞれが異なった文体で書かれている．翻訳手法として，フレーズベース統計的機械翻訳とニューラル機械翻訳を使用し，日本語のレシピを英語に翻訳する．フレーズベース統計的機械翻訳は，1 から数単語をフレーズとみなし，フレーズごとに翻訳したのち，得られた各フレーズの訳出を並べ替えることで翻訳を行う．ニューラル機械翻訳は，ニューラルネットワークによって，入力された単語列をベクトルに変換し，これをもとに単語列を出力することで翻訳を行う．どちらの翻訳

*首都大学東京大学院 システムデザイン研究科 情報通信システム学域 修士論文, 学修番号 15890521, 2017 年 2 月 1 日.

手法も，現在，インターネット上の機械翻訳サービスのベースとして用いられているものである．翻訳誤りは，QTLaunchPad の Multidimensional Quality Metrics（以下 MQM）を参考に，各翻訳手法の訳出に対してブラックボックス分析を行うことで分類する．MQM における誤り分類は，妥当性と流暢性の二種類に大別される．妥当性は入力文と翻訳結果の整合性の度合いを測る分析の観点であり，流暢性は翻訳結果の語法や文法の正しさを測る分析の観点である．本稿では，妥当性は置換誤り，位置誤り，消失，未翻訳，挿入，そして妥当性一般の六種類の誤りに細分類し，流暢性は並べ替え，語形，機能語，文法誤り一般，理解困難の五種類の誤りに細分類して誤り分類を行った．そして，得られた誤りにどのように対処すべきかを検討する．本稿の構成は以下のようになっている．第 1 章では本稿全体の概要を述べる．第 2 章では機械翻訳手法について述べる．第 3 章では本稿で採用した誤り体系について事例とともに詳細に述べる．第 4 章では実験の設定，結果を述べ，第 5 章で得られた結果に対して考察を行う．第 6 章では，レシピを対象とした関連研究について述べる．最後に第 7 章で本研究のまとめについて述べる．

Japanese-English Machine Translation of Recipe Texts*

Sato Takayuki

Abstract

Concomitant with the globalization of food culture, demand for the recipes of specialty dishes has been increasing. The recent growth in recipe sharing websites and food blogs has resulted in numerous recipe texts being available for diverse foods in various languages. However, little work has been done on machine translation of recipe texts. In this work, we address the task of translating recipes and investigate the advantages and disadvantages of traditional phrase-based statistical machine translation and more recent neural machine translation. Specifically, we translate Japanese recipes into English, analyze errors in the translated recipes, and discuss available room for improvements.

*Master's Thesis, Department of Information and Communication Systems, Graduate School of System Design, Tokyo Metropolitan University, Student ID 15890521, February 1, 2017.

目次

第 1 章　はじめに　1

第 2 章　機械翻訳手法　2

2.1　フレーズベース統計的機械翻訳 (Phrase-Based Statistical Machine Translation, PBSMT) 2

2.1.1　言語モデル 4

2.1.2　翻訳モデル 5

2.1.3　歪みモデル 6

2.1.4　デコーダと最適化 8

2.2　ニューラル機械翻訳 (Neural Machine Translation, NMT) . . . 9

2.2.1　リカレントニューラルネットワーク (Recurrent Neural Network, RNN) 9

2.2.2　エンコーダ・デコーダモデル 10

第 3 章　誤り体系　13

3.1　妥当性 13

3.2　流暢性 16

第 4 章　実験　18

4.1　実験データ 18

4.2　手法の設定 20

第 5 章　結果と考察　23

5.1　誤り分析 23

5.1.1 妥当性 23
5.1.2 流暢性 25
5.2 自動評価 27
5.3 モデルの拡張 28

第 6 章 関連研究 31

第 7 章 おわりに 33

参考文献 34

第 1 章　はじめに

近年，インターネット上で取得可能なレシピが増加している．例えば，日本の料理レシピサービスであるクックパッド*では 250 万品以上のレシピが取得できる（数字は 2016 年 9 月のもの）．同様に，アメリカの料理レシピサービスである Yummly†でも 100 万品以上のレシピが取得できる．取得可能なレシピが増加するにつれ，これらに関する研究も増加している．これまでに研究されてきたトピックとしては，例えば，レシピの解析 [1] や検索 [2]，要約 [3]，推薦 [4] などがある．レシピは文法が単純であるが，レシピ特有の単語や表現によって，その解析が難しいという問題がある．そのため，専用の辞書構築 [5] や特有のアノテーション [6] なども研究されている．レシピに関する研究が増加する中，これまでに研究されていないトピックとして機械翻訳がある．食文化の多様化とともにレシピの需要も国際的に拡大している．特に，日本食は健康にも良いことから，日本以外でも需要が大きい．機械翻訳で日本語のレシピを他言語に翻訳することで，多くの人がそれらを利用できるようになると思われる．そこで，本研究ではレシピ翻訳の現状と課題を確認するため，機械翻訳でレシピを翻訳し，その誤りを分析する．翻訳対象には 16,280 レシピから構成される日英対訳コーパスを使用する．翻訳手法としてフレーズベース統計的機械翻訳 [7] とニューラル機械翻訳 [8] を使用し，日本語のレシピを英語に翻訳する．翻訳誤りは，QTLaunchPad*3 の Multidimensional Quality Metrics（以下、MQM） [9] を参考に分類する．最後に，分類された誤りを分析し，それらの誤りにどのように対処すべきかを検討する．

*https://cookpad.com

†https://www.yummly.com

第 2 章　機械翻訳手法

本論文では，フレーズベース統計的機械翻訳とニューラル機械翻訳の 2 つの手法を用いて翻訳を行う．この章では，それらの手法について詳細に述べる．以下より，ある対訳コーパスを構成する 2 言語について，翻訳元の言語を原言語，翻訳先の言語を目的言語と呼ぶ.

2.1　フレーズベース統計的機械翻訳 (Phrase-Based Statistical Machine Translation, PBSMT)

PBSMT は統計的機械翻訳 (Statistical Machine Translation, SMT) の一種である．以下より，SMT と PBSMT のそれぞれについて述べる．

SMT では，単言語コーパス，対訳コーパスを用いて学習される複数の統計モデルから構成される [7]．学習されたモデルによって，訳出の意味的な正しさだけでなく，目的言語としての流暢さといった尺度も同時に考慮して訳出を生成する．SMT を構成する統計モデルは，ある原言語文 f が与えられたときに目的言語文 e が出力される確率 $P(e|f)$ を最大化するような $\hat{e}$ を選択するように学習を行う．$\hat{e}$ を求める式は，条件付き確率 $P(e|f)$ をベイズの定理より変形することで以下のように得られる．

$$\hat{e} = \underset{e}{\operatorname{argmax}}\, P(e|f) = \underset{e}{\operatorname{argmax}}\, \frac{P(f|e)P(e)}{P(f)} = \underset{e}{\operatorname{argmax}}\, P(f|e)P(e) \tag{2.11}$$

式変形途中に現れた分母の $P(f)$ は定数であるため無視することができる．右辺を構成する $P(e)$ と $P(e|f)$ は，それぞれ言語モデル確率，翻訳モデル確率と呼ばれる．言語モデルは，翻訳結果の目的言語文 e が文としてどれだけ自然かを確率的に保証するモデルであり，目的言語文のみに対して学習される．翻訳モデルは，ある目的言語文 e に対する原言語文 f の意味的正しさを条件付き確率 $P(f|e)$ によって表している．ここで，f から e ではなく，e から f を生成するモデルであることに留意されたい．また，実際には直接 e と f を関連づけず、ある複数の過程を経て e から f が生成されると仮定している．その過程を導出 d と呼び，d を考慮し隠れ変数として扱うと $\hat{e}$ を求める式は以下のようになる．

$$\begin{aligned}\hat{e} &= \underset{e}{\operatorname{argmax}} P(f|e)P(e) \\ &= \underset{e}{\operatorname{argmax}} \sum_{d} P(f,d|e)P(e)\end{aligned} \tag{2.12}$$

式 2.12 では，同じ翻訳 e が得られたとしても、異なる導出が行われる場合があり，すべての導出に対してその確率の合計を求めている．

PBSMT における翻訳過程は，式 2.12 における隠れ変数 d に，フレーズ（文を構成する部分的な単語列）単位のアライメント α と，対訳文を構成するフレーズペアを表す ϕ を導入することで以下のように定式化される．

$$\begin{aligned}\hat{e} &= \underset{e}{\operatorname{argmax}} \sum_{\phi,\alpha} P(f,\phi,\alpha|e)P(e) \\ &= \underset{e}{\operatorname{argmax}} \sum_{\phi,\alpha} P(f,\alpha|\phi,e)P(\phi|e)P(e) \\ &\approx \underset{e}{\operatorname{argmax}} \sum_{\phi,\alpha} P(f,\alpha|\phi)P(\phi|e)P(e)\end{aligned} \tag{2.13}$$

