

総説・解説

博物学分野でWeb上のAI翻訳から和英対訳を得るシェルスクリプト

高崎浩幸[1]

Shell scripts tuned for natural history to retrieve AI Japanese-English translation on the web in parallel format

Hiroyuki TAKASAKI[1]

Abstract: This article shows shell scripts that access a free AI translation demo URL on the web and retrieve the translation output via Linux command line. The source text prepared beforehand is submitted to the external AI translator. After dividing the text into sentence units, mild but simple "encryption" is applied by shuffling the sentences, later to be decrypted for reconstructing the text after translation. The sentences with pre-replacement of technical terms are fed, and automatically corrected for common error patterns in the output translation. The source text in Japanese and the output AI-translated text in English are arranged in a parallel Japanese-English sentence-by-sentence format. These tasks are combined into a single shell script operated with one command. In the parallel format, the translation can be easily checked and corrected in post-editing. Then another command is executed to extract only the post-edited output English sentences to be concatenated into paragraphs. Thus, it is much easier to translate Japanese into English. Using natural history as an exemplary field of science, this article explains how technical terms can be replaced from Japanese to English before submission of the source to the external AI translator. Once the structure and mechanism of the shell scripts used here are grasped, a system can be remade for any other field of science in which no tuned AI translator will ever be commercially supplied due to the demand too low. By implementing the present system using Linux running on a USB flash memory, any PC (whether Windows, Mac, or Linux) can be turned into an AI translator tunable to the user's preference by simply booting the PC with the USB memory. Once the PC is shut down and the USB drive unplugged, and rebooted, the PC will be back to the original state. This article explains also how to prepare such a booting USB flash drive.

I. はじめに

自然科学者のなかでも日本の博物学分野の研究者たちは，「学名をはじめとする大量の専門用語の翻訳が容易ではない」というハンディキャップに長年にわたって苛まれている．研究成果を現代科学の世界共通語である英語で発信しにくい状況にある．本稿は，AIや情報処理の素人による，素人のための，現状打破を目指しての試みである．

2018年頃まで，いわゆる機械翻訳であるGoogle翻訳などは，とても学術論文の英訳には使えなかった．ところが，2019年には工夫次第で使えるAI翻訳システムが出現した．国立研究開発法人情報通信研究機構（NICT）の研究成果を一部に利用した「みらい翻訳」である．さらに2020年３月下旬には，ドイツの企業が開発した「DeepL翻訳」で日本語も使えるようになった．すでに公開されている学術情報の和訳などは，DeepLデモURL（字数制限5000文字；https://www.deepl.com/ja/translator；2020年10月３日閲覧）を使えば十分だろう．しかし未発表の学術情報の英訳となると事情は異なる．

無料お試し「みらい翻訳」のデモURL（https://mirai-translate.com/trial；2020年10月３日閲覧）で，サイトのページ片隅の「利用規約」を開いて適当にコピー＆ペーストして英訳させると，「こんな訳は素人には無理」というレベルの出力が現れる．原文言語に日本語を選び，訳文言語に英語を選んで，「巨人の肩に乗って」と原文入力し，翻訳ボタンにカーソルを合わせてクリックすれば，「on the shoulders of giants」という訳が出力窓に出る．冠詞や名詞の単数形・複数形の区別も慣用句通りで，これなら多少の工夫を加えれば実用に堪えそうである．

使い勝手をよくするために，前もって準備したテキストファイルを指定して操作できるようにする．「みらい翻訳」が手に負えない専門用語については，原文の送出前に専門用語を置換することで対処する．誤訳の点検には，１文ずつ原文と訳文を容易に比較・修正できるようにする．さらに原文に難読化（簡単な“暗号”化）をほどこして，訳文回収後に復号する．これらのことができるスクリプト集を本稿では扱う．

[1] 岡山理科大学理学部動物学科，〒700-0005 岡山県岡山市北区理大町1-1. Department of Zoology, Faculty of Science, Okayama University of Science, 1-1 Ridai-cho, Kita-ku, Okayama-shi, Okayama 700-0005, Japan.
*Correspondence: Hiroyuki TAKASAKI, Email: takasaki@zool.ous.ac.jp

「非力な小人」のPC端末のコマンドラインから，「みらい翻訳」（以下，MT）すなわち「怪力無双ながら博物学の細部には歯が立たない巨人」たる無償デモURLの「肩にのる」工夫の概略を，GNU/Linuxのシェルスクリプトとともに，和文英訳の場合に限定して本稿では公開し，そのシステム導入とシェルスクリプトの解説をする．

博物学分野に限って仕掛けをブラックボックスとして使うだけなら，Linux Mint 19.3（Tricia，2023年までのLTS）などで，本稿に紹介するシェルスクリプト集「mistletoe」を動かせる．あるいは起動用USBメモリLinuxを作って，USBメモリを挿したPCをそのOSで起動するだけで使える．すなわち，「**Ⅱ．USBメモリLinuxの準備**」，「**Ⅲ．スクリプト集mistletoeのダウンロードとRPDの準備**」，「**Ⅳ．スクリプト集mistletoeの使用法**」だけを読めばよい．

残りは，この「mistletoe on MT」複合システムを使いこなすための補足情報である．普通のGNU/Linux機でコマンドモードを操作できる場合は，mistletoeをダウンロードして適切に設定すれば使える．

各自の使用実態に応じてチューニングしたい場合，参考書籍としてLinux（たとえば，奈佐原 2016，中島 2018，隔月誌「日経リナックス」）やシェルスクリプト（上田 2015），AWKの入門書（中島ら 2015）などを必要に応じて参照されたい．使い方に関する質問は，執筆者は受けかねる．また，スクリプトの利用者による行為およびその行為の結果についても責任は負わない．なお本稿で扱うスクレイピング手法（Mitchell 2016 参照）については，節度をもった利用が求められる（「Webスクレイピングの注意事項一覧」https://qiita.com/nezuq/items/c5e827e1827e7cb29011；「【2020年度版】個人用クローラーの開発手順とその注意点」https://qiita.com/nezuq/items/9e297d0e3468c24e8afa；2020年９月29日閲覧）．

Linuxのコマンドラインでの使用時に，自家製スクリプト**script.sh**の実行を「**./script.sh**」と略す設定法やその記法は，本稿では省き，「**bash script.sh**」に統一した．素人のつまづきを，極力，回避するためである．玄人から見れば無駄な箇所も多々あることだろうが，「目的を達するスクリプトであれば十分，素人作の無骨なプロトタイプ」として報告する．

