Project Next における機械翻訳の誤り分析

赤部 晃一 ♠, Graham Neubig♠, 工藤 拓 ♡, John Richardson♣, 中澤 敏明 ♣, 星野 翔 ♢
♠ 奈良先端科学技術大学院大学, ♡ グーグル, ♣ 京都大学, ♢ 総合研究大学院大学

1 はじめに

現在の日英翻訳は何ができるのか？何ができないのか？本稿は、この疑問に答えるべく、Project Next の機械翻訳分析グループが取り組んだ誤り分析タスクの結果を報告する。

具体的には、システムの中身を考慮しないブラックボックス分析とシステムの中身を考慮するグラスボックス分析（2 節）を行い、各システムの誤りの分類を専用の誤り体系（3 節）に基づいて行う。分析の対象として、3 つの商用システム及び 3 つのオープンソースシステムを用いて、現代日本語書き言葉均衡コーパスに対する翻訳結果を用いる（4 節）。この分析に基づき、各システムの全体的な精度と、システムの誤り傾向の違いを明らかにする（5 節）。この議論に基づき、これから日英翻訳において解決すべき問題を議論する。

2 機械翻訳システムの誤り分析

誤り分析の際に、訳出の導出過程を考慮せず出力結果のみに着目した分析を**ブラックボックス分析**という。ブラックボックス分析は、商用システムのように訳出の導出過程が把握困難な場合にも利用可能である。一方、訳出の導出過程を分析対象としたものを**グラスボックス分析**という。グラスボックス分析をするためには、分析対象とするシステムの仕組みが把握可能でなければならず、商用システムの分析には向かないが、ブラックボックス分析に比べて誤りの原因をより具体的に把握できるため、分析結果がシステムの改善に直接反映されることが期待できる。

本稿では、オープンソースの機械翻訳システムと商用の機械翻訳システムに対して誤り分析を行う。商用のシステムはシステムの導出過程を把握することが困難なため、ブラックボックス分析のみを行う。一方、オープンソースのシステムに対してはブラックボックス分析に加え、グラスボックス分析を行うようにする。次節では、両分析手法を利用する際の誤り体系について説明を行う。

3 誤り体系

機械翻訳システムに含まれている誤りを分類することは、システムにどのような誤りが含まれており、どのような改善が必要かを客観的に把握する上で不可欠である。文献 [4] では階層化された機械翻訳の誤り体系が提案されている。本稿ではこの誤り体系を出発点として、ブラックボックス分析の誤り体系を定義する。図 1 は本稿で実際に使用するブラックボックス分析の誤り体系である。

単語の削除 … 必要な単語が欠落
- 内容語の削除
- 機能語の削除

並べ替え … 不正確な語順

単語の誤り … 本来とは異なる単語
- 語義選択誤り
  - 置換誤り (文脈非依存)
  - 語義曖昧性解消 (文脈依存)
- 語形・活用誤り
- 挿入誤り
- スタイル誤り
- 慣用句の直訳

モダリティ
- 否定
- 可能
- 受け身

未知語 … 翻訳がモデルに存在しない
- 未知語幹
- 未知活用形
- 翻字必要（本研究で導入）

句読点

図 1 ブラックボックス分析の誤り体系

この中で「モダリティ」誤りは、書き手の立場に則した表現方法が失われている場合に分類される。具体的には、「能動態」と「受動態」の誤りや、英語の “can” や “may” で表現される「可能」「許可」、“must” で表現される「義務」などが挙げられる。これらの誤りは、単語の削除誤り、単語の誤り、並べ換え誤りの複合と考えることが出来る。このため、分類を行う際にはモダリティ誤りを優先的にアノテーションするようにする。また、「翻字必要」は本稿で新たに導入した種類の誤りである。これは、日本語の固有名詞をローマ字表記に置き換える場合等が当てはまる。

一方、オープンソースの翻訳システムは、翻訳結果の導出過程が把握可能なため、図 1 に示した誤り体系よりも詳細な誤りの分類が可能である。本稿では、図 2 に示すグラスボックス分析のための誤り体系を定義し、オープンソースの翻訳システムではグラスボックス分析も行うようにした。