上式では，隠れ変数 α および ϕ が導入された確率モデルにおいて，各変数が独立であると仮定しているため，最終行で近似を行っている．そして，翻訳生成の過程は，言語モデル $P(e)$ による目的言語文 e の生成，句翻訳モデル $P(\phi|e)$ による，フレーズへの分割および対訳フレーズの生成，そして句歪みモデル $P(f,\alpha|\phi)$ による，フレーズの並び替えの決定と原言語文 f の生成，の統計モデルで表現される．つまり，単語単位，フレーズ単位で得られた翻訳を自然な文になるよう並べ替える統計モデルとなっている．

PBSMT は，英語とフランス語のような語順が似ている言語間の翻訳で高い精度を達成している [7]．一方，日本語と英語のように語順が大きく違う言語間の翻訳ではこの限りではない．これは，組み合わせの探索空間が広くなり，並び替える距離を制限する必要があるためである．また，文法を仮定しない手法であるため，フレーズ単位の対応づけおよび並び替えなどの非常に弱い制約により，どのような言語対に対しても適用できるが，文法的に誤った訳出が多く見られるという欠点もある．

以下より，それぞれの統計モデルについて以下より詳細に述べる．

2.1.1 言語モデル

言語モデルは流暢な文が生成されることを保証するのに重要な役割を果たす．原言語文 f から目的言語文 e へ翻訳するとき，e の流暢さを担保するよう機能する．

例えば，機械翻訳によって，he is big, is big he, this is a red dog の 3 つの訳出が得られたとする．ここで，2 つめの例には語順に誤りがあり，3 つめの例には red dog は意味的に正しい状況は少ない．

ここで、言語モデルは統計的な枠組みで訳出の流暢さを言語モデル確率 $P(e)$ で表し，$P(e)$ は以下の式で与えられる．なお $count$ は頻度を表す．

$$P(e) = \frac{count(e)}{\sum_{e'} count(e')} \tag{2.14}$$

しかし，文単位での頻度では，ある文 e の言語モデル確率を考えたとき，全く同じ文が出現しなければその頻度は 1 となり，非常に小さな確率となってしまう．この問題を解消するために，$P(e)$ は確率を文に含まれる単語ごとに計算する．ここで，$P(e)$ は文 e を構成する N 個の各単語の同時確率とすると，以下の式で与えられる．

$$\begin{aligned} P(e) &= P(e_1, e_2, e_3, ..., e_N) \\ &= P(e_1)P(e_2|e_1)P(e_3|e_1, e_2)...P(e_n|e_1, e_2, e_3, ..., e_{N-1}) \\ &= \prod_{i=1}^{N} P(e_i|e_1, e_2, ..., e_{i-1}) \\ &\approx \prod_{i=1}^{N} P(e_i|e_{i-n+1}, e_{i-n+2}, ..., e_{i-1}) \end{aligned} \tag{2.15}$$

条件部は文 e の時と同様，スパースになりうるため $n-1$ 個前までの単語にのみ依存するとして近似している．これは，機械翻訳で広く利用されている n-gram 言語モデルであり，本論文の言語モデルにおいて適用されている．現在の統計的機械翻訳システムでは，n=5 もしくは n=4 までの n-gram 長を用いるのが一般的である．

2.1.2 翻訳モデル

翻訳モデルは，ある目的言語文 e が与えられたときにその対訳として f が適切であるか，を統計的に与えるモデルである．言語モデルの時と同じく，文単位では翻訳モデルとして機能するような確率を得ることが困難であるため，単語またはフレーズの対訳対の翻訳確率を組み合わせることによって，文の翻訳確率を求める．ここでは，単語単位の翻訳確率の導出について述べたのち，単語対とフレーズ対の抽出方法について述べる．

はじめに，単語に基づく翻訳モデルである IBM モデル [10] について述べる．IBM モデルにはモデル 1 からモデル 5 まで存在し，それぞれは翻訳モデル $P(f|e)$ の近似方法が異なる．また，数字が大きくなるつれてより精巧なモデルとなっている．以下より，IBM モデル 1 に焦点をあて $P(f|e)$ の導出について述べる．

式 2.12 にあるように，翻訳モデル $P(f|e)$ には隠れ変数 d を導入している．ここで，対訳文 (f, e) の単語単位の対応 (f_j, e_i) を表す単語アライメント α を隠れ変数として導入する．すると，翻訳モデルは，単語アライメント α を介して単語単位で e から f を生成する確率モデルとなる．IBM モデルでは，原言語文 f(単語数 m) の各単語は，目的言語文 e(単語数 l) に対応する単語をそれぞれ 1 つもち，目的言語文 e には空単語 e_0 が存在するという仮定をする．つまり，目的言語文の各単語は原言語文の複数の単語に対応する可能性があり，方向のある 1 対多のアライメントとなっている．以下に，IBM モデル 1 における文の翻訳確率 $P(f|e)$ を示す．

$$
\begin{aligned}
P(f|e) &= \sum_{a} P(f, \alpha|e) \\
&= \operatorname*{argmax}_{a} P(\alpha|e) P(f|e, \alpha) \\
&\approx \sum_{\alpha} \epsilon \sum_{j=1}^{m} t(f_j|e_{a_j}) \\
&= \epsilon \sum_{a_1=0}^{l} \sum_{a_2=0}^{l} \cdots \sum_{a_m=0}^{l} \prod_{j=1}^{m} t(f_j|e_{a_j}) \\
&= \epsilon \prod_{j=1}^{m} \sum_{i=0}^{m} t(f_j|e_i)
\end{aligned}
\tag{2.16}
$$

$a_m = l$ は，原言語文 f の m 番目の単語に対して目的言語文 f の l 番目の単語が

対応してることを意味する．IBM モデル 1 は，上式における $P(a|e)$ を一様分布 ϵ としており，原言語文 f の単語 f_j が目的言語文 e の単語 e_i に翻訳される確率 $t(f_j|e_i)$ のみを用いた翻訳モデルとなっている．なお，単語翻訳確率 $t(f_j|e_i)$ の初期値は共起頻度などで与えられ，$P(f|e)$ の対数尤度 $\log P(f|e)$ を最大化するよう EM アルゴリズムで更新される．その他の IBM モデルはより精巧なモデルとなっており，各単語の絶対位置や e_i が原言語文の何単語分に対応するかの確率，e_i に対応する原言語の単語 f_{ik} の原言語文中での位置 j の確率などを考慮している．

ここで，目的言語文のフレーズ $\overline{e}$ が原言語文のフレーズ $\overline{f}$ に翻訳される確率 $P(\overline{f}|\overline{e})$ を考える．フレーズアライメント $\overline{\alpha}$ は単語のときより複雑となるため，その数は膨大となり $P(\overline{f}|\overline{e})$ の計算には近似が必要となる．そこで，IBM モデルを各方向に適用して各方向の単語翻訳確率 $P(f_j|e_i)$，$P(e_i|f_j)$ を得る．得られた両方向のアライメントに対し，両方向共にある対応点のみを用いる intersection，両方向の対応点を全て用いる union，intersection と union の中間のように機能する grow などのヒューリスティックによって，両方向のフレーズの対応を得ることができる．grow は intersection で得られた点からはじめ，すでに採用した対応点の周りに union で得られた点を加えるものである．grow は縦・横までを補い，その派生である grow-diag は縦・横・対角を補う．図 2.1 に grow-diag でフレーズ対応を獲得する例を示す．

フレーズの翻訳確率 $P(\overline{f}|\overline{e})$ は，ヒューリスティックにより抽出されたフレーズペアの頻度をもとに求められる．しかし，文長だけのフレーズペアを抽出した場合，その数は膨大になってしまうため、抽出されるフレーズペアの原言語側あるいは目的言語側の長さを 5 以下とするような制約を加えたりする．

2.1.3 歪みモデル

歪みモデル $P(f, \alpha|\phi)$ は，目的言語側で連接するフレーズのペアに対して，原言語側の距離に対応するペナルティを考慮するモデルである．つまり，言語間の対応するフレーズの位置が近いほど訳出として正しい，という仮定のもと，そのペナルティが小さくなるように機能する．ここで π を，フレーズペア ϕ とアライメント α から求められる，各フレーズペアが被覆している原言語のスパンの集合