Ⅱ. USBメモリLinuxの準備

各種LinuxやWindows，MacなどPCの機種ごとにコマンドラインからMTの翻訳作業に使うための操作や設定の説明はここでは記さないが，USBメモリから起動できて，シャットダウン後も設定の変更が残る「USBメモリLinux」をPCに装着して使用できる方法を解説する．そのようなOSの１つが，ここに紹介するDebianの軽量Linuxディストリビューションである（以下，RPDと略記；https://www.raspberrypi.org/downloads/raspberry-pi-desktop/；2021年１月３日ダウンロード確認）．

同OSの日本語解説「Raspberry Pi Desktop（Buster）をUSBフラッシュドライブから起動して使う」（https://qiita.com/ht_deko/items/1b21ea24b7670ad188f5；2020年10月３日参照）を参照しながら，USBメモリLinuxの準備をしていただきたい．RPD（本稿での使用はネット接続が前提）のUSBメモリへの書き込み後，「ハードウェアクロックをローカルタイムに変更」までは，上記ネット記事の説明に従って，各自のPC環境に合わせた設定を行う．他の設定は「mistletoe on MT」の利用には，省略してよい．

RPDを書き込んだUSBが準備できたら，このUSBでPCを起動してみよう．外部USB接続のOSから優先起動するように，PCの起動（boot）オプションを設定変更する．機種ごとのマニュアル記載に従って，起動あるいは再起動時に［ESC］や［DEL］，［F1］，［F2］，［F7］，［F12］，［option］キーを押すなどして変更すればよい．

ここで重要な注意は，できあがったUSBメモリRPDでPCを起動するにあたっては，起動の最初の画面で第１の選択肢「Run with persistence」を選ぶことである．決して「Run and reset persistence」で起動してはならない．設定「記憶」が消え，すべてが初期状態に戻り，再設定することになる．

RPDが立ち上がると，画面の上端に横バーが現れる．その左端にあるキイチゴ印をクリックすれば，アプリケーションメニューが現れる．キイチゴ以外にも，青い地球儀のChromium「Web browser」やファイルマネージャ，端末「LXTerminal」のボタンもあり，それらをクリックすれば，画面上に対応する窓が開く．シャットダウンするには，画面左上端のキイチゴ印をクリックして，「ログアウト」をクリック，最後に「Shutdown」をクリックする．電源が切れたところで，USBメモリを抜いて起動しなおせば，PCは元の状態に戻る．

メンテナンスには，時々あるいは定期的にRPDのアップデートやアップグレードを行って，OSを最新状態に保つ．そのためには次の２つのコマンドを，端末「LXTerminal」から実行すればよい．

```
$ sudo apt update
$ sudo apt upgrade
```

ウイルススキャンのソフトも入れたい場合には，clamavを入れておくとよいだろう．「Debian clamav」で検索すれば，情報が得られる．

なお，端末でコマンドラインを使うときに，打ち間違って意図したとおりに実行されない場合には，［PgUp］キーで直前の間違いコマンドを端末に出して，［←］キーでカーソルを間違い箇所に合わせ，その部分を消したり打ち直したりして，［Enter］キーで実行する．

III. スクリプト集mistletoeのダウンロードとRPDの準備

圧縮した「mistletoe」(**mistletoe3.1.zip**)は，URL(https://drive.google.com/file/d/1rmX-EE_e2d91xoDk6kIjJ72pY3gcAa7J/view)からダウンロード・展開して使用する．なお，MTをコマンドラインから使うためのスクリプトには，Python3で書かれた**A_mirai.py**(y-ichi氏 2019，https://gist.github.com/y-ich/41f70bf1f98fab37b-43f93b8b7d32c1d;2019年12月30日ダウンロード)があるが，少なくとも2021年１月まではメンテナンスが行われていない．スクリプト集mistletoeには，代わりにパッチを当てた英語版MTデモURL(https://miraitranslate.com/en/trial/;2021年１月11日閲覧)にアクセスする改変版を**future.py**として入れてある．

この**future.py**をRPDで使うには，少々の事前準備が必要である．mistletoeを動かすのに必要なパッケージは，python3(+ pip)，selenium(ウェブクライアント)，chromium-driver(google-chromeのブラウザコンポーネントをseleniumにつなぐ)，gawkのほかにbash，sed，grep，cut，teeなどの標準コマンド，さらにperl(改行まわりの正規表現の処理用)，nkf(原文の出所によらず文字コードを現在のGNU/Linux標準のutf-8に統一的に変換する)等である。RPD以外のLinux/Unixを使う場合など，必要に応じて不足するパッケージをインストールする．

RPDには，最初からPython2とPython3がそれぞれのパッケージ管理ツールと一緒に入っていて，端末での**python**は，Python2(日本語の使用が不得手)を意味する．他方，Python3の動作をさせるには，**python3**を使う．そのパッケージ管理ツールは**pip3**である．

ウェブブラウザーのGoogle Chrome/Chromiumを使うにあたって，RPDではseleniumとchromium-driverをインストールする．そのために次の２つのコマンドを端末「LXTerminal」で実行する．

```
$ pip3 install selenium
$ sudo apt install chromium-driver
```

他に，mistletoeを動かすのに必要なコマンドユーティリティで，RPDに足りないのはnkfとgawk(RPDに付属のmawkに代えて日本語の扱いにしくじらないGNU/Linux標準のAWK)の２つだけである．インストールには，次のコマンドを端末で実行する．

```
$ sudo apt install nkf
$ sudo apt install gawk
```

アクセサリーに入っている mousepad「Text Editor」(以後「エディタ」と略記)は，各自の好みに応じて設定を調整しておく．出力の点検と修正には，このエディタを使うので，その環境設定の良否が本システムの使い勝手を左右する．

エディタの設定は，[編集(E)]を左クリックし，その最下端の「設定」を選択して行う．推奨設定は，「表示」は「行番号を表示する」から「行を折り返す」まですべてチェックを入れてON，行の右マージンの桁数は旧来の英作文時の定番だったタイプライターでの標準に倣って「65」，カラースキームは「コバルト」とする．「エディター」の項目ではタブの幅「4」，タブモードは「タブを挿入」とし，あとは元のままでいじらない．「ウィンドウ」ではノートブックタブ以外にすべてチェックを入れる．

IV. スクリプト集mistletoeの使用法

ダウンロードして展開したディレクトリmistletoeをデスクトップにでもコピー＆ペーストする．ディレクトリmistletoeに，カーソルを合わせてマウスを右クリックして「端末を開く(L)」を左クリックし，開いた端末(LXTerminal)で，次のコマンドを実行して動作テストをする．

```
$ python3 future.py JA EN "何？"
```

