

GitLab CIを活用した 英日翻訳改善

HIGUCHI Daisuke
CREATIONLINE, INC.

2019/05/09



この発表について

■ 目的

- GitLab CIを活用してLaTeX英和翻訳を改善した概要の紹介

■ ゴール

- みんなが「我がチームでも採用しよう」と動き始める状態



GitLabのWeb IDEを利用したLaTeX英和 翻訳

- 原文をGitLabに取り込む
- 章ごとにブランチを切る
- ブランチごとにWeb IDEで翻訳・レビュー・修正
- 最後にブランチをマージして完成



章ごとにブランチを切る

4

5

27

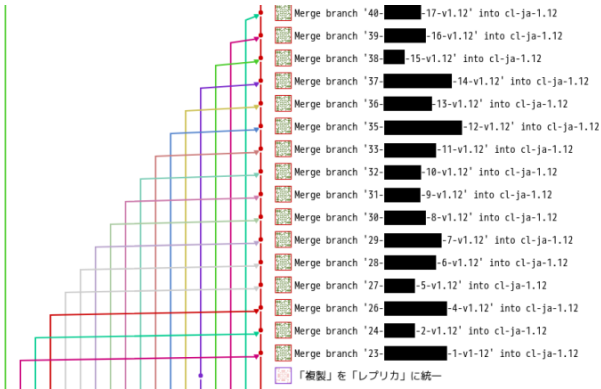
26

25

21

- 途中
- の更新箇所の翻訳完了
- の1行のみ修正
- の「してください」を「します」に
- の「してください」を「します」に
- の「してください」を「します」に
- の更新箇所の翻訳完了
- fix a couple of changes in \file{} macros with \$ and _
- まだ途中です。
- 4章 更新部分の翻訳完了
- 続きは明日！
- gitlab-ci: add docker:dind and get rid of custom docker image
- Update .gitlab-ci.yml
- 1章 更新部分の翻訳完了
- 差分比較における線94行目まで終わり。次（来週）138行目から作業開始する。--20181214
- 2つの変更のうち、1つ翻訳しました。続きは後日。--20181211
- Merge V1.12J(1.12英) into ci-ja-1.12(1.9日)

最後にブランチをマージして完成



GitLabのWeb IDEのメリット

- Gitコマンドラインを知らなくても翻訳作業ができる
- GitコマンドラインorWebUIで履歴をたどれる
- イシューボードと連携してレビューができる
- ブランチを切って手分けして翻訳・レビュー・マージができる



GitLabのWeb IDEのデメリット

■ 既存の和文との比較が大変

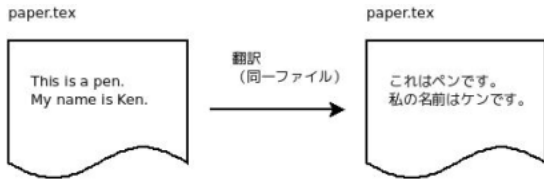
- Web IDEは翻訳ツールではないので、同じ文や似た文を参考として表示する機能がない

■ 原文の更新の反映が大変

- 同一ファイルを英文から和文に書き換えてしまっているため、単純なマージでは衝突してしまう



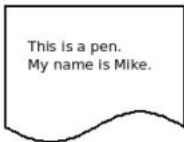
原文の更新の反映が大変(1/4)



同一ファイルを英文から和文に書き換え

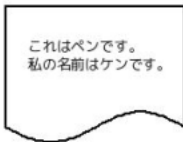
原文の更新の反映が大変(2/4)

paper.tex



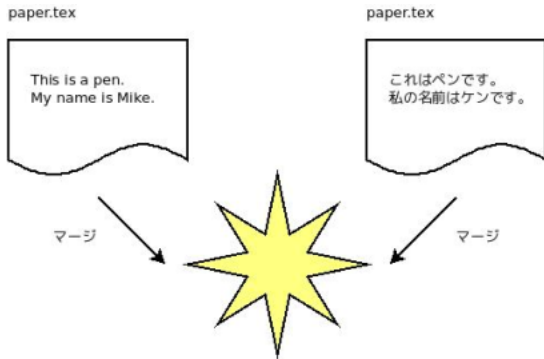
原文アップデート(Ken→Mike)