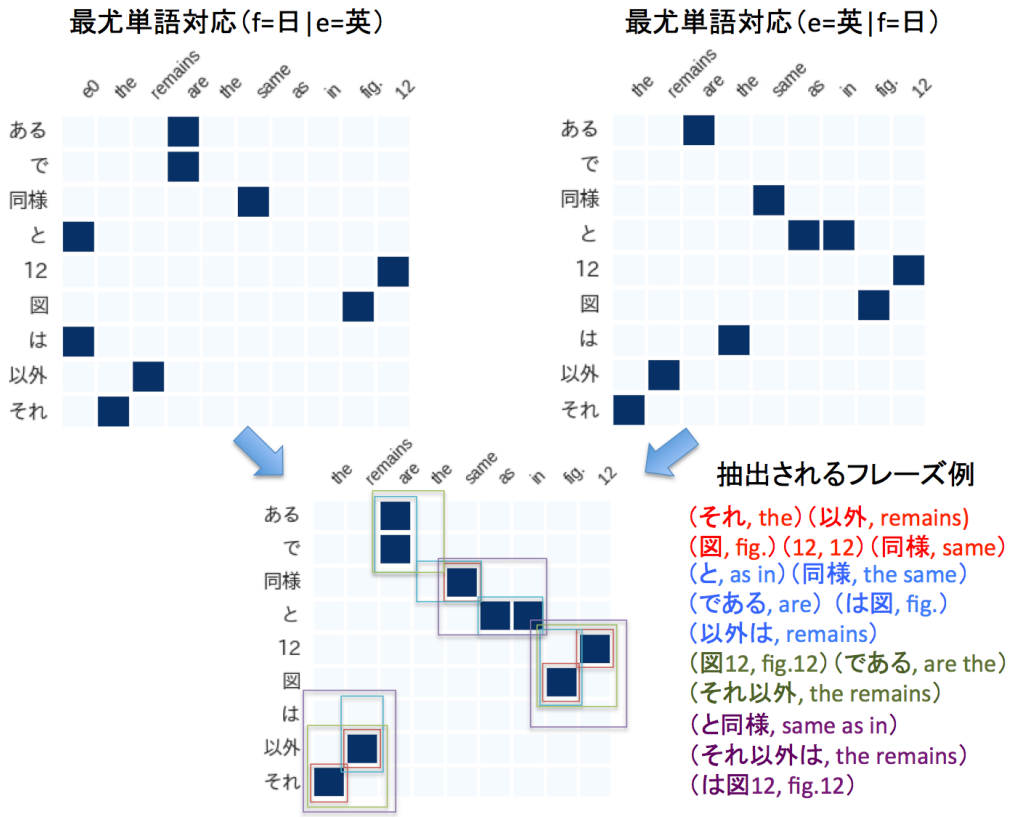

図 2.1　grow-diag でのフレーズ対応獲得例

とする．π_k は，k 番目のフレーズペアが被覆している原言語文のスパンであり，$\pi_k = [start_k, end_k)$ となり，$start_k$ と end_k は以下のように与えられる．

$$start_k = 1 + \sum_{k'|\alpha_{k'}<\alpha_k} |\overline{f}(\phi_k)| \tag{2.17}$$

$$end_k = start_k + |\overline{f}(\phi_k)| \tag{2.18}$$

歪みモデルは $start_k$ と end_k を用いて以下のように与えられる．

$$\log P(f, \alpha|\phi) = \log P(\pi|\phi) \approx \sum_{k=1}^{L} -|start_k - end_{k-1}| \tag{2.19}$$

2.1.4 デコーダと最適化

デコードでは入力文 f に対し，式 2.12 における右辺を最大化するような目的言語文 $\hat{e}$ を出力する．このとき，入力 f が与えられていることから，以下のような線形モデルの最大化を解く問題として表すことができる．

$$\begin{aligned}\hat{e} &= \underset{e}{\operatorname{argmax}}\, P(f|e)P(e) \\ &= \underset{e}{\operatorname{argmax}}\, \frac{\exp(w^{\mathrm{T}}h(f,e))}{\sum_{e'}\exp(w^{\mathrm{T}}h(f,e'))} \\ &\approx \underset{e}{\operatorname{argmax}}\, w^{\mathrm{T}}h(f,e')\end{aligned} \tag{2.110}$$

実際のデコーディングでは、全ての候補を考慮するのは計算量の問題で不可能であるため，導出過程で制約を加えることでその候補の数を削減し，翻訳を実現する．

ここで，式 2.110 における重みベクトル w の最適化方法について述べる．$h(f,e)$ は素性ベクトルと呼ばれ，言語モデルや翻訳モデル，その他の SMT を構成するモデルから得られる値が，各成分として組み込まれている．そして，重みベクトル w は，正しい翻訳を得るのにどのモデルから得られる値が重要かを表すこととなる．

言語モデルや翻訳モデルなど各モデルの値は，大規模対訳コーパスを用いて学習される．一方，w の最適化では，実際に翻訳したい文書（テストセット）に近い対訳コーパス（開発セット）があると仮定し，その対訳コーパスで翻訳を行った時に，できるかぎり誤りが小さくなるようにパラメータを決定する．つまり，開発セットで翻訳した結果を，翻訳文の自動評価尺度の一つである Bilingual evaluation understudy (BLEU) [11] などを用いて評価し，そのスコアがなるべく良くなるよう w の値を調整するのである．w の最適化手法で用いられる手法のうち，Minimum Error Rate Training (MERT) [12] と呼ばれるものでは，翻訳結果のエラー率に基づく損失関数を定義し，それを最小化するようにパラメータを決定する．

2.2 ニューラル機械翻訳 (Neural Machine Translation, NMT)

NMT は入力された単語列を低次元で密なベクトルに変換し，これをもとに単語列を出力することで翻訳を行う [13] [8]．翻訳では各単語の依存関係を考慮する必要があるため，ベクトルが持つ情報の伝播には系列データを扱うのに適したものであるリカレントニューラルネットワークを用いる．以下より，リカレントニューラルネットワークの構造と，それを用いてどのように翻訳が行われるかを述べる．

2.2.1 リカレントニューラルネットワーク (Recurrent Neural Network, RNN)

RNN は，内部に有向閉路を持つニューラルネットワークを指す．この構造により，系列データに対して振る舞いを動的に変化させることができる．RNN の動作は，各タイムステップ t につき入力ベクトル x_t と h_{t_1} を受け取り，h_t を更新，そして出力 y_t を返す．RNN を適用するタスクによっては，出力 y_t を系列データにおける最後の入力を受け取った時のみ返す場合もある．h_t は，入力 x_t と h_{t_1} のそれぞれに対する重み行列 W_{xh}，W_{hh} と活性化関数 f を，y_t は h_t に対する重み行列 W_{hy} と活性化関数 g を用いて以下の式のように表される．なお，活性化関数にはロジスティック関数や双曲線正接関数，正規化線形関数などが用いられる．

$$h_t = f(W_{xh}x_t + W_{hh}h_{t-1}) \tag{2.21}$$

$$y_t = g(W_{hy}h_t) \tag{2.22}$$

それぞれの重み行列は y_t に対応する誤差をもとに誤差逆伝播法によって学習される．

RNN は，各重み行列によって系列データにおける依存関係を捉えることができる．ゆえに，系列ラベリングなどのタスクだけではなく，言語モデルの計算に適用することで n-gram 言語モデルより長い距離を考慮することもできる [14]．しかし，その系列が長くなるにつれ，誤差逆伝播法で計算される勾配が爆発的に大きくなるか，あるいは 0 に消失してしまいやすい性質がある．そのような問題に対し，長・短期記憶 (Long Short-Term Memory, LSTM) [15] や Gated Reccurent

Unit (GRU) [16] などが提案されている．

2.2.2 エンコーダ・デコーダモデル

NMT を構成するのは，Encoder と Decoder と呼ばれる二つの RNN であり，単語列からベクトルへの変換は Encoder，訳出は Decoder の役割によるものである．Bahdanau ら [8] は注意型ネットワークを用いたモデル [8] を提案し，Decoder が出力単語を求める際，Encoder で表現されている各単語についてその位置に対応する隠れ層の情報をどれだけ使用するか（注意度）を動的に決定している．. 以降，本論文で NMT と称した場合，Bahdanau らの注意型ネットワークモデルを指すものとする．以下に，NMT について式とともに詳細に述べる．

Bahdanau らのモデルでは，Encoder は双方向性 RNN (bidirectional RNN) となっており，長期依存を考慮できる GRU で構成されている．順方向の RNN は原言語文 ($\boldsymbol{x} = [x_1, x_2, \cdots, x_{|\boldsymbol{x}|}]$) を順番に，逆方向の RNN は逆順に受け取り，それぞれの隠れ層は式 2.21 により更新され，$(h_1, h_2, \cdots, h_{|\boldsymbol{x}|})$ を得る．

ここで x は語彙数次元のベクトルであり，一つの次元のみ 1 でその他は 0 で埋められているベクトルである．一般的な NMT では，考慮する語彙数を $30,000$ から $80,000$ 程度に制限するが，そのパラメータ数は非常に大きいため，RNN への入力とする前に低次元で密なベクトル $e(x) = W_{xe}x$ に変換する．

各 $e(x_j)$ を変換した双方向の RNN により，Encoder の各位置 j での隠れ層 h_j は以下のように表される．

$$h_j = [\overrightarrow{h_j}^\top : \overleftarrow{h_j}^\top]^\top \tag{2.23}$$

Decoder は順方向の RNN であり，Encoder で得られた情報をもとに，先頭から 1 単語ずつ出力し，翻訳文を生成することで目的言語文 ($\boldsymbol{y} = [y_1, y_2, \cdots, y_{|\boldsymbol{y}|}]$) を得る．各 y は x と同様，語彙数次元のベクトルであり，1 つの次元のみ 1 でその他が 0 で埋められているベクトルである．Decoder は各 y の予測を，語彙数だけ候補のある分類問題として解き，その出力は確率分布となっている．

ここで，Encoder からの情報はアテンションベクトル c によって Decoder に渡される．目的言語文における i 番目の単語の出力時，アテンションベクトル c_i は