4 実験設定

4.1 分析対象のシステム

本稿の目標は、現在の最先端の日英翻訳システムがどのような誤りを起こすかを調べ、今後の課題を特定す

前処理誤り
├── 形態素解析・トークン化
├── 構文解析
├── 文分割
├── 事前並べ替え
└── 事前編集（主語挿入等）
ルール抽出
├── 誤ったルールがモデルに存在
├── 必要なルールが存在しない
└── 必要なルールのスコアが低い
モデル化誤り（悪い候補に高いスコア）
探索誤り
原文の誤り
├── 原文のスペル誤り
├── 原文の文法誤り
├── 原文のスタイル（口語，古文）
└── 直訳が難しい

図 2　グラスボックス分析の誤り体系

ることである。このため、システムの種類や学習データに制約を与えず、なるべく様々な翻訳方式をカバーするシステムを評価対象とした。評価の対象とした 6 つのシステムは下記の通りである。

4.1.1　商用システム

まず、3 種類の商用システムを使用した。すべてのシステムは一般ユーザーがウェブ経由で使用できるもので、そのゆえにオープンドメインの入力を想定していると言っても良い。商用システムであるため詳細な中身は明かされていないが、おおよその分類として、ルールベースシステムが 1 つ（RBMT）、統計ベースシステムが 2 つ（それぞれ SMT1、SMT2）である。

商用システムは総じて、一定の開発コストを掛けているため、前処理や後処理など細かい調整がある程度行われていると想定できる。また、ルールベース翻訳と統計ベース翻訳の違いをまとめると、ルールベース翻訳は言語学者が構築したルールに基づくため、短く規則的な文で安定した精度が期待できる。これに比べて、統計翻訳は多少安定性に欠ける一方、文体が崩れた文、専門用語を含む文、大規模な統計により曖昧性が解消しうる文などに対して比較的高い精度となると期待できる。

4.1.2　Moses

次に、統計翻訳で最も代表的なオープンソースソフトである Moses で、フレーズベース翻訳のシステム [1] を作成する。フレーズベース翻訳は、図 3(a) のように、文を数単語からなる単語列に分割し、この単語列を翻訳して並べ替えることで文を生成する。フレーズベース翻訳の利点は、構文解析などの高度な言語処理ツールを必要とせず、比較的簡単に作成できるところにある。その一方、一般にフレーズベース翻訳は並べ替えを多く必要とする言語において、精度が低下するとされている。これは並べ替え確率の推定が困難となることや、計算量上の理由で正確な並べ替えを実現できない場合があるためである。

4.1.3　Travatar

また、Tree-to-string 翻訳に基づくツールキットである Travatar を使ったシステムも構築する [2]。Tree-to-

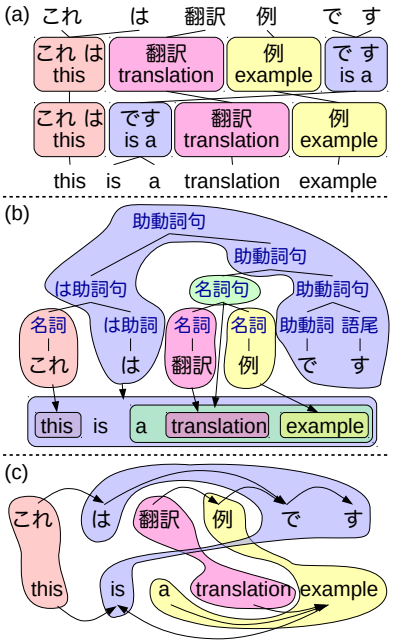

図 3　(a)Moses のフレーズベース翻訳 (b)Travatar の tree-to-string 翻訳 (c)Kyoto EBMT の用例ベース翻訳

string 翻訳は、図 3(b) のように、まず原言語文に対して構文解析を行い、この構文木に対する翻訳ルールを用いて翻訳を行う仕組みである。構文解析により得られる文の構造を翻訳の過程で利用することにより、主に 2 つの利点がある。まず、構文解析結果で翻訳に考慮する仮説の空間を小さくすることで、計算量上の理由で並べ替えを制限する必要がなく、文全体にわたり並べ替えを行うことが可能である。また、構文解析結果が正しければ、この構文解析結果に合った翻訳結果を生成するため、出力がフレーズベース翻訳に比べて文法的であることが多い。その一方、構文解析結果の誤りや、文法に沿わない意訳が原因で精度が低下することもある。