paper.tex



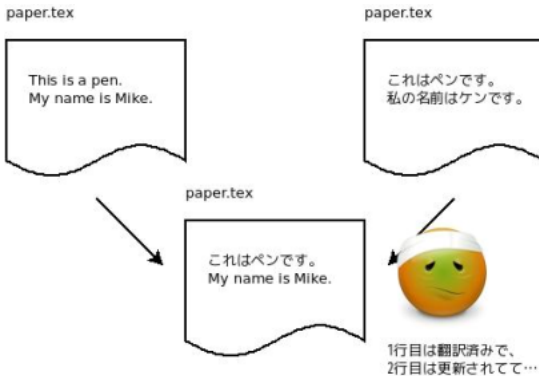
原文に更新があった(Ken→Mike)

原文の更新の反映が大変(3/4)



マージしようとするするとコンフリクト！

原文の更新の反映が大変(4/4)



手動でコンフリクト解消…

翻訳支援

ツールを

使いたい

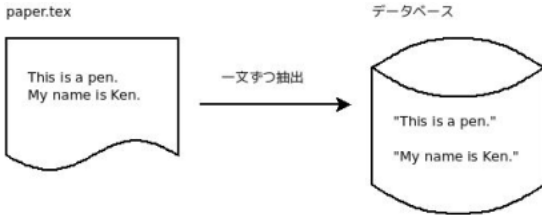
CL標準翻訳ツールOmegaT

コンピュータ翻訳支援 (Computer Assisted Translation; CAT) ツール

- 翻訳する際は、原文ファイルを直接書き換ええない
- 原文ファイルから「一文ずつ」抽出してデータベースに格納する
- 翻訳する際は、そのデータベースを翻訳する
- 原文ファイルとデータベースを突き合わせて、新たに訳文ファイルを生成する



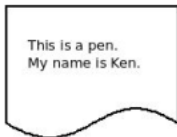
OmegaTの動き(1/3)



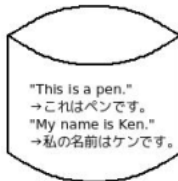
原文ファイルから「一文ずつ」抽出してデータベースに格納する

OmegaTの動き(2/3)

paper.tex

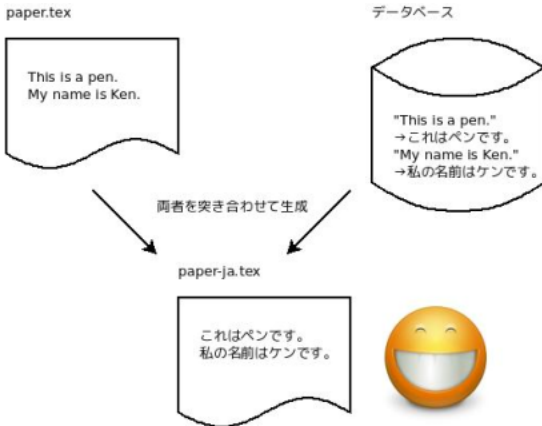


データベース



翻訳する際は、そのデータベースを翻訳する

OmegaTの動き(3/3)



原文ファイルとデータベースを突き合わせて、新たに訳文ファイルを生成する

OmegaTのメリット

- 原文ファイルを直接操作しないので更新に強い
- 翻訳単位が「一文ずつ」なので、ファイルが違ってても原文が同じなら自動的に訳文を当てはめてくれる
- 似た文を参考訳として表示してくれる
- カスタマイズ可能な用語集を持ち、単語レベルでも参考訳を表示できる
- データベースはGitなので、GitLabで管理できる



OmegaTのデメリット

- Gitは基本的な機能しか対応していない
 - master以外のブランチは扱えない(v3現在)
 - 手分けして翻訳する際は、声をかけあわないと作業が衝突してパーになる
- 訳文ファイルを生成する際、原文ファイルのレイアウト通りにならないことが多い
 - PDF、Word、Powerpoint、**LaTeX**などはレイアウトが崩れる場合がある