Encoder の各隠れ層 h_j の重みつき和で与えられる．

$$c_i = \sum_{j=1}^{|x|} \alpha_{i,j} h_j \tag{2.24}$$

重み $\alpha_{i,j}$ はその和が 1 になるよう正規化され，Decoder の隠れ層の出力 s_{i-1} と h_j より以下の式で与えられる．

$$\alpha_{i,j} = \frac{\exp(v_a^\top \tanh(W_a s_{i-1} + U_a h_j))}{\sum_{j'=1}^{|x|} \exp(v_a^\top \tanh(W_a s_{i-1} + U_a h_{j'}))} \tag{2.25}$$

ここで，v_a は重みベクトル，W_a，U_a はそれぞれ重み行列である．

Decoder の隠れ層 s_i は，入力に s_{i-1}，y_{i-1}，c_i を受け取り，非線形関数 f と各重み行列を用いて以下のように表される．

$$s_i = f(W_{ss} s_{i-1} + W_{ys} e(y_{i-1}) + W_{cs} c_i) \tag{2.26}$$

各位置 i での Decoder の出力層の出力 o_i は， 非線形関数 g と各重み行列を用いて得られ，以下のように表される．

$$o_i = g(W_{sy} s_i + W_{yy} e(y_{i-1}) + W_{cy} c_i) \tag{2.27}$$

最終的に，$(y_i|\boldsymbol{y}_{<i}, \boldsymbol{x})$ は，o_i をソフトマックス関数により確率分布に正規化したもので得られる．各重み行列のパラメータ θ は以下の目的関数 $\mathrm{L}(\theta)$ を最大化するよう学習される．なお，N は学習データの総数を示す．

$$L(\theta) = \frac{1}{N} \sum_{n=1}^{N} \log(y_i^{(n)} | \boldsymbol{y}_{<i}^{(\boldsymbol{n})}, \boldsymbol{x}^{(n)}, \theta) \tag{2.28}$$

NMT は構文情報を利用していないにもかかわらず，自然な文を生成する．また，従来の統計的機械翻訳にある言語モデルや翻訳モデルなどをモデル化する必要がないため，それらと比較するとシンプルな構造となっている．一方，NMT では出力可能な語彙の数を制限する必要がある．これは，一般的な NMT では，出力層においてソフトマックス関数による演算を行なっているためである．ソフトマックス関数による演算には，出力可能な語彙の数に比例して計算量が増加する．ゆえに，多

くのフレーズ候補を確保しておける PBSMT と比較すると，NMT では低頻度語の翻訳が難しい [13]．また，原言語側のどの単語にも対応しない単語を出力しやすいという欠点もある [17].

第 3 章　誤り体系

本論文では，PBSMT と NMT の訳出に対してブラックボックス分析を行う．ブラックボックス分析とは，訳出の導出過程を考慮せずに出力のみを分析するものである．そのため，翻訳前に必要な過程（単語分割や単語アライメント獲得）を無視する．ブラックボックス分析に用いる誤り体系は，MQM ANNOTATION DECISION TREE [9] を参考にした．これは誤りを決定木で分類するものである．各誤りは優先度を持っており，より高い優先度を持つ誤りに分類された場合，優先度の低い誤りに分類されるかどうかは考慮しない．それぞれの誤りに対し，Yes/No で答えられるような問いがあり，Yes ならばその誤りに分類される．同様の作業を，最も優先度の低い誤りまで繰り返す．MQM ANNOTATION DECISION TREE を用いることで，一貫性を保って誤りを分類できる．

MQM における誤り体系は妥当性と流暢性の二種類に大別される．妥当性は入力文と翻訳結果の整合性の度合いを測る分析の観点であり，流暢性は翻訳結果の語法や文法の正しさを測る分析の観点である．以下より，妥当性と流暢性の細分類と分類・分析方法について述べる．

3.1　妥当性

MQM における妥当性に関する誤りを以下に示す．

1. 消失
2. 未翻訳
3. 挿入
4. 術語
5. 誤翻訳
6. 妥当性一般

本論文での妥当性における誤り分類には，通常の MQM ANNOTATION DECISION TREE と異なる点が三つある．一つ目に，誤翻訳を置換誤りと位置誤りを二つの別々の誤りとして考える．前者はある単語・フレーズを異なる意味の単語・フ

レーズに翻訳している誤りであり，後者は適切な位置に訳出できていないために意味が異なる誤りである．ここで，フレーズは日本語における一つ以上の文節，英語における句もしくは節を指す．変更した理由として，各翻訳手法で置換誤りと位置誤りの傾向が大きく異なり，その差を反映させるためである．二つ目は，誤りの分類が MQM の決定木の順番でなく，置換誤りと位置誤りを優先誤りとした点である．これは，置換誤りなのか、消失＋挿入なのかという分類を容易にするためである．また，NMT では消失や挿入が多いという点もこの変更の理由の一つである．三つ目は，術語誤りを除いた点である．術語誤りはドメインの差によって起きる語義の選択誤りである．本論文で用いたコーパスは一つの分野に限定したものであり，術語誤りはほとんど見られない．以上を踏まえて，本論文では，以下の優先度の細分類を採用する．

1. 置換誤り
2. 位置誤り
3. 消失
4. 未翻訳
5. 挿入
6. 妥当性一般

誤り分類は，変更した誤り分類に従いつつ，以下の流れに沿って行う．

1. 単語・フレーズで対応の取れている箇所を，原言語文の文頭から順に主観によって判定する．（この時，単語・フレーズの位置の正誤は問わない）
2. 対応の取れた単語・フレーズを正しいものとし，周辺単語に対し，品詞の一致などの情報から置換誤りを決定する．
3. 置換誤りを決定した後，新しく完成したフレーズがあればそれも含めて，位置誤りに該当するかを決定する．
4. 置換誤りと位置誤りに分類されなかった単語・フレーズに対して，残りの誤り体系を考える．

本論文では，原言語に日本語を用いているため，文節単位での誤翻訳一つにつき一つの誤りとする．以下より，各誤り体系について例とともに説明する．

置換誤り: 原言語文のある単語の意味が，置換の誤りによって目的言語文のある単語において別の意味に変わっている場合の誤りである．以下の例では，‘Heat’ は「割る」の置換誤りとして分類される．‘Heat’ は動詞であるため，「割る」との品詞の一致がとれる．加えて「卵を」の訳出である ‘an egg’ を目的語としているので，「割る」が ‘Heat’ に翻訳されたものとして扱う．

卵 を 割る 。
Heat an egg .

位置誤り: 原言語文のフレーズが不適切な位置へ出力することで別の意味に変わっている場合の誤りである．以下の例では ‘from step 1’ が「1 の」の位置誤りとして分類される．誤り数は一つである．

1 の 器 に レタス を 入れる 。
Add the lettuce from step 1 into a bowl .

消失: 原言語文に存在し，かつ，省略されてはいけない単語の意味が目的言語文で表されていない場合の誤りである．以下の例では，「はちみつ」に対応する単語が消失している．

はちみつ 生地 は 1 次 発酵 まで 済ませる 。
Make the dough until the first rising .

未翻訳: 原言語文の単語がそのままの形で目的言語文に出現している場合の誤りである．そのままの形で出現している単語一つにつき一つの誤りとする．本論文で用いた NMT は原言語文の単語をそのまま出現するようなモデルではない．よって，この未翻訳誤りは PBSMT における誤り分析でのみの分類となる．以下の例では，原言語文の「狭い」をそのまま出力している．

長 さ を 整え ， 幅 の 狭い ほう で カット する 。
Adjust the length , and cut the 狭い into it .

挿入: 原言語文に存在しない情報が目的言語文で表されている場合の誤りである．本論文では，英語側に出現した単語を日本語に翻訳し，1 文節につき一つの誤りと

する．以下の例では，‘red’ は「赤い」，‘into a pot’ は「鍋に」と翻訳されたとして，二つの誤りとする．

ソース を 加える 。
Add the red sauce into a pot .

妥当性一般: 上記のどの誤りにも分類が難しい場合，この「妥当性一般」に分類する．誤り個数は原言語文の文節の個数とする．以下の例では四つの誤りとする．

出来上がった 時 に 倒れ ない ため です 。
It will be hard to cover the cake .

3.2 流暢性

MQM における流暢性に関する誤りを以下に示す．

1. 並べ替え
2. 語形
3. 機能語
4. 文法誤り一般
5. 理解困難

並べ替え，語形，機能語，文法誤り一般は文法的に不適切な場合に分類される誤りである．理解困難は，文法的には適切だが語義を考慮すると不適切な場合に分類される誤りである．分類方法は妥当性の時に従ったものから，原言語文と対応をとる過程を除いたものになる．文法誤りは文法的にみて誤りを含む単語・句・節に適用され，理解困難は文法的には正しいが意味をとれない箇所に適用される．本論文では，タイトルと材料に対しては，名詞句のみの出力でも誤りとしない．手順は，主語と動詞を含んだものを文として正しいとする．つまり，手順で名詞句のみの出力ならば誤りとする．以下より，各誤り体系について例とともに説明する．

並べ替え: 不適切な位置に単語・フレーズが出現している場合の誤りである．複数の誤り候補が考えられる場合には，全体の誤り個数が最小となるような候補に適

用する．目的言語側のフレーズ単位で並べ替えが必要とされる際には，そのフレーズに含まれる内容語の数だけ誤り数を加算した．以下の例では，‘Parts of the face’ の場所が不適切であり，正しくは ‘place’ と ‘on’ の間にあるべきである．よって，対象フレーズに含まれる内容語は ‘Parts’ と ‘face’ であり，誤り数は二つとなる．

Parts of the face , place on a baking sheet .