しばらく待って「What?」と端末に英訳が返れば，これで端末には和英辞書相当以上の機能が実現できている．左半角引用符「"」と右半角引用符「"」の間に和文を書き込んで実行すれば英訳される．そのようにできたところで，mistletoeのサブディレクトリ「**./sources**」に前もって入れてある翻訳用の実例原文**how2use.txt**で次のコマンドを試す．

```
$ bash mistjaen.sh how2use.txt
```

テストが成功すれば，**how2use.txt**を入力とした英訳結果が端末に返り，エディタが開いて，その中に編集可能な和英対照形式で出力される(図１)．エディタを閉じれば，端末の表示に戻る(エディタをコマンドラインから開くと，Gtk-WARNINGが表示されることがあるものの実害はなく，無視してよい)．

次のメッセージが，端末に示されることもある．この場合，エディタは開かない．

```
The source text was sent to MiraiTranslate.
But the busy MiraiTranslate returned no reply.
Try again!
```

MTが混み合って「出力が返されなかった」ということなので，[PgUp]キーで直前のコマンド「**bash mistjaen.sh how2use.txt**」を端末に出して[Enter]キーで実行する．出力が返されてエディタが開き，出力が現れるまで

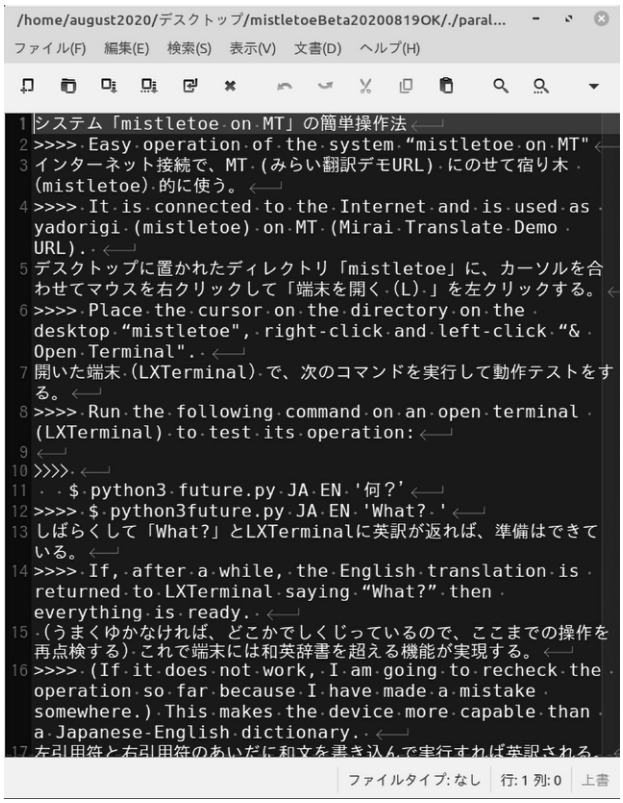

図 1 . 和英対照形式出力例. 「**./sources/how2use.txt**」を入力としてエディタに自動出力される編集可能な「**./parallel/edited_parallel_how2use.txt**」.

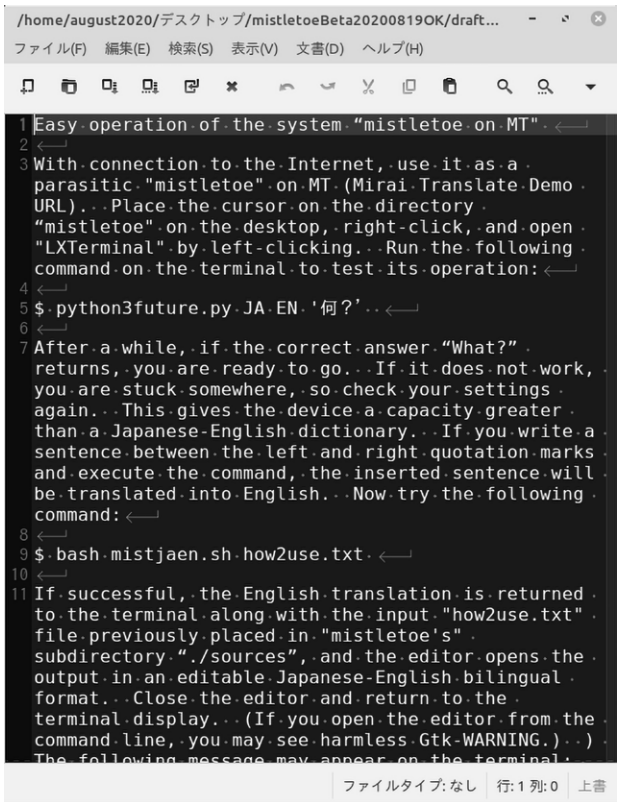

図 2 . 英文抽出出力例. 和英対照形式(図 1)での英文出力だけを点検・修正・保存後, **depara.sh how2use.txt**で英文出力だけが抽出され, 段落ごとに自動連結されてエディタに示される編集可能な「**./draft/draft_how2use.txt**」.

同じコマンド操作を繰り返す.

出力が返れば,「**./parallel**」サブディレクトリのなかに入力原文と出力翻訳を対照させたパラレル出力ファイルが, **parallel_how2use.txt**として格納される. それぞれ１文ずつ交互に, 入力原文が左端から記号なしに始まり, 出力翻訳は「**>>>>**」で始まる. エディタが開いて編集可能な状態で示されているのは, このファイルのコピー**edited_parallel_how2use.txt**である.

パラレル出力ファイルの原文と翻訳を, 1 行ごとに比較点検して, エディタを使って修正を加えて, 保存してエディタを閉じる.「**>>>>**」の記号部分には手を加えない.「**>>>>**」で始まる行だけを抜き出して連結できれば, 草稿ができあがる. そのためのスクリプトは, **depara.sh**である.

エディタを閉じると, 開いたままでプロンプト記号「**$**」が示されるように復帰している端末で「**bash depara.sh how2use.txt**」を実行する. すると行の左端が「**>>>>**」で始まる英訳だけが抽出され, 段落ごとに連結された**draft_how2use.txt**が「**./drafts**」サブディレクトリの中に出力される. すぐに自動でエディタが開いて英訳草稿**draft_how2use.txt**が示されて, 再点検・修正できる(図２). 草稿が完成したところで, エディタを閉じる.

新たに試したいテキストファイル(仮に**newtext.txt**)があれば, エディタで書き, あるいは他所からコピーして,「**./sources**」サブディレクトリのなかに格納(セーブ)する. そのファイルに対して「**bash mistjaen.sh newtext.txt**」を実行し, 同様の操作を繰り返す.