OmegaTでのLaTeX英和翻訳

OmegaTでのデメリット：

訳文ファイルを生成する際、原文ファイルのレイアウト通りにならないことが多い

に抵触して失敗

- LaTeXはレイアウトが崩れてしまった
 - 標準的なLaTeXのみ対応で、設定で拡張できない
- OmegaT以外にうまい方法はないものか？



ある

po4a: Portable Object for Anything

- さまざまな原文から翻訳用のデータベースを作成し、原文とデータベースから訳文を生成するツール
 - このデータベースを「POファイル」と呼ぶ
 - 「for Anything」の通り、**LaTeX**にも対応し、設定で拡張できる
- 翻訳支援機能そのものは持っていないが、POファイルは共通フォーマットなのでさまざまな翻訳ツールがある
 - 雑に言えばOmegaTから翻訳支援機能を抜いた感じ

POファイルの例

```
if (ALL_ELEMENT_SUB (w[1]) && len == 2)
{
    last_command_exit_value = 1;
    if (assoc_p (var))
        report_error (_("%s: invalid associative array key"), w);
    else
        report_error (_("%s: cannot assign to non-numeric index"), w);
    continue;
}
```

GNU Bashのソースコード

```
#: arrayfunc.c:586
#, c-format
msgid "%s: invalid associative array key"
msgstr "%s: 無効な連想配列のキーです"

#: arrayfunc.c:588
#, c-format
msgid "%s: cannot assign to non-numeric index"
msgstr "%s: 配列の添字に非数字を設定できません"
```

GNU Bashの日本語POファイル



OmegaT meets po4a

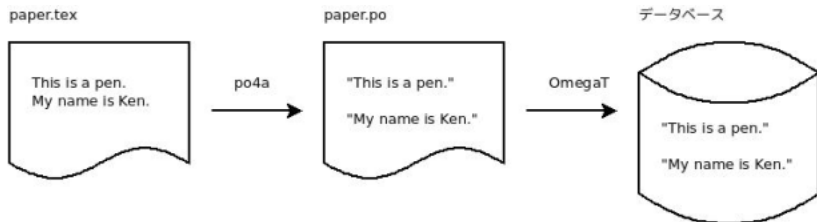
- OmegaTはLaTeXに対応していない
- po4aはLaTeXに対応している
- OmegaTはPOファイルに対応している

つまり

- po4aを使って原文LaTeXからPOファイルを生成し、OmegaTはPOファイルを原文として訳し、po4aを使って訳したPOファイルから訳文LaTeXを生成すればいい

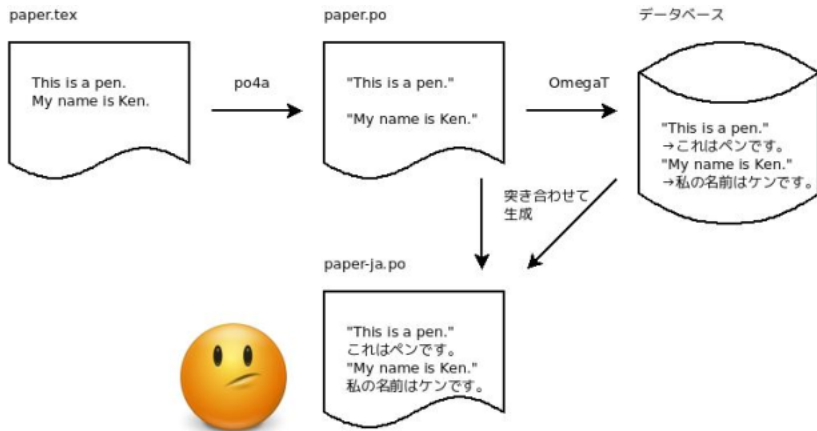


OmegaTとpo4aの組み合わせ (1/3)



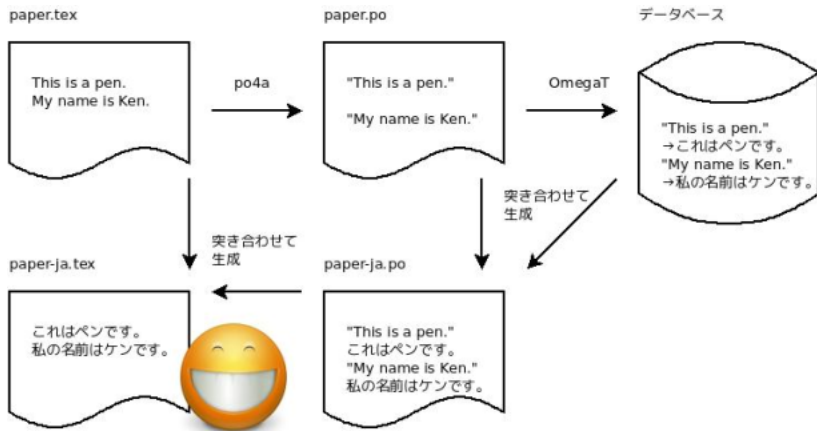
po4aで原文LaTeXから原文POファイルを生成し、OmegaTに取り込み

OmegaTとpo4aの組み合わせ (2/3)