語形: 主語との不一致，または時制の不一致の場合の誤りである．動詞の個数だけ誤りを加算する．以下の例では，‘uses’ が不適切であり誤り数は一つである．

I uses the dough for step 4 .

機能語: 前置詞，限定詞，助動詞，関係詞の誤用の場合の誤りである．不要な機能語の挿入，必要な機能語の消失，機能語の使用誤りが該当する．以下の例では，不要な ‘to’ が挿入されているため，誤り数は一つである．

It ’s finished to .

文法一般: 上記三つの誤りに該当しない場合の誤りである．主に，不要な内容語の挿入や必要な内容語の消失が該当する．以下の例では，動詞が欠落しているため，誤り数は一つである．

The honey dough for the first rising .

理解困難: 文法的には正しいが意味が取れない場合の誤りである．文の冒頭にある単語・フレーズは正しいとし，意味が取れなくなる箇所から内容語の数だけその誤りを加算する．以下の例では，‘I was going to be taken’ までは正しいとし，以後の ‘from the cake’ と ‘in the future’ が誤っているとする．各フレーズに含まれる内容語はそれぞれ ‘cake’ と ‘future’ なので，誤り数は二つである．

I was going to be taken from the cake in the future .

第 4 章　実験

4.1　実験データ

本実験では 16,283 レシピから構成される日英対訳コーパスを使用する．このコーパスは，クックパッドが海外向けサービスを開発する過程で構築されたものである．クックパッドのレシピは主にタイトルや材料，手順などのフィールドから構成されている．以下はレシピのタイトルの対訳である．

> 簡単シンプル！ふわふわ卵のオムライス
> Easy and Simple Fluffy Omurice

以下は材料の対訳の一例である．材料は名前と分量から構成されている．

> ご飯 (冷ご飯でも可)
> Rice (or cold rice)
>
> 2 杯分
> 2 rice bowl's worth

以下は手順の対訳の一例である．一般的な対訳コーパスと違って，一つの対訳が一つの文とは限らない．この例では一つの対訳が二つの文となっている．

> ケチャップとソースを混ぜあわせます．味見しながら比率は調節してください．
> Mix the ketchup and Japanese Worcestershire-style sauce. Taste and adjust the ratio.

これらの対訳は，初訳と修正という二つの作業を通して収集されたものである．まず，日本語ネイティブ 1 名がレシピを英語に初訳した．ただし，日本語ネイティブは海外在住の日本人や，配偶者が英語ネイティブの日本人であった．次に，英語ネイティブ 2 名が初訳結果を確認して，必要があればこれを修正した．なお，日本語ネイティブと英語ネイティブはともに料理に精通するものであった．

この人手翻訳により構築されたコーパスはタイトル 16,283 文と材料 139,477 文，手順 118,002 個（≠ 文）から構成されている．なお，手順 118002 個を構成する文の数は日本語側で 209,291 文，英語側で 190,111 文であった．ただし，文の数は，日本語側は句点で，英語側はピリオドで分割することで計数した．各文に対し，日本語文には形態素解析器 MeCab（+IPADIC）[18] で単語分割し，英語文は Moses [19] の添付スクリプトで単語分割した．

上記の処理に加え，機械翻訳のモデル学習を妨げないよう，次の 3 つの前処理を行った．1 つめは，手順に対して行なった前処理である．手順は上記の例にあるように，生データのままでは 1 行に複数の文（句点もしくはピリオドまで文とする）が含まれうる．また，この対訳コーパスは各手順において意味が同等になるように構築されたため，必ずしも 1 行に含まれる文数が日本語側と英語側で一致するとは限らない．このような対訳文は学習を妨げうる．例えば，日本語側 2 文，英語 3 文といった対訳文を学習時に用いると，やはり多くの文は文対応が取れているために，影響を受けてしまい最終的に得られるモデルの翻訳精度が下がってしまう．そこで行うのが，こうした対訳文を排除する前処理である．日本語側の各手順文を句点，その対訳をピリオドで分割し，得られた文対のうち，次の 2 つの条件のいずれかを満たすものに限り実験に用いた．ただし，2 つめの条件を満たす文については，1 文目のピリオドを ‘, and’ の文字列に置換することで 2 文を接続した．

1. 日本語側と英語側の文数が一致するもの
2. 日本語側が 1 文かつ英語側が 2 文であるもの

この前処理によって，25,654 手順対が除かれた．日本語側に含まれていた文数は 59,282 であり，英語側の文数は 57,016 であった．2 つめの前処理は，1 つめの前処理で得られた対訳文のうち，文の単語数比が 4 以上である文を除くものである．このレシピコーパスでは，英語側の文が比較的簡易に記述されていることがあり，そのような場合において日本語側と英語側で文の単語数が大きく異なってしまう場合がある．以下にその例を示す．

> 関西 の お店 の 味 !! 我が家 の お好み焼き 。
> kansai-style okonomiyaki .

表 4.1　各フィールドの文，単語の総数

	言語	タイトル	材料	手順	総数
文		16,170	131,938	124,771	272,879
単語	日本語	115,336	322,529	1,830,209	2,268,074
	英語	100,796	361,931	1,932,636	2,395,363

3 つめの前処理は，どちらか一方の言語で 40 単語以上を含む文を除くものである．表 4.1 は前処理後に得られた各文数と単語数を示している．語彙数は日本語が 23,519，英語側が 17,307 であった．前処理によって，レシピから一部のタイトル，材料，手順は削除されうる．

このうち，レシピ単位で 100 レシピずつランダムにサンプリングしたものをそれぞれ開発セット（1,706 文），評価セット（1,647 文）とした．誤り分析は，評価セットからランダムにサンプリングした 25 レシピ（タイトル 25 文，材料 222 文，手順 195 文）に対して行った．また、前述の 100 レシピに対して，BLEU [11] と RIBES [20] による自動評価も行なった．RIBES の単語適合率に対する重み α は 0.25 とした．また，（出力文長 ÷ 参照訳の長さ）で与えられるペナルティ（以下、BrevityPenalty）に対する重み β は 0.10 とした．なお、BLEU は Moses の添付スクリプトを用い，RIBES はバージョン 1.03.1*を用いた．

4.2　手法の設定

PBSMT には最も代表的な PBSMT のツールである Moses（ver2.1.1）[19] を用いた．単語アライメントは Giza++†により獲得し，言語モデルは単言語コーパスとして対訳コーパスのうち英語側全文を用いて学習した．フレーズテーブルサイズは約 300 万対であった．各素性については開発セットで MERT によるチューニングを行い，重みを決定した．

また手順文の翻訳では，複数の訳出候補 N-best に対して RNN 言語モデル [14] によってリランキングする実験も行った．評価セットのうち，手順文 722 文を翻

*http://www.kecl.ntt.co.jp/icl/lirg/ribes/index-j.html
†http://github.com/moses-smt/giza-pp

訳対象とした．リランキングは，RNN 言語モデルで PBSMT から得られる複数の訳出候補をリスコアすることで行う。文の各単語 w_t を順に RNN に入力することで，次の単語 w_{t+1} の予測確率が得られ，その対数の総和をとることで文をリスコアする．なお，得られたスコアは文が短いほど高くなってしまうため，文長を考慮した正規化を行う．正規化には， [21] で NMT におけるビームサーチで短い文を優先して選択しないよう行われた，スコアを (1 + 文長)/6 で割ったものを採用した．リランキングを行う訳出候補数 N は 10，20，30，40，50，100 とし，ベースラインは 1 のものである．RNN 言語モデルは，RNNLM Toolkit[‡]を用い，手順文 124, 771 文から学習した．ハイパーパラメータである隠れ層の次元数 h は 32，64，128，256 とし，Back Propagation Through Time(BPTT) は 7 単語とした．なお，PBSMT の誤り分析はリランキングしていないモデルの出力より行った．

NMT には Bahdanau らの手法を参考 [8] に独自に実装したものを用いた．モデルを構成するユニットには LSTM を採用した．NMT の埋め込み層と隠れ層の次元数はともに 512 で，隠れ層は 1 層とした．入出力可能な語彙は制限せず，未知語に対する特定の記号への置換は行なっていない．最適化手法には学習率の初期値を 0.01 とした Adagrad [22] を用いた．また，原言語と目的言語の埋め込み層の初期値は word2vec[§] のデフォルト設定で学習したものを用いた．原言語の埋め込み層の初期値は対訳コーパスとは別に用意した手順約 1, 300 万文から学習した．ここで，目的言語の埋め込み層の初期値は対訳コーパスの英語文のうち手順約 12 万文から学習した．タイトルを学習データから除いたのは，タイトルが自由な文体で書かれているため，学習を妨げると考えたためである．また，材料を学習データから除いたのは，平均単語数が少ないため，窓幅による文脈を考慮できず，学習を妨げると考えたためである．その他の重み行列については，いずれもランダムの初期値を与えた．学習時のバッチサイズは 64 とした．エポック数は 10 で，各エポックのモデルのうち開発セットで最も高い BLEU を示すモデルを選択した．