原稿を仕上げるために, 他のPC上でワープロを使って草稿**draft_newtext.txt**を推敲する場合, 移動やコピー&ペーストには他のUSBメモリを使うか, メールに添付して自分宛に送ったり, DropboxやGoogle Driveなどクラウドのストーレッジに格納することで対処する. もちろん名前の付け替えやコピーはかまわない. ここまでで, mistletoe on MTシステムはとりあえず使えるようになっている. 次は本スクリプト集の仕掛けや細部のチューニング法の解説である. 詳細は興味に応じて解読するだけでよい.

ここで, mistletoeのディレクトリ構造の概略を説明しておく(図３). ディレクトリmistletoeの直下には, 1 個のPythonスクリプト(**future.py**)と８個のシェルスクリプト(**.sh**)がある. その他には, 5 個のサブディレクトリがある.

シェルスクリプトのうち, 翻訳作業を主導するのは**future.py**を操作する「**(1)mistjaen.sh**」であり,「**(2)preedit2.sh**」と「**(3)postedit.sh**」はその部品的な役割を担う.「**(4)depara.sh**」は編集の終わった和英対照形式から英文だけを抽出する.「**(5)dictrepl.sh**」は**preedit2.sh**の部品的な役割を担う.「**(6)precheck.sh**」は与えられた原文の訳出可能性を探ると同時に不足する用語置換表候補を作り,「**(7)fogjaen.sh**」はその作業の部品的な役割を担う.「**(8)sortanewtbl.sh**」は新たに作った用

```
|-future.py
|-mistjaen.sh …(1)
|-preedit2.sh …(2)
|-postedit.sh …(3)
|-depara.sh …(4)
|-dictrepl.sh …(5)
|-precheck.sh …(6)
|-fogjaen.sh …(7)
|-sortanewtbl.sh …(8)
|-./sources …(a)
|-./parallel…(b)
|-./drafts …(c)
|-./tbl…(d)
|-./work…(e)
```

図３.「mistletoe」の概略構造. ディレクトリ直下には，１個のPythonスクリプト(**future.py**)と８個のシェルスクリプト(**.sh**)，５個のサブディレクトリがある.

語置換表を**preedit2.sh**で参照できるように整形する.

サブディレクトリのうち,「**(a)sources**」には訳出用の原文(**.txt**)が格納される.「**(b)parallel**」には**mistjaen.sh**によって原文が訳出処理された結果の訳文(**.txt**)が和英対照形式で格納される.「**(c)drafts**」には編集の終わった和英対照文から**depara.sh**で抽出・連結された英文だけ(**.txt**)が格納される.「**(d)tbl**」には**preedit2.sh**を使って原文に用語置換をほどこす置換表(**.tbl**)（実体はテキストファイル）が格納される.「**(e)work**」には，**mistjaen.sh**が訳出途中に作る中間ファイル類が格納される.

V. みらい翻訳の問題点

この章から「**Ⅸ. カスタム置換表の追加法**」までは，mistletoe on MTを使いこなすためのシステムの仕組みや拡充に関する補足説明である. なお，無償お試しMTデモURLでは2000文字までの入力制限がつくが，普通の実務翻訳単位としては十分である.

MTの「利用規定」の試訳を含む，さまざまな和文英訳を試すと，次の６つの短所に気づく(同様の短所は2020年９月現在のDeepLにも見られる).

短所１:いちいち入力や出力をコピー＆ペーストしていたのでは面倒.

短所２:文の切れ目は「. 」(「。」も同等)で判断されるので,「この利用規約(以下「本規約」といいます.)は，・・・です. 」のように，１文の中に「. 」が複数回「.)」の形などで出現する場合，訳出に不具合が生じる.

短所３:文単位で翻訳され，前後の文脈・脈絡は活かされないので，各訳文での時制，単数形・複数形の区別，定冠詞・不定冠詞の選択が適切とは限らない. 原文と訳文の各文ごとの対照点検が不可欠である.

短所４:需要が高く，翻訳例の多い分野(法務，医学，金融，行政，特許，報道，情報処理など)の文章は，それなりに訳出されるが，一般の辞書に載っていそうにない生物名など，博物学のいわゆる専門用語には対応できない.

短所５:学習の元となったパラレル対訳データ(日本人学習者らによる不適切な英作文も含まれうる)に由来するのか，定形パターンの不具合が訳文に残る.

短所６:出力に，会話文などのインフォーマルな文体から，学術論文調のフォーマルな文体までが混在する. 日本語なら「です・ます」調と「だ・である」調の混在のようなもので，文体の統一に細工・点検が不可欠である.

とりわけ博物学の分野では，短所４は他人任せでは解決しない. 翻訳システムに連結できる専門電子辞書の編纂が，商業ベースには乗りにくいからである.「新・蝶類生物学英和辞典」(鍛冶2018)にも電子書籍版は今のところ存在しないが，あったとしても，そのままではAI翻訳には連結できないだろう. また，たとえ有償版のMTを使用したとしても，同じ困難は博物学的な学問分野では必ず残る. したがって，この部分には自らの専門知識と情報収集を元に，和英置換表の形でのインターフェースを整備して対処するのが，唯一の解決策である.

以上の欠点について，入力原文をテキストファイルとして事前に準備し，MTへの入力・出力の前後に，mistletoeのシェルスクリプトによる自動処理工程を組み込むことで対処する. このアプローチによって，作業効率を大幅に向上させることができる. なお，短所３の解決，すなわち文単位を超えて文脈まで解釈するAI翻訳は，遠くない将来に実現するかもしれない. しかし現状でのこの短所は，Ⅶ章で解説するように，逆に利用上の攻略点として使うことができる.

VI. スクリプト集misletoeによるMTの短所対策

前章の短所に自動で対策をほどこす**mistjaen.sh**のスクリプトの大筋を解説する.

(A)GUI画面でのコピー＆ペーストが不要となるように，保存してあるテキストを，コマンドラインでファイル名を指定するだけで読み込む仕様に細工する(短所１の対策). 以下に示す(E)で実行するコマンドを想定しつつ，まずは「**bash python3 future.py JA EN "**」で始まるテキストファイルとして，(B)～(D)の処理で変換・追記を繰り返しながら，(E)のシェルスクリプトの草稿を自動生成させる.

前処理として，次の(B)～(D)の工程を，置換用のスクリプトで工夫する. 主にコマンドツールstream editor「sed」を使う.

(B)パターン「.)」は，パターン「)」で全置換(短所２の対策). なお，句点(終止符)をあらわす全角「. 」と「。」も以下，同等に処理する.