4.1.4　Kyoto EBMT

最後に両言語の依存構造に基づく用例ベース機械翻訳システム Kyoto EBMT を用いた実験も行った [3]。図 3(c) のように、まず原言語文に対して依存構造解析を行い、これにより得られる依存構造木を目的言語の依存構造木へと変換する。両言語の構文情報を用いるため、Kyoto EBMT は他の翻訳システムに比べて、両言語の構文構造が類似している文において文法的な出力が生成できるという仮説を立てることができる。逆に、両言語の構文構造が合わず直訳が難しい文において、Moses や Travatar のような、より柔軟性の高い定式化を用いているシステムに比べて精度が低下する可能性も考えられる。

4.2　テストデータ

機械翻訳システムのテストデータには、現代日本語書き言葉均衡コーパス（BCCWJ）の一部を利用した。

これをテストデータとして選択した理由は主に下記の 2 つである。まず、Project Next の他の分析グループは同じデータを用いて分析を進めているため、知見の共有を促進すると考えられる。また、BCCWJ はオープンドメインであるため、特定の分野に対するシステムの得意苦手による影響を除外することもできる。

BCCWJ は本来、日本語のみからなる単言語コーパスである。しかし、翻訳機の精度比較や、機械翻訳と人手翻訳の違いを明らかにするためには、対訳データがあることが望ましい。このため、英語を母語とする著者 2 人が計 818 文を文脈を考慮しながら英語に翻訳し、機械翻訳システムとともに人手による精度評価を行う。

4.3 学習データ

用例ベース翻訳システムや統計的翻訳システムを作成するのに、学習データが必要である。商用システムに関しては、学習データは制限されておらず、どのデータを使っているかが明確ではない。

オープンソースソフトに基づくシステムは、「日英対訳コーパス」サイト[*1]を参考に、様々な分野の翻訳データを用いて学習した。具体的には対訳コーパスとして、例辞郎例文、京都フリー翻訳タスクの Wikipedia データ、田中コーパス、日英法令コーパス、青空文庫、TED 講演、BTEC、オープンソース対訳を利用した。また、辞書として英辞郎、WWWJDIC、Wikipedia の言語リンクを利用した。これをすべて合わせて、コーパスとして 255 万文、辞書として 277 万エントリーとなった。

4.4 分析方法

各システムの学習を行ってから、テストデータに対して翻訳結果を生成し、分析の対象とする。

まず、各翻訳結果に対して 0〜6 の 7 段階人手評価を行う。なお、評価は参照文を提示せずに行い、人手翻訳結果も評価対象とする。このため、各文に対して、6 通りの翻訳システムと 1 通りの人手翻訳を評価することになる。結果の提示順序の影響を取り除くために、7 通りの翻訳結果をランダムな順で提示し、また複数の結果が同一である場合、1 回のみ提示することとした。なお、人手評価を行ったのは、本稿の研究内容を知らない、日本語と英語に精通している評価者である。

人手による数量評価が終わってから、機械翻訳システムのスコアの平均が低い文の順に、本稿の著者が誤り分析を行った。具体的には、すべてのシステムに対して、3 節の誤り体系に基づいて、ブラックボックス分析を行い、問題を特定した。また、オープンソースシステムに対して、グラスボックス分析も行い、その傾向をまとめた。次節では、この分析の結果について述べる。

5 分析結果

5.1 各システムの総合評価

まず、各システムの総合的な人手評価結果を図 4 に示す。図 4(a) には、システムの平均評価値を示し、図 4(b) に評価値 4〜6（良い）、2〜3（理解可能）、0〜1（理解不可能）の内訳を示す。この中で、Moses と KyotoEBMT、Travatar と SMT1 の間に有意差がなく、それ以外のシステムには有意差があった。