また，NMT では複数のモデルによるアンサンブル出力をすることで翻訳精度が向上することがいくつかの研究で報告されている [23] [24] [25]．本実験でも同様に，4 つのモデルによるアンサンブル出力を行うことでその効果を検証した．アン

[‡]http://www.fit.vutbr.cz/~imikolov/rnnlm

[§]https://radimrehurek.com/gensim/models/word2vec.html

サンブルに用いるそれぞれのモデルのハイパーパラメータはすべて上記のものと同一である．なお，NMT の誤り分析は単一のモデルの出力より行った。

第 5 章　結果と考察

5.1　誤り分析

5.1.1　妥当性

各手法の妥当性の誤り数を表 5.1 に示す．() 内は全体における誤り個数の割合を示す．表 3 から，NMT と比較すると，PBSMT は位置誤りが多いことがわかる．一般的に，PBSMT は語順が離れた言語対に対し，並べ替えが困難となり，翻訳精度が落ちる．本論文で用いたコーパスは単語数が少ない文が多数を占める．最も単語数の多い手順のみを考慮しても，平均単語数は日本語が 14.0，英語が 15.0 であった．単語数が少ない場合，並べ替えの最大距離も小さくなるため，PBSMT での翻訳は容易になる．しかし，手順の多くは英語側で命令文となっている．そのため，単語数が比較的短いときでも，長い距離での並べ替えが頻繁に起き，位置誤りが生じたと考えられる．以下の例は PBSMT の翻訳結果である．名詞を複数列挙するような文の一部であり，日本語側と同じ語順で訳出してしまっている．

> 4 の 鍋 に 1 の ブリ & 3 の 大根 & しいたけ & 生姜 を 入れ ,。
> Amberjack and daikon radish and shiitake mush-rooms , and add the ginger from step 1 to the pan from step 3 .

レシピの手順では複数の材料を列挙することが多くあり，このような文が多く見られる．複数の名詞が並ぶことによって，日本語側の動詞「入れ」と英語側の動詞 ‘add’ の並べ替えの距離が大きくなり，このような訳出になったと考えられる．また，「数詞 + の」に対応する前置詞句を正しい場所に訳出できていない誤りがある．これは，言語モデルから得られる確率には，数詞を含む前置詞句が訳出候補内で誤った位置に訳出されていても不確かさがないためである．「〜へ」や「〜の」のような他の前置詞句で表現されるものについても，同様の誤りが多く見られた．これらの誤りに対しては，句構造や依存構造を考慮するなどの構文情報を組み込んだ翻訳システムが対応可能であると考えられる．

一方，PBSMT と比較すると，NMT は置換誤りが多いことがわかる．置換誤りには，意味が近い単語を出力している誤りから，品詞のみが一致している単語を出

表 5.1　妥当性の誤り個数

手法	置換誤り	位置誤り	消失	未翻訳	挿入	妥当性一般	総数
PBSMT	49 (11.0)	98 (21.9)	139 (31.1)	23 (5.1)	95 (21.3)	43 (9.6)	447
NMT	102 (19.2)	20 (3.8)	176 (33.1)	0 (0.0)	114 (21.5)	119 (22.4)	531

力している誤りまで，様々なものがあった．例えば，前者では，「炒める」に対して ‘Heat’ を出力する誤りがあった．後者では，「キャベツ」に対して ‘sweet potato’ を出力する誤りがあった．頻度が少ない単語でも，翻訳候補の揺れが少なければ，PBSMT は正しく翻訳できる傾向がある．以下に例を示す．

入力: クリーム ツイスト
PBSMT: Twisted cream
NMT: Cream cream
（参照訳: Twisted cream bread）

これは，低頻度のフレーズに対して PBSMT が NMT より有効にはたらいた例である．

消失と挿入はどちらの手法にも多くあった．特に消失はどちらの手法においても最も大きい割合を占めた．以下に消失と挿入が PBSMT と NMT の両方で起きている例を示す．

ホーム ベーカリー の 生地 作り コース で 生地 を 作る 。
PBSMT: Make the dough in the bread maker to make the dough .
NMT: Make the dough using the dough setting .
（参照訳: Use the bread dough function on the bread maker to make the bread dough .）

消失や挿入が起きている文は，誤り箇所は異なるものの PBSMT と NMT で同じ文である傾向があった．PBSMT では「生地作りコースで」の消失と ‘to make’ や’the dough’ の挿入が見られる．一方，NMT では「ホームベーカリーの」の消失が見られる．このように，それぞれの手法で翻訳困難な文はある程度共通している

のではないかと考えられる．

また，挿入誤りは，日本語側で目的語が省略されている文に見られた．レシピの手順の日本語側では，同一レシピ内で一度出現した単語を省略することがある．そのような文の訳出で，省略された目的語の位置に何かしらの単語が挿入されることがあった．以下に例を示す．

> 紙 に 包ん で
> NMT: Wrap the cake in the cake paper
> （参照訳: Wrap the cakes in parchment paper）

この例では，‘the cake’ にあたる原言語文の単語は存在しないが，このフレーズが訳出されている．これは学習時の参照訳の省略度合いによるものだと考えられる．この例でも，参照訳は ‘the cakes’ を補完している．しかし，文によっては，他動詞でありながら補完していないものも多くある．従って，省略するか補完するかは外部から何かしらの形で情報を与える必要があると考えられる．

最後に，未翻訳は PBSMT でのみ考慮する誤りであるが，その割合は最も少ないことがわかる．今回用いたコーパスは語彙数が小さいため，訓練時に出現する語彙が評価セットのほとんどの語彙を含んだ．従って，評価セットに含まれる未知語の割合がわずかで，このような結果になったと考えられる．

5.1.2 流暢性

各手法の流暢性の誤り数を表 5.2 に示す．並べ替えの誤りは，妥当性の位置誤りの時と同様の原因で起きていると考えられる．ただし，妥当性での位置誤りと違って，流暢性における並べ替え誤りは日本語側の意味を考慮しない．よって，妥当性の位置誤りで誤りに分類されたものも流暢性の並べ替え誤りには該当しないため，誤り数は少なくなる．以下の例は，妥当性の位置誤りの例として示したものだが，誤りとなるのは ‘add’ のみとなる．

> Amberjack and daikon radish and shiitake mush-rooms , and add the ginger from step 1 to the pan from step 3

表 5.2　流暢性の誤り個数

手法	並べ替え	語形	機能語	文法一般	理解困難	総数
PBSMT	18 (14.0)	2 (1.6)	24 (18.6)	73 (56.9)	12 (9.3)	129
NMT	4 (4.8)	1 (1.2)	6 (7.2)	17 (20.5)	55 (66.3)	83

機能語の誤りは PBSMT で多く見られた．主な誤りは不要な前置詞の挿入であった．これは，フレーズ抽出で得られた前置詞について，適切な挿入場所が存在しなかったためであると考えられる．つまり，フレーズ抽出の時点で誤っていたと考えられる．以下の例では，‘in’ が不要であるとした．

Remove the sinew from the chicken tenders and fold in lightly .

文法一般についての主な誤りは基本的に内容語の誤りであった．特に，動詞や名詞の消失や挿入が多くの出力文で見られた．これも，機能語での誤りと同じ理由で起きていると考えられる．以下の例は動詞が消失したものである．

Basic chiffon cake milk to make the dough .

NMT には理解困難な文が非常に多く見られた．並べ替えや機能語，文法一般などの文法的な誤りはなくても，同じ単語・フレーズの繰り返しや，ある動詞に対して意味的整合性の取れない目的語が見られた．以下の例では，‘and open the pot’ が繰り返されている．

leave to steam for about 2 hours , and open the pot , and open the pot

語形については，各手法においてほとんど誤りが見られなかった．タイトルや材料は名詞句であり，手順の多くは命令文で表される．命令文における接続詞節での時制は現在形で表される．その時の主語はほとんどが材料を指す名詞であり，三人称単数である．ゆえに，時制の不一致や，主語と動詞の不一致が起きなかったと考えられる．

表 5.3　自動評価の結果 (BLEU/RIBES)

手法	タイトル	材料	手順	全種類
PBSMT	22.15 / 61.85	56.10 / **90.03**	25.37 / 74.98	28.09 / 81.72
NMT (single)	19.68 / 61.49	55.75 / 89.70	25.68 / 77.84	28.01 / 82.79
NMT (4 Ensemble)	**22.35 / 63.44**	**58.90** / 89.90	**27.64 / 79.29**	**30.05 / 83.66**

5.2　自動評価

評価セットにおける BLEU と RIBES での評価結果を表 5.3 に示す．NMT については，シングルモデルと複数のモデルを組み合わせたアンサンブルモデルの結果の二つを示す．