(C)最終的な文単位での点検を容易にするために，文ごとに切り分ける．「．」を「．<改行>」で全置換（短所３の対策）（短所６の対策(G)との併用で各文ごとの時制，単数形・複数形，定冠詞・不定冠詞の適・不適などの点検に便利）．

(D)動物名（日本産全種，アジア，世界著名種）やそれらの科名，属名，植物科名，植物和名には，学名（ときには＋英名），そのほか特殊専門用語や学会名，個人名など固有名詞の翻訳置換表を使って全置換する（短所４の対策）（Ⅷ章に述べる部品スクリプト**preedit2.sh**および**dictrepl.sh**を使う）．

(B)～(D)の前処理が終われば，みらい翻訳（MT）に送って出力を得る．

(E)(A)のテキストから(B)～(D)の処理を通して得られた前処理済の中間テキスト出力を，**future.py**に組み合わせて生成されるテキストファイル「**bash python3 future.py JA EN "**<前処理済テキスト本体>**"**」を実体とするコマンド「**mistjaen.sh** <前処理済テキスト名>」をbashで実行し，出力をファイルに書き出す．

MTからの出力を回収するところまでを(E)で実現しているので，後処理として次の３工程を工夫する．(F)と(G)には，主にsedを使う（スクリプト**postedit.sh**に統合）．

(F)定形パターンの出力の不具合は，置換表を使って全置換する（短所５の対策）（たとえば日本人がよく誤用する「Especially」→「In particular」；ピーターセン（1988: 152-155）を参照）．

(G)学術文書としての英訳の出力を想定した言い回しに変換する置換表（たとえば「won't」→「will not」）を使って全置換する（短所６の対策）．

(H)入力原文と(G)を経由した出力の文ごとの対照点検が不可欠なので，それらが交互に並ぶように，新たなファイルに書き出す（「**paste -d "\n"** <入力原文> <出力訳文>」を使う）．

このように(A)～(H)の一連処理が，入力原文テキスト名を指定するだけのコマンド１発「**bash mistjaen.sh** <入力原文テキスト名>」で自動化される（フィルター的なスクリプトをパイプ「**|**」でつないだり，中間ファイルを作ったりしながら変換を重ねて，(H)の出力を得るシェルスクリプトに化けるテキストファイルを作る）．

(I)出力を点検・修正し終えたら，コマンド**bash depara.sh**で処理する．すなわち「**>>>>**」で始まっている英文行だけを抜き出して，「**>>>>**」を「」（空）に，「．<改行>」を「．」に全置換することで，「．」が直前に存在する改行が消えて全英文が段落に分けられて段落ごとに連結される（抜き出しにはgrep，置換には主にsedを使って，１つのシェルスクリプトにまとめてある）．

Ⅶ. 原文シャッフルによる難読化（簡易"暗号"化）と復号

MTの「利用規約」第３条（許諾条件）の中には，「当社及び当社が指定する業務委託先のサーバーに保存される」，「個人を特定できる情報（氏名，住所等）等が含まれる場合も・・・当該情報もテキストデータ等の一部として保存される」等々，注意しなければならない条項がある．保護すべき個人情報に該当する文字列は，伏字にするなどの対処が望ましい．これは自動化できないので，エディタを使って手作業で行う．

さらに念のため，全体としては脈絡のつかめない意味不明の文章が入力となるように，上記(C)の直後に細工(C')の工程を追加挿入する．これには「現状のMTが１文ずつしか翻訳しない仕掛けである」ことを逆手にとる．すなわち，(C)で１文ずつに切り分けた各文に文番号を振る．さらに番号付きの状態でシャッフル，言い換えればランダムに並べ替える．そして文番号部分を切り離して隠す．文番号の並びはMTに送る前に手元に保存しておく．

順序を復元する情報を隠した文の羅列は，そのまま読んだのでは脈絡もとれず，支離滅裂，全体としては意味不明となる．この脈絡のとれない文の羅列を「みらい翻訳」に送り，１文ずつの訳文を回収する．さらに(E)で回収した訳文に文の順番を復元するための(E')の工程を(E)の直後に追加する．それには(C')で保存してあった文番号情報を行左端に付加して復活させ，番号順に文をソート整列させることで，意図した意味を汲み取れる訳文の並びが得られる．

n個の文からなる文章であれば，文ごとに切り離してシャッフルすると，n!（nの階乗）通りの文の羅列の組み合わせが可能となる．600字の文章を想定してみよう．１文の平均が40文字長であるとすれば，この文章を構成する文の数は15個となる．シャッフルすれば，15!＝1,307,674,368,000通りの並べ替えが可能で，事前情報なしには脈絡のとれる並び順への復元は困難だろう．

MTデモURLでは，入力に2000文字までが許される．原文が相当に平易であっても，さらに支離滅裂な文章を入力として送ることができる．たとえば，制限2000文字に達するまで，ダミーの文章を末尾に足して，上記の処理をすればよい．「サルがタイプライターキーをランダムに叩き続ければ，Shakespeareの作品を打ち出すこともありうる」という「無限サルの定理 infinite monkey theorem」で「タイプライターキー」を「文」に置き換えた場合の逆応用である．

Ⅷ. 専門用語置換表の構造

専門用語等の置換表は，サブディレクトリ「**./tbl**」に置かれている．その構造は「**./tbl**」中のたとえば，

```
#A
bash dictrepl.sh ./tbl/custom01.tbl ./work/preedited00.txt > ./work/preedited01.txt
bash dictrepl.sh ./tbl/custom02.tbl ./work/preedited01.txt > ./work/preedited02.txt
・・・<中略>・・・
bash dictrepl.sh ./tbl/custom10.tbl ./work/preedited09.txt > ./work/preedited00.txt
#B
bash dictrepl.sh ./tbl/customX.tbl ./work/preedited00.txt > ./work/preedited0x.txt
bash dictrepl.sh ./tbl/<中略>.tbl ./work/preedited0x.txt > ./work/preedited0y.txt
・・・<中略>・・・ >./work/preedited00.txt
```

図４. カスタム置換表の追法例. 10個のカスタム置換表(**custum01.tbl**, **custum02.tbl**, …, **custom10.tbl**)をセットにして追加する場合, **preedit2.sh**の適当な部分に前半10行(**#A**)を追加挿入し, 単独に**customX.tbl**を追加するには置換系列の適切なところに, **0x**と**0y**が連番となって, さらにそれらの前後の行とスクリプトの流れがうまくつながるように, 後半(**#B**)を挿入.

dinosaurs.tblを見れば分かる. 日本語による論文等に見つかる恐竜の名称や用語が, 和文単語の文字列長の順に並べられ, 対応させて置換する<学名>や<英訳>が, 「**カツヤマリュウ <Fukuiraptor_kitadaniensis>**」のように和文単語との間にスペースを挟んで配置されている.

