日本語→琉球諸語翻訳モデルの構築に向けて

當間愛晃[1] 狩俣繁久[2] 岡﨑威生[1]
[1]琉球大学 工学部工学科知能情報コース
[2]琉球大学 戦略的研究プロジェクトセンター 産学官連携研究員
tnal@ie.u-ryukyu.ac.jp karimata@ll.u-ryukyu.ac.jp
okazaki@ie.u-ryukyu.ac.jp

概要

消滅の危機に瀕している琉球諸語の記録，記述や復興への一助のため，日本語を琉球諸語へ機械翻訳するモデル構築に向けた 3 つの検証を行った．実験結果から,対象言語を含まない翻訳学習を一度行い，その上で日琉翻訳学習を行うことが翻訳品質の向上に寄与することを確認した．品質改善に向けては，本ドメインに特化した評価指標の構築，ルールベース翻訳した中間言語を介した翻訳，検証用・テスト用コーパスの構築，新規トークン埋め込みベクトルの検討が寄与すると考えられる．

1 はじめに

グローバル化が進む中，世界中で約 2,500 の言語（日本では 8 つの言語）が近いうちに消滅する危険性があると指摘されており，危機言語や危機方言に関する記録・記述・復興といった研究が取り組まれている [1].

危機言語の一つに琉球諸語がある．琉球列島の人口や面積は日本全体の約 1%に過ぎないが，広い海域に点在しているために琉球諸語内部の方言差は非常に大きく，シマ（琉球諸語の最小単位としての島や集落のこと）が違うと何を言っているのか理解できないことがあるほど多様である．琉球は日本国内にあってマイノリティだが，琉球内部にもマジョリティとマイノリティがあり，小さなシマはマイノリティの中のマイノリティとして二重，三重の圧力を受けている．復興に向けた取り組みは重要だが，子供の立場からすると何故方言教育が必要なのか，方言教育を通して何を学ばせ，どんな人間になってもらいたいのか等の議論は不十分 [2] である．

我々は,消滅の危機に瀕している琉球諸語の記録,記述や復興への一助のため，日本語を琉球諸語へ翻訳（以降日琉翻訳と呼ぶ）するモデルの構築を目指している．著者の一人である狩俣は琉球諸語における 28 の方言についての文法記述 [3] を初めとし，琉球諸語の記録・記述・復興に向けた取り組みを数多く行っている．これらの研究で蓄積されたデータを元に翻訳コーパスを構築し，日琉翻訳モデルの構築を行うことを目指す．コーパスは最終的には数十万文規模の対訳文（日琉対訳コーパス）で構成されることを想