まず，タイトルについて議論する．タイトルには自由な語彙や意訳が多く見られる．言い換えれば，比較的低頻度な形態で書かれている．また，タイトルが占める割合は表 4.1 からわかるように非常に小さい．以上から，タイトルの翻訳は材料や手順の翻訳より困難であった．表 5.3 のタイトルの項目を見ると，PBSMT が NMT に対して BLEU でも RIBES でも良い結果を示している．PBSMT は数単語からなる単語列をフレーズとして翻訳するため，自由な語彙や意訳で記述されているタイトルでも，部分的に正しく翻訳できる．一方，NMT にこのようなテキストを入力すると，原言語文のどの単語も訳せていなかったり，極端に短い出力となってしまい，BLEU が低くなってしまった．

次に，材料の評価結果について議論する．材料は 3 単語程度と短い文であり，かつ，単語ごとに翻訳候補が少ない．そのため，PBSMT と NMT ともに非常に高い結果が得られた．このように，辞書引きのような翻訳が要求される文には PBSMT が優位であると考えられる．そのため，わずかではあるが，どちらの評価尺度においても PBSMT が上回る結果となった．

手順では，PBSMT の位置誤りの例としてあげたような複数の名詞を列挙する文が見られる．そして，目的言語文の文体は命令文であることが多く，並べ替えの距離が大きくなってしまう．このような場合，NMT の方が誤りが少ない．また、原言語文において省略が起き，目的言語文でその補完をしなければならない場合がある．PBSMT も NMT も，どの単語を補完するべきかという情報を明示的に与えて

いない以上，正しい単語を訳出するのは難しい．ただし、NMT では，何かしらの単語で補完する傾向が見られた．

次に，NMT のアンサンブルモデルについて述べる．表 5.3 より，タイトル，材料，手順のいずれにおいても，BLEU と RIBES の両評価尺度においてスコアが改善したことがわかる．ゆえに，本実験で用いたレシピドメインかつ比較的小規模のデータで学習された NMT のモデルによるアンサンブルでも効果が得られることがわかった．

最後に，RIBES について補足する．RIBES は NMT に有利な尺度となっている可能性がある．RIBES は，単語適合率に対する重み α と Brevity Penalty に対する重み β をハイパーパラメータとして決定する．一方で，BLEU は Brevity Penalty のみ考慮し，かつ，重みは決定せず倍率は 1 となる．NMT では，参照訳に対して短い文を訳出することが多くあるが，この β によってその問題が無視されうる．語順は正しいことが多いため β が低い際には高いスコアが出やすい．PBSMT は原言語文の単語・フレーズをもとに，目的言語側とのフレーズ対応を獲得し，それを並べ替えることで訳出する．そのため，NMT ほど極端に短い文を訳出することはほとんどない．しかし，並べ替える候補が増えるほど，正しい語順にして訳出するのは難しくなり，RIBES の高いスコアを得にくくなる．以上より，RIBES はハイパーパラメータ次第で NMT に有利となっている可能性がある．

5.3 PBSMT モデルの拡張

PBSMT の訳出候補をリランキングし，BLEU と RIBES で評価したものを表 5.4 に示す．BLEU については，訳出候補数 N=10 でリランキングしたものが，いずれの隠れ層の次元数においてもベースラインを上回る結果となった．その中でも，h が 256 の時に最も高くなり，0.50 ポイントの向上がみられた．一方で，N の増加にともない BLEU の値は減少する傾向があり，N=40 以降でいずれもベースラインを下回った．この原因としては，RNN 言語モデルによって，多くの候補の中からより流暢な文を選択しているが，意味の同じ異なる語彙を使っているために低い BLEU となっている可能性が考えられる．しかしながら，リランキングで得られた出力には BLEU の向上した条件も含めて，流暢性を損なった文が多くみら

表 5.4　PBSMT の N-best リランキング (BLEU)

次元数	N=1	N=10	N=20	N=30	N=40	N=50	N=100
Baseline	25.37						
h=32		**25.36**	25.13	25.02	24.89	24.54	24.42
h=64		**25.50**	25.46	25.46	25.25	25.27	24.82
h=128		**25.66**	25.47	25.50	25.16	25.34	25.07
h=256		**25.87**	25.64	25.74	25.23	25.30	24.82

表 5.5　PBSMT の N-best リランキング (RIBES)

次元数	N=1	N=10	N=20	N=30	N=40	N=50	N=100
Baseline	74.98						
h=32		**74.43**	74.33	74.01	73.74	73.85	73.65
h=64		74.43	**74.54**	74.05	73.79	73.98	73.86
h=128		74.03	**74.56**	74.37	74.04	73.56	73.09
h=256		**74.35**	74.10	73.81	73.80	74.13	73.65

れた．以下にその例の一つを示す．

> よく きれる 包丁で 真ん中 から 二つ に カット する 。
> ベースライン: cut in half with a sharp knife from the center .
> リランキング: with a sharp knife from the center and cut in half .
> 参照訳: use a sharp knife to slice the dough down the center .

この場合，参照訳の適合率に基づく BLUE では，どちらの出力に対しても近いスコアが与えられる．しかし，流暢性ではリランキングしたものの語順に誤りが見られる．つまり，RNN 言語モデルのリスコアでより流暢な出力を得ようとしたものの，結果的に不自然な文を選択してしまっている．原因としては，リランキングをするのに適切に機能するよう学習できていない可能性が考えられる．また，N の増加による BLEU の低下は，より多くの出力が上の例のようになり 3gram や 4gram

の適合率低下に起因すると考えられる．

RIBES については，リランキングしたものはいずれもベースラインを下回る結果となった．これは BLEU の時と同様，リランキングしたものに語順の誤りが含まれやすくなっているためであると考えられるが，RIBES は語順を考慮した評価尺度であるためその影響がより顕著となる．また，N の増加による RIBES の低下も，同様の理由が考えられる．

ここで，リランキングに用いた RNN 言語モデルは学習が不十分だった可能性を考える．ngram 言語モデルについては，[26] より，学習に用いた文数が多いほど良いモデルが得られることが報告されている．同様に，RNN 言語モデルにおいてもより多くの文数を学習に使うことができれば，流暢な文の選択ができるのではないかと考えられる．

第 6 章　関連研究

利用可能なレシピデータの増加に伴い，これまでにレシピを対象とした様々な研究がなされてきた．レシピの解析に焦点を当てた研究では次のようなものがある．Kiddon ら [27] は調理行動をノード，それらの関係をエッジとするグラフによってレシピを表現する手法を提案している．一方，Jermsurawong ら [28] は材料を終端のノード，調理行動を内部のノードとする木構造でレシピを表現している．Mori ら [6] はレシピを手続き文書とみなしてフローグラフとして表し，材料や調理器具，調理行動をノード, それらの関係をエッジとするグラフでレシピを表現している．また，Nanba ら [5] はレシピ解析に利用するため，料理用語に関する専用の辞書を構築している．上記の研究はレシピの基礎解析に焦点を当てたものであるが，情報検索や要約，推薦などの分野でレシピを扱った研究には次のようなものがある．Yamakata ら [3] は，複数のレシピに共通するグラフ構造を検出することで，レシピを要約する手法を提案している．Forbes ら [4] は，レシピの推薦における Matrix Factorization の有効性を検証している．Wang ら [29] は，中国語のレシピに対して，類似するレシピを検索する手法を提案している．

一般的に，レシピを構成する文の多くは構文的に簡易に記述されているものの，解析が困難な場合がある．例えば，あるレシピを構成する手順を対象とした場合，それぞれの手順は前後に依存関係をもち，目的語の省略，特に材料の省略が起きやすい．機械翻訳において，本論文で扱う日英言語対に見られるような，ある対訳文で一方の言語でのみ省略が起きていた場合，正しく翻訳するのは非常に困難となる．ゆえに，ある文で省略された名詞句を補完する処理であるゼロ照応解析が必要であると考えられる．省略された情報を適切に補完することができれば，レシピ翻訳でみられた誤りのいくつかを解決できる可能性がある．

機械翻訳の誤り分析に焦点を当てた研究を次に挙げる．Bentivogli ら [30] は英独言語対における PBSMT と NMT の誤り分析を行なった．彼らの研究は，初めて NMT の出力に対して詳細な誤り分析を行なったものであり，PBSMT と木構造に基づく機械翻訳の出力とどのような差異が見られるかを，様々な方法で検証している．用いられた自動評価尺度は BLEU と 2 種類の TER [31]，Human-targeted TER と Multi-reference TER であり，それぞれの尺度から誤りの傾向を見ている．

誤り体系に関しては，形態素誤り，語彙誤り，単語並べ替え誤りの 3 種類を導入している．なお，単語並べ替え誤りを細分類したものも用いており，品詞誤りや係り受け誤りが考慮されている．

最後に，本論文で用いたレシピコーパスを実験に用いている研究を挙げる．Ishiwatari ら [32] らはこのコーパスを用いて SMT における分野適用を行なった．彼らの研究では，レシピとは大きく異なる分野である，京都における日本の歴史や寺院に関するコーパスで機械翻訳モデルを学習した．次に，学習したモデルに単語分布表現を導入することで，レシピコーパスにおける未知語の翻訳を試みた．分野外のレシピコーパスにおける未知語翻訳の点で精度が向上したが，レシピそのものの翻訳に焦点を当てている本論文とは異なる．