各置換表がディレクトリ**mistletoe**直下の部品スクリプト**preedit2.sh**およびその下位スクリプト**dictrepl.sh**によって参照され, 原文文字列中に当該文字列**カツヤマリュウ**が出現すると**<Fukuiraptor_kitadaniensis>**に置換され, 置換の完了した文書がMTに送られる. 文字列長の順に置換されるように並べてあるのは, たとえば**キチョウ**が少し長い**モンキチョウ**よりも先に置換されると, 不具合が起こるからである.

学名・英訳側の文字列中では, スペースはアンダーバー_に置換されて連続した文字列を構成し, 文字列は**<>**に囲まれている. この**<>**は, MTによる不都合な置換の点検を編集時に行うために付けてある. 原因不明のMTの不具合(おそらくはMTのもっている単語や連語出現統計データなどからのAI演算置換による)によって, たとえば近縁種**<allied species>**が**<allowed species>**に化けたり, アサギマダラの種小名**<sita>**が**<site>**に化けたり, **<insecticide>**が**<inside>**に, **<predation>**が**<preparation>**に化けるなどする. そのような点検箇所の目視検出に使う. 「<」や「>」の不要なものは, **depara.sh**での処理後, 草稿の段階に入ってから, 一括置換によって消す.

今回のmistletoeに使った置換表の多くは, 1997年頃から2019年秋まで20年以上にわたって整備拡充しながら, 数々の和文英訳(五十嵐・福田 2000, 松香 2001, 五十嵐・原田 2015, 長谷川 2020)に使った私家版スクリプト集Trex(高崎 2017)の置換表の一部に, 必要な修正を加えて流用した. また高崎(2018, 2019)で例示した原始的なテキストマイニングによって, Web上に公開されている和英用語集や和名学名対照表などから置換表を整備したものもある. おもだった出所については, 極力, 置換表の冒頭に#を付けるなどして, 各置換表(「**.tbl**」ファイル内)に明示した.

置換表では, 上記の「文字列長順への並べ替え」などのほか, 不具合回避の細工をほどこし, 気づいた学名の間違い等は修正した. さらに原典の利用規定に求められるなど, 文献表での明示が適切と判断されるのは, 次の通りである:石垣(1988), 本村(2020), 日本モンキーセンター霊長類和名編纂ワーキンググループ(2018), 日本爬虫両生類学会標準和名委員会(2020), 米倉・梶田(2003-). すでにmistletoeに入れてある置換表を使うことで, 分野ごとに多少の粗密はあるものの広範な種類の動植物の和名が, 学名あるいは英名に置換されて英訳文中に埋め込まれて出力される.

なお哺乳類の学名については, 「世界哺乳類標準和名目録」(川田ら 2018)の電子データ版(http://www.mammalogy.jp/list/index.html)から訂正・再構成した置換表も個人的に使っているが, 改変・再配布を禁じられているので, 今回のスクリプト集に含めていない. 該当する変換表を入れてあったサブディレクトリ「**./tbl/mammals**」は空にし, **preedit2.sh**の同サブディレクトリ内の置換表を参照する部分は#を付けてコメントアウトしてある.

IX. カスタム置換表の追加法

分野の必要に応じて, 専門用語置換表はテキストマイニングによって作成し, 追加すればよい. 適切な名称を使って「名前を付けて保存」する. 保存場所は, サブディレクトリ「**./tbl**」の中である. 次に, 部品スクリプト**preedit2.sh**をエディタで開いて編集する. 適当な分類群ごとに10行の組になるように#付きの空行でおおまかに区分してある.

この**preedit2.sh**スクリプトには, さらに下位の部品スクリプト**dictrepl.sh**が使われている. 置換表(たとえば**./tbl/custom.tbl**)を参照して, 入力の中間置換ファイル**./work/preedited0x.txt**を置換して, 出力の中間置換ファイル**./work/preedited0y.txt**に格納する. 辞書引き(dict<<dictionary)を繰り返しながら何段にも置換(repl<<replace)を重ねる**dictrepl.sh**の働きによって, **preedit2.sh**は数々の専門用語の置換を実現する.

仮に10個のカスタム置換表(**custum01.tbl**, **custum02.tbl**, …, **custom10.tbl**)をセットにして追加する場合, **preedit2.sh**の適当な部分に図４の前半10行(**#A**)を追加挿入する. さらに単独に**customX.tbl**を追加するには置換系列の適切なところに, 「**0x**」と「**0y**」が連番となって, さらに前後の行とスクリプトの流れがうまくつながるように, 図４の後半(**#B**)を挿入して保存する.

今回のmisteltoeに入れてある置換表の中には，各自の英訳作業には不要な分野の置換表も含まれていることだろう．不要な変換表を含む**preedit2.sh**の行を一括でコメントアウト（行頭に#を付ける）あるいは削除して，スクリプトの流れがうまくつながるように調整編集すれば，スクリプトの実行時間の多少の節約にはなる．あるいは，10行ごとの組を単位にして，置換の優先順位を入れ替えるのもよい．

また無関係の分野の置換表の文字列が原因となって，変換時に不具合を生じる場合など，置換表（**.tbl**）の原因文字列の行頭にも#を付けて，誤変換されないようにする．たとえば「ジャコウ」という文字列は，哺乳類名に複数回（ジャコウウシ，ジャコウジカ，ジャコウネコ，ジャコウネズミ，・・・）現れるだけでなく，昆虫名（ジャコウアゲハ，・・・）にも出てくるので，行頭に#を付ける．このような誤変換を未然に回避するために，不具合を生みやすく頻出しそうな文字列には，最初から#を付けておく．

英訳に先立つ文献調査やウェブ検索等で集めた細切れの情報を継ぎ合わせて作成した**dinosaurs.tbl**のような置換表もあり，そのようなカスタム置換表については，出所の明示は不可能である．そのような置換表は使用者各自が拡充することが想定される．そのための小道具スクリプトが，**precheck.sh**である．

この**precheck.sh**を使えば，入力テキストの翻訳に既成のmistletoe on MTが未加工のままで使えるか簡易判定できる．文書**sampletextX.txt**があるとしよう．これが容易に英訳可能かは，次のコマンドを実行することで判断できる．

$ bash precheck.sh sampletextX.txt

２文字長以上の漢字やカタカナ，漢字カタカナ合成語が抽出されて，置換表には登録されていない英訳候補（MTによる）が対に並んで「**./tbl/newtbl2check.tbl**」として出力される．そのままエディタが開いて，画面上に**newtbl2check.tbl**が編集可能な状態で示される（図５）．

エディタ上で**newtbl2check.tbl**から問題のない原語・英訳対を削除して，不具合のある原語・英訳対については，必要に応じて新たな置換対を整備し，適切に（仮に**./tbl/customX.tbl**）「名前を付けて保存」する．あとは，上記のように**customX.tbl**を参照して置換する**dictrepl.sh**を使う行が加わるように，**preedit2.sh**を編集して保存する．