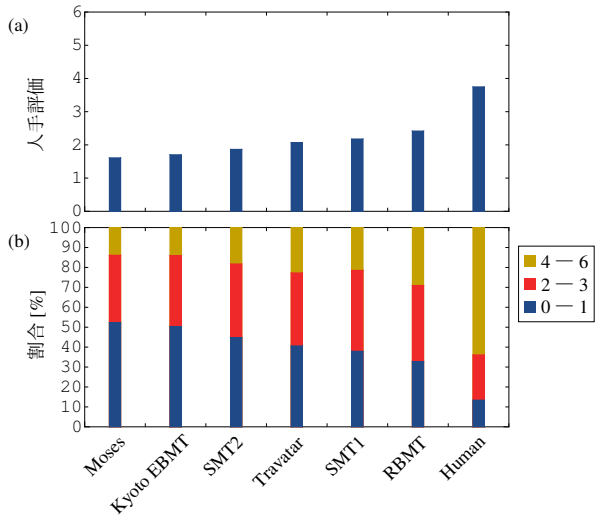

図 4 (a) 各システムの平均評価値 (b) 各システムの評価値の内訳

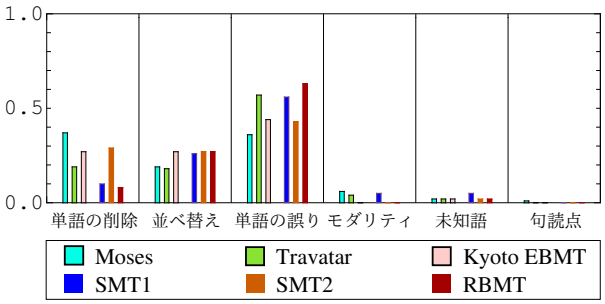

図 5 ブラックボックス分析における誤り傾向

この結果から、機械翻訳全体で RBMT は最も精度が高いことが分かる。また、オープンソースシステムの中では Travatar が最も高く、商用システムの SMT1 と同程度であった。しかし、いずれのシステムにおいても、文の 3 分の 1 以上は理解不能となっており、課題が残ることも分かる。

人手による翻訳の結果に目を向けると、すべての機械翻訳システムを大幅に上回っていることが分かる。しかし、平均評価値が 4 を下回っており、人手翻訳でも厳しい評価がされている。人手翻訳が低い評価値になっている文を分析したところ、原因は主に 1) 人手翻訳は文脈を用いて行っているにも関わらず、翻訳の評価は文脈を考慮しておらず、翻訳時と評価時に差が生じたこと、2) 単純な翻訳誤り、もしくは 3) 原文はそもそも曖昧で、直訳することが難しいことから起因した。

5.2 各システムの誤り傾向

5.2.1 ブラックボックス分析

まず、ブラックボックス分析の結果を図 5 に示す。この中で、モダリティー・未知語・句読点誤りはいずれのシステムにおいてもほとんど検出されなかった。モダ

[*1] http://phontron.com/japanese-translation-data.php

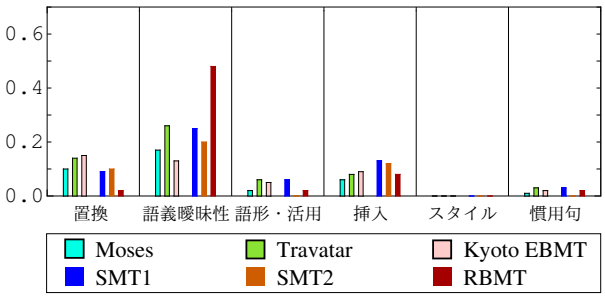

図 6　各翻訳システムに含まれる単語誤りの内訳

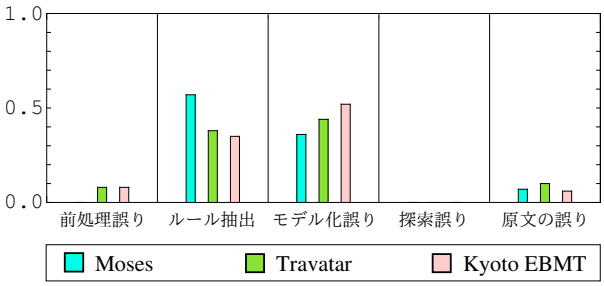

図 7　グラスボックス分析における誤り傾向

リティーと未知語に関しては、これらの誤りはその他の誤りより珍しいことに起因すると考えられる。句読点に関しては、今回の評価対象を評価の低い文にしたため、句読点以外の誤りが目立ったことが考えられる。

次に、削除誤りに目を向けると、興味深い傾向が見られる。具体的には、Moses、Kyoto EBMT と SMT2、Travatar、SMT1、RBMT の順に削除誤りが減っていく。この順は、人手評価の順と同等であり、単語の削除はシステムの人手評価に比例することが分かる。特に内容語の削除によって文の意味が大きく損なわれるので、直感に合った結果であると言える。

また、並べ替え誤りの比率を見ると、Moses は比較的少ないことが分かる。フレーズベース翻訳は一般的に並べ替えに弱いと言われるため、直感に反する結果であるとも言える。しかし、これは並べ替え問題がないことを意味しているわけではなく、Moses の結果にあまりにも多くの単語に関する誤りが含まれているため、並べ替え誤りの発見が困難であり、単語に関する誤りに偏って発見されることを指しているだけである。