OmegaTで翻訳し、原文POファイルと突き合わせて訳文POファイルを生成

OmegaTとpo4aの組み合わせ (3/3)



po4aで原文LaTeXと訳文POファイルを突き合わせて訳文LaTeXを生成

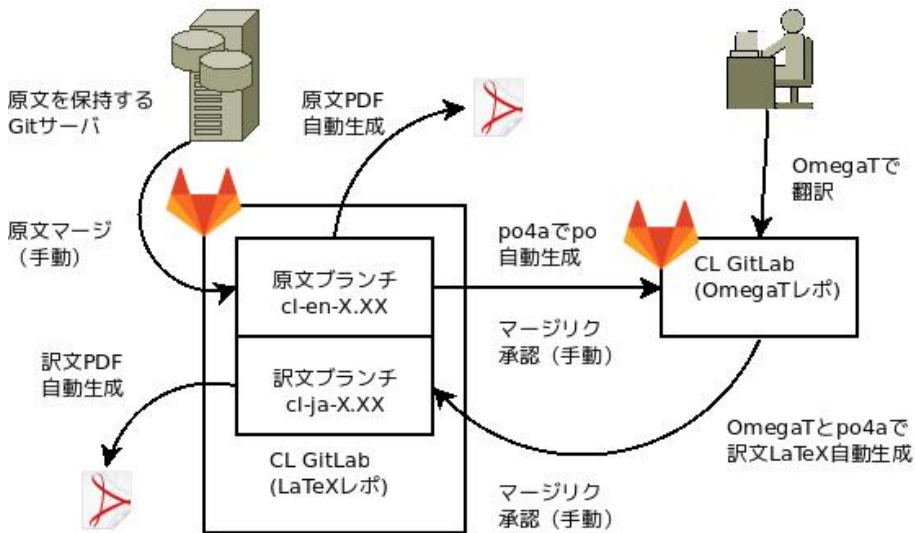
自動化すべき箇所

- po4aで原文LaTeXを原文POファイルに変換
- OmegaTで原文POファイルを翻訳
- OmegaTで訳文POファイルを生成
- po4aで訳文POファイルと原文LaTeXから、訳文LaTeXを生成
- LaTeXコンパイラで訳文PDFを生成

翻訳作業以外**すべて**自動化実現しました！



概略図



享受できたメリット

■ OmegaT

- Gitコマンドラインを知らなくても翻訳作業ができる
- 原文ファイルを直接操作しないので更新に強い
- 原文が同じなら自動的に訳文を当てはめてくれる
- 似た文を参考訳として表示してくれる
- 単語レベルでも参考訳を表示できる

■ GitLab

- GitコマンドラインorWebUIで履歴をたどれる
- イシューボードと連携してレビューができる



解消できたデメリット

■ OmegaT

- 既存の和文との比較も簡単にできる
- 原文に更新があっても、ファイルに触っていないので、そのままマージできる

■ po4a

- LaTeXファイルは触らないので、レイアウトを保持できる



運用でカバー

- ブランチは切れないが、それでも手分けして翻訳・レビューはできる



定量的な変化

■ v1.12 (Web IDE)

- 12/03 - 12/21 : マージ+コンフリクト修正
- 12/21 - 03/01 : 追加翻訳+レビュー+修正

■ v1.13 (OmegaT)

- 03/15 - 03/15 : マージ (**コンフリクトなし!**)
- 03/15 - 04/07 : 追加翻訳+レビュー+修正

■ マージ+コンフリクト修正 : 18日→**1日**

■ 追加翻訳+レビュー+修正 : 70日→**23日**



振り返り

■ 目的

- GitLab CIを活用して英和翻訳を改善した概要の紹介

■ ゴール

- みんなが「我がチームでも採用しよう」と動き始める状態

動き始めましたか？