なおMTの入力制限2000文字を超える文書であっても，２文字長以上の漢字やカタカナ，漢字カタカナ合成語を抽出した２次テキストの長さが制限文字数内であれば，**newtbl2check.tbl**は出力される．またmistletoe on MTで処理するには長すぎる場合，**newtbl2check.tbl**の末尾に点検した元テキストの長さや分割の目安となる文字数，行数の情報も出力される（図５）．

図５．「mistletoe on MT」による英訳可能性の簡易判定をかねる英訳候補用語出力例．コマンド**precheck.sh sampletextX.txt**で出力される置換表にはない編集可能な英訳候補リスト「**./tbl/newtbl2check.tbl**」．末尾に点検した元テキストの長さや分割の目安となる文字数，行数の情報も出力される．

新しく作成した置換表や複数の置換表をつないで作り直した置換表**customX.tbl**の中身を「文字列長順に並べ替える」小道具スクリプトも準備してある．次のコマンド**sortanewtbl.sh**である．

$ bash sortanewtbl.sh customX.tbl

博物学関連の学名の追加や点検に使える電子テキスト目録で有用性が高いものは，たとえば次のサイトにある：「BISMaL: Biological Information System for Marine Life（https://www.godac.jamstec.go.jp/bismal/j/about.html；2020年７月20日参照）」，「日本産蛾類総目録（http://listmj.mothprog.com/；2020年７月10日参照）」，「日本列島の甲虫全種目録（2020年）（https://japanesebeetles.jimdofree.com/；2020年７月10日参照）」，「日本産昆虫名称辞書（DJI）（https://konchudb.agr.agr.kyushu-u.ac.jp/dji/index-j.html；同日参照）），「日本産生物の種名チェックリスト一覧（http://www.ujssb.org/checklist/index.html；2020年８月13日参照）」，「琉球産蛾類目録2020（https://gashowkimura.wixsite.com/website/moths-of-ryukyu；2020年８月11日参照）」，「雑草名リスト（http://wssj.jp/academic/；2020年８月10日参照）」．したがって，カスタム変換表の拡充は，かける手間次第

でいくらでも可能となる.

今後も電子テキスト化された情報が加速的に得やすくなるので, 分野の必要に応じて拡充すればよい. しかし, 用語変換処理にはコンピュータの性能向上を期待するとしても, 拡充すればするほど処理速度は遅くなる. 使用者が高頻度で翻訳する分野に合わせた拡充に止める.「 生物名の学名と和名の対応 (http://togodb.dbcls.jp/species_names_latin_vs_japanese;2020年８月10日参照)」から, 最新の学名情報であるかは別にして, mistletoeの公開にあたって, なるべく広範な禽獣草木虫魚をカバーするように作成した置換表を補充して入れてある. 利用者の分野に不要の置換表は, スクリプト**preedit2.sh**のなかで該当する行をコメントアウトして, スクリプトの流れを修正することで, 処理速度が多少は向上する.

X. おわりに

本即席細工(bricolage)は, 学術情報発信のグローバル化に貢献することを目指して書いた. ここにあげた例は, 動物学や植物学など「博物学」の分野に限定したが, MTという「巨人の肩にのる」細工の骨組みは, 分野ごとに最適化する需要が小さすぎて, 自分たちで何とか対処するしかない他分野にも使える. 細部の仕掛けは, 原文言語と翻訳言語における専門用語の対照置換表を, 本稿での例を参考に自力で整備すればよい. 研究者の絶滅危惧分野や真に萌芽的な最先端分野など, 研究者や同好者の数が少なく, 加えて研究費が乏しい分野であっても, 成果発信のグローバル化に突破口が開けるだろう. なお, ここに準備した変換表を参照する必要のない分野の文書については, そのまま問題なく使用することができる.

いずれMTのデモURLが閉鎖されるときも到来するかもしれない. そのときは, さらに別の「巨人」無料AI翻訳サイトの「肩にのる」工夫をmistletoeの骨組みを下敷きに細工すればよい. DeepLについては, 本稿執筆中にもコマンドラインから操作するためのPythonスクリプトが, 複数種類, 出始めている(たとえば**deepl-cli.py**;https://github.com/eggplants/deepl-cli;2020年８月20日参照).

その場合でも, 専門用語の事前置換はせざるをえない. かつて膨大な私家版の専門用語置換表を蓄えたTrex(高崎2017)は20年以上も命脈を保ちながら巨大化した. その一部の置換表と置換用スクリプトは, mistletoeの継承資産となって今も生き続けている. 本誌*Naturalistae*や日本蝶類学会誌Butterflies, 鹿児島昆虫同好会誌Satsumaの校正作業を省力化するために使われているスクリプト集**foolproof2_scripts**(高崎2019)のなかに, 修正加工することで**precheck.sh**や**sortanewtbl.sh**に使える部品があった. また文脈・脈絡まで解釈する次世代AI翻訳が出現しても, 文単位に切り分けての**原文シャッフル処理による難読化(簡易"暗号"化)と復号**は使い続けられる.

振り返れば, 2019年に依頼を受けていた長谷川(2020)による「日本のゼフィルス」の英文テキストの翻訳作業の最終段階になって, 無償公開された「みらい翻訳」デモURLの存在を知り, 関連情報を収集するうちにコマンドライン操作を可能にするy-ichi氏(2019)による**A_mirai.py**があることに気づいた. シェルスクリプト集mistletoeの着想は, 2019年10月のことだった. 同図説の編著者の快諾のもと, 最後に届いた６種分, 400字詰め原稿用紙30枚相当の原稿で試して, 翻訳効率に手応えをえた. 続いて2019年末, 日本蝶類学会大会の要旨集の英訳では, 2018年までの同学会要旨集とは比較にならないスピードで訳出を完了した(高崎2020). さらに2020年１月下旬に集められた岡山理科大学理学部動物学科の2020年３月卒業生の卒業研究要旨集では, 大半の希望者の研究題目の粗訳, 加えて希望者９名には要旨本体の粗訳を作成した.

いまや伝統建築の復元でさえ, 電動工具を利用することで宮大工も効率化を図る. 現代の翻訳実務も, 同じ感覚でAI翻訳を使うことが可能になっている. 木工職人は作業の効率を上げるべく, 電動工具を便利に使いこなすために, 創意工夫を重ねながら, さまざまなジグを自作する. 翻訳職人の端くれでもある執筆者にとっては, mistletoeはAI翻訳MTを使いこなすジグに相当する.