次に、表 6 に、最も頻度が高かった単語に関する誤りの詳細な分析結果を示す。まず、文脈に依存しない置換誤りに着目すると、オープンソースシステムに比べて商用 SMT システムは置換誤りが少ないことが分かる。これは、オープンソースシステムに比べて、商用システムが大規模な学習データを利用しており、誤った翻訳ルールを学習する比率が少なくなっていることが原因であると考えられる。更に、RBMT のシステムはほぼ文脈依存の置換誤りを起こさないことが分かり、安定した翻訳ルールを用いていることが分かる。

その一方、RBMT システムで圧倒的に多かったのは語義曖昧性による誤り（つまり文脈依存の置換誤り）である。この理由としては、2 つが考えられる。まず、SMT システムは統計情報を用いて周りの文脈で曖昧性を解消しているのに対して、RBMT システムはこのような統計情報を取り入れていない（もしくは限定した形でしか取り入れていない）ことが考えられる。もう 1 つの理由として、RBMT システムは安定した動作で他の誤りが比較的少なく、まだ未解決問題である語義曖昧性の数が相対的に多く見えることも可能性としてある。

5.2.2　グラスボックス分析

図 7 にオープンソース機械翻訳システムのグラスボックス分析の結果を示す。この結果から、各システムに共通してルール抽出誤りとモデル化誤りが非常に多いことが分かる。

まず各システムに共通してルール抽出誤りが多く見つかった原因として、単語アライメント（両言語間の単語対応）を抽出する際の問題が考えられる。翻訳ルールを抽出する際、各システムともに単語アライメントの情報を利用するが、この情報は大量の機械翻訳の学習データから EM アルゴリズムによって自動生成される。その際、生成されたアライメント情報に誤りが含まれていると、誤った翻訳ルールが抽出される。このことから、ルール抽出誤りを削減するためには、単語アライメントの精度改善が必要といえる。ルール抽出誤りについて各システムを比較すると、Moses は他の 2 システムに比べ誤りが多く発見されていることが分かる。この原因として、Moses が構文情報を利用せずにルールテーブルを参照することが挙げられる。各システムとも同様に誤ったルールがルールテーブルに存在すると考えられるが、Travatar システムと Kyoto EBMT システムでは挿入位置に適した翻訳ルールを構文情報を利用して選択できる。一方 Moses では翻訳モデル・並べ替えモデル・言語モデルといった比較的貧弱な情報によってのみ翻訳ルールを選択するため、誤った翻訳ルールを選択しやすいと言える。

モデル化誤りは、利用した翻訳ルールが誤っているが、そのルールがルール抽出誤りでない場合（文脈依存のルール選択誤りである場合）に分類される。各システムを比較すると、Travatar と Kyoto EBMT は Moses に比べてモデル化誤りが多く見つかっているが、これは文脈非依存の誤りが単純に文脈依存の誤りに改善されたためと考えられる。

6　まとめ

本稿では、6 つの日英機械翻訳システムに対する誤り分析結果を報告した。まず、日英のオープンドメイン翻訳に対して、商用のルールベース翻訳システムに次いで、商用の SMT システムとオープンソースの tree-to-string 翻訳システムが最も高い精度となった。その一方、各システムとも人間の翻訳者には及ばず、機械翻訳に多くの課題が残っていることが分かった。特に、文脈依存の語彙選択は評価の良い翻訳システムであれ

ばあるほど大きな割合を占めていたことから、文脈を取り入れた語彙選択法は今後の有望な研究課題であると言える。今後は、更なる詳細な分析と分析結果を取り入れた新たな手法の提案に取り組んでいく。

参考文献

[1] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst. Moses: Open source toolkit for statistical machine translation. In *Proc. ACL*, pp. 177–180, 2007.

[2] G. Neubig. Travatar: A forest-to-string machine translation engine based on tree transducers. In *Proc. ACL Demo Track*, pp. 91–96, 2013.

[3] J. Richardson, F. Cromières, T. Nakazawa, and S. Kurohashi. Kyotoebmt: An example-based dependency-to-dependency translation framework. In *Proc. ACL*, pp. 79–84, 2014.

[4] D. Vilar, J. Xu, L. F. d'Haro, and H. Ney. Error analysis of statistical machine translation output. In *Proc. LREC*, pp. 697–702, 2006.