第 7 章　おわりに

本論文では，レシピに対する日英機械翻訳を行なった．翻訳手法には PBSMT と NMT を用い，その出力における誤り分析を行なった．誤り体系には MQM ANNOTATION DECISION TREE を拡張したものを用いた．誤りを分類したところ，各誤りの傾向はそれぞれの手法において大きく異なることがわかった．PBSMT は NMT と比較すると文法的な誤りが多かった．一方で，NMT は PBSMT より置換誤りが多く，別の語義の単語を出力する傾向が見られた．また，NMT は文法的に正しいが，意味がとれない訳出も多かった．そして，どちらの手法でも消失と挿入が多く見られた．

レシピを構成する 3 種類の文では，それぞれにおいて異なる特徴が見られた．タイトルは分量が少ない割に語彙が多かったため，学習が難しかった．そのため、NMT によるタイトルの訳出には，原言語文をほとんど訳せていない，訳出が短いなどの問題が起きていた．一方，PBSMT では，タイトル全体を訳せていなくても，フレーズ対によって部分的に訳せていた．材料はタイトルや手順と比較すると非常に平易な文体であり，どちらの手法でも高い精度が得られた．これは，どちらの手法でも辞書引きのような翻訳が可能であることを示している．最後に，手順では PBSMT と NMT で異なった誤りの傾向が見られた．PBSMT は NMT と比較すると，多くの位置誤りが見られた．手順では複数の名詞を列挙した後，動詞が続く文がある．かつ，手順では目的言語文の多くが命令文で書かれている．そのため，原言語文の動詞と目的言語文の動詞で並べ替えの距離が大きくなり，位置誤りを起こしたと考えられる．NMT は文法的に正しい文を生成するため，PBSMT に比べ位置誤りは少ない．RIBES においても，NMT が約 1 ポイント上回る結果となった．また，RNN 言語モデルによる PBSMT の翻訳候補リランキングでは，BLEU での精度向上がみられたが，事例からは流暢性を増しておらず，RIBES も下がってしまっている．本論文で用いたコーパスでは，RNN 言語モデルを学習するのに不十分であった可能性が考えられる．一方で，複数の NMT のモデルをアンサンブルすることで，大きな精度向上がみられた．ゆえに，本論文で用いたようなレシピドメインかつ小規模なコーパスで学習されたモデルにおいても，アンサンブルの有用性が示された．

参考文献

[1] H. Maeta, T. Sasada, and S. Mori, “A Framework for Procedural Text Understanding,” Proceedings of the 14th International Conference on Parsing Technologies (IWPT 2015), pp.50–60, 2015.

[2] M. Yasukawa, F. Diaz, G. Druck, and N. Tsukada, “Overview of the NTCIR-11 Cooking Recipe Search Task,” Proceedings of the 11th NTCIR Conference (NTCIR-11), pp.483–496, 2014.

[3] Y. Yamakata, S. Imahori, Y. Sugiyama, S. Mori, and K. Tanaka, “Feature Extraction and Summarization of Recipes using Flow Graph,” Proceedings of the 5th International Conference on Social Informatics (SocInfo 2013), pp.241–254, 2013.

[4] P. Forbes and M. Zhu, “Content-boosted Matrix Factorization for Recommender Systems: Experiments with Recipe Recommendation,” Proceedings of the 5th ACM Conference on Recommender Systems (RecSys 2011), pp.261–264, 2011.

[5] H. Nanba, Y. Doi, M. Tsujita, T. Takezawa, and K. Sumiya, “Construction of a Cooking Ontology from Cooking Recipes and Patents,” Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing Adjunct Publication (UbiComp 2014 Adjunct), pp.507–516, 2014.

[6] S. Mori, H. Maeta, Y. Yamakata, and T. Sasada, “Flow Graph Corpus from Recipe Texts,” Proceedings of the 9th International Conference on Language Resources and Evaluation (LREC 2014), pp.2370–2377, 2014.

[7] P. Koehn, F.J. Och, and D. Maruc, “Statistical phrase-based translation,” Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT 2003), pp.48–54, 2003.

[8] D. Bahdanau, K. Cho, and Y. Bengio, “Neural Machine Translation by Jointly Learning to Align and Translate,” 5th International Conference on Learning Representations (ICLR 2015), 2015.

[9] A. Burchardt and A. Lommel, "Practical Guidelines for the Use of MQM in Scientific Research on Translation Quality," Technical report, QTLaunchPad, 2014.

[10] Peter F.Brown, S.A.D. Pietra, V.J.D. Pietra, and R.L. Mercer, "The mathematics of statistical machine translation: Parameter estimation," 1993 Association for Computational Linguistics, pp.263–311, 1993.

[11] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "BLEU: A Method for Automatic Evaluation of Machine Translation," Proceedings of the 40st Annual Meeting of the Association for Computational Linguistics (ACL 2002), pp.138–145, 2002.

[12] F.J. Och, "Minimum Error Rate Training in Statistical Machine Translation," Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL 2003), pp.160–167, 2003.

[13] I. Sutskever, O. Vinyals, and Q.V. Le, "Sequence to Sequence Learning with Neural Networks," In Advances in Neural Information Processing Systems 27 (NIPS 2014), pp.3104–3112, 2014.

[14] T. Mikolov, M. Karafiat, L. Burget, Jan "Honza"Cernocky', S. Khudanpur, "Recurrent neural network based language model," INTERSPEECH 2010, 2010.

[15] S. Hochreiter and J. Schmidhuber, "LONG SHORT-TERM MEMORY," Neural Computation 9, pp.1735–1780, 1997.

[16] K. Cho, B. vanMerrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation," Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp.1724–1734, 2014.

[17] Z. Tu, Z. Lu, Y. Liu, X. Liu, and H. Li, "Modeling Coverage for Neural Machine Translation," Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL 2016), pp.177–180, 2016.

[18] T. Kudo, K. Yamamoto, and Y. Matsumoto, "Applying Conditional

Random Fields to Japanese Morphological Analysis," Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014), pp.230–237, 2004.

[19] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, and R. Zens, "Moses: Open Source Toolkit for Statistical Machine Translation," Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions, pp.177–180, 2007.

[20] H. Isozaki, T. Hirao, K. Duh, K. Sudoh, and H. Tsukada, "Automatic Evaluation of Translation Quality for Distant Language Pairs," Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing (EMNLP 2010), pp.944–952, 2010.

[21] Y. Wu, M. Schuster, Z. Chen, Q.V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, J. Klingner, A. Shah, M. Johnson, X. Liu, ŁukaszKaiser, S. Gouws, Y. Kato, T. Kudo, H. Kazawa, K. Stevens, G. Kurian, N. Patil, W. Wang, C. Young, J. Smith, J. Riesa, A. Rudnick, O. Vinyals, G. Corrado, M. Hughes, and J. Dean, "Google' s neural machine translation system: Bridging the gap between human and machine translation," arXiv:1609.08144v2, 2016.

[22] J. Duchi, E. Hazan, and Y. Singer, "Adaptive Subgradient Methods for Online Learning and Stochastic Optimization," Journal of Machine Learning Research 12, pp.2121–2159, 2011.

[23] T. Luong, H. Pham, and C.D. Manning, "Effective approaches to attention-based neural machine translation," Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pp.1412–1421, 2015.

[24] S. Jean, K. Cho, R. Memisevic, and Y. Bengio, "On using very large target vocabulary for neural machine translation," Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, pp.1–10, 2015.

[25] T. Luong, I. Sutskever, Q. Le, O. Vinyals, and W. Zaremba, "Addressing the rare word problem in neural machine translation," Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, pp.11–19, 2015.

[26] T. Brants, A.C. Popat, P. Xu, F.J. Och, and J. Dean, "Large language models in machine translation," Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pp.858–867, 2007.

[27] C. Kiddon, G.T. Ponnuraj, L. Zettlemoyer, and Y. Choi, "Mise en Place: Unsupervised Interpretation of Instructional Recipes," Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP 2015), pp.982–992, 2015.

[28] J. Jermsurawong and N. Habash, "Predicting the Structure of Cooking Recipes," Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP 2015), pp.781–786, 2015.

[29] L. Wang, Q. Li, N. Li, G. Dong, and Y. Yang, "Substructure Similarity Measurement in Chinese Recipes," Proceedings of the 17th International World Wide Web Conference (WWW 2008), pp.979–988, 2008.

[30] L. Bentivogli, A. Bisazza, M. Cettolo, and Marcello, "Neural versus phrase-based machine translation quality: a case study," Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2016.

[31] M. Snover, B. Dorr, R. Schwartz, L. Micciulla, and J. Makhoul, "A Study of Translation Edit Rate with Targeted Human Annotation," Proceedings of the Biennial Conference of the Association for Machine Translation in the Americas (AMTA), pp.223–231, 2006.

[32] S. Ishiwatari, N. Yoshinaga, M. Toyoda, and M. Kitsuregawa, "Instant Translation Model Adaptation by Translating Unseen Words in Continuous Vector Space," The 17th International Conference on Intelligent Text

Processing and Computational Linguistics (CICLing), 2016.