木工職人は効率化で余裕の生まれた時間を, さらなる工夫のアイデアを温めたり, 手を抜くことのできない手鉋での仕上げ作業に当てる. 翻訳にも類似のアプローチをとることができる. 翻訳の仕上げ作業では, さまざまな関連情報を参照する. とりわけ非母国語で英作文する場合には, 個々の用語の用例を専門分野ごとに参照しなければならない. 用例を集めたコーパスの集積と検索を容易に実現する仕掛けとして, AntConc(Anthony 2019)がある. その活用に関する短い解説(高崎 2012;Takasaki 2013)も本稿と同時に参照いただければ幸いである.

謝辞

無償デモURL「MT」を提供する株式会社みらい翻訳やDebian RPD, さらにMTにコマンドラインからアクセスできる**A_mirai.py**を作成・公開したy-ichi氏の業績なしにはmistletoe on MTは実現しませんでした. 本稿の準備には, 推敲やさまざまな段階でのテスト等で多くの方々の協力をいただきました. 西村直樹氏には, 長年にわたって同氏が集積した日本産コケ類の和名・学名対照の電子テキストを提供いただきました. 次の方々(敬称略, 順不同)には大いに助けていただきました:秋野順次, 橋本十夢, 稲岡茂, 石堂陽一, 石垣忍, 北川文夫, 小平将

大，栗山定，大原賢二，斎藤基樹，鈴木浩太，竹内剛，名取真人，三高雄希，守田益宗，薮田慎司．また長谷川大や福田晴夫，増井暁夫の３氏，本誌*Naturalistae*および日本蝶類学会誌Butterfliesの編集事務局には，実務での英訳作業に本稿で紹介したシステムを試用する機会をいただきました．以上，記して感謝します．

引用文献

Anthony, L. (2019) AntConc (Version 3.5.8) [Computer Software]. Tokyo, Japan: Waseda University. https://www.laurenceanthony.net/software（2020年２月16日参照）．

長谷川大（編著）（2020）日本のゼフィルス．むし社，東京．

川田伸一郎，岩佐真宏，福井大，新宅勇太，天野雅男，下稲葉さやか，樽創，姉崎智子，横畑泰志（2018）世界哺乳類標準和名目録．哺乳類科学 58 Supplement: S1-S53．

五十嵐邁・福田晴夫（2000）アジア産蝶類生活史図鑑Ⅱ．東海大学出版会，平塚．

五十嵐邁・原田基弘（2015）続アジア産蝶類生活史図鑑．自費出版．

石垣忍（1988）足跡学の用語．生物科学 40(1): 31-38．

鍛冶勝三（2018）新・蝶類生物学英和辞典，岩野秀俊（監）．北隆館，東京．

松香宏隆（2001）トリバネチョウ生態図鑑．松香出版，東京．

Mitchell, R.［著］・黒川利明［訳］・嶋田健志［監修］（2016）PythonによるWebスクレイピング．オライリー・ジャパン，東京．

本村浩之（2020）日本産魚類全種目録．これまでに記録された日本産魚類全種の現在の標準和名と学名．鹿児島大学総合研究博物館，鹿児島．Online ver. 3．https://www.museum.kagoshima-u.ac.jp/staff/motomura/jaf.html（2020年８月10日閲覧）．

中島雅弘（2018）UNIX独習への近道．技術評論社，東京．

中島雅弘・富永浩之・國信真吾・花川直己（2015）AWK実践入門．技術評論社，東京．

奈佐原顕郎（2016）入門者のLinux―素朴な疑問を解消しながら学ぶ．講談社，東京．

日本爬虫両棲類学会標準和名委員会（2020）日本産爬虫両生類標準和名リスト．http://herpetology.jp/wamei/index_j.php（2020年８月10日閲覧）．

日本モンキーセンター霊長類和名編纂ワーキンググループ（2018）日本モンキーセンター 霊長類和名リスト 2018年11月版．http://www.j-monkey.jp/publication/specieslist/JMC-PrimateSpeciesList-Nov2018.pdf（2020年８月５日閲覧）．

マーク・ピーターセン（1988）日本人の英語．岩波書店，東京．

高崎浩幸（2012）一日で作る私家版「蝶類学英語活用電子辞典」．Butterflies (*Teinopalpus*) 61: 48-51.

Takasaki, H. (2013) An instant electronic dictionary of English collocations in natural history realized using a concordancer and open resources. *Naturalistae* 17: 59-62.

高崎浩幸（2017）日本蝶類学会大会2016要旨英訳：舞台裏のからくり．Butterflies Newsletter 71: 5-8.

高崎浩幸（2018）原稿中の専門用語・学名等アルファベット文字列のシェルスクリプトによる簡易つづり校正法．*Naturalistae* 22: 9-29.

高崎浩幸（2019）原稿中の漢字カタカナ文字列のシェルスクリプトによる簡易つづり校正法．*Naturalistae* 23: 75-86.

高崎浩幸（2020）コマンド１発AI翻訳：日本蝶類「楽」会的な利用法．Butterflies Newsletter 80: 14-17.

上田隆一（2015）シェルプログラミング実用テクニック．技術評論社，東京．

米倉浩司・梶田忠（2003-）BG Plants 和名－学名インデックス（YList）．http://ylist.info（2020年３月10日閲覧）．

要約

Web上に無償公開されているAI翻訳のデモURLに対して，Linuxのコマンドラインからアクセスして翻訳出力を自動回収するシェルスクリプトを公開する．本稿は，博物学分野を例に学術情報のグローバル化に不可欠の和文英訳で，専門用語の置換の仕組みなどを解説する．事前に準備したテキストファイルに専門用語の事前置換をほどこしたものを一文ずつ外部システムにAI翻訳させ，出力を和英でのパラレル対照訳形式に加工して返す．文単位に切り分けたあと，入力前の原文シャッフル処理による難読化（簡易“暗号”化）と復号の工程も組み込んである．回収した翻訳出力に現れる定形パターンの不具合を自動修正する．これらの作業がシェルスクリプトにまとめられ，コマンド一発で操作できる．パラレル対照の出力形式で一文ずつの点検が容易で，点検と修正が済めば，また別の一発コマンドで，出力訳文だけが抽出され，段落形式に連結される．すなわち，和文英訳作業がこれまでよりもはるかに楽になる．ここに使われているシェルスクリプトの骨子は，需要が見込めず，専門的なAI翻訳が商業的に提供されない他の分野への移植などに応用可能である．なお，この仕組みをUSBフラッシュメモリ起動の軽量Linuxで実現することで，WindowsであるかMacか，Linuxかを問わず，PCをUSBメモリから起動するだけで，自分の好みにチューニングできるAI翻訳機にPCが化ける．スイッチを切り，起動USBメモリを抜いて再起動すれば，PCは元に戻る．その作り方も解説する．

（2021年１月13日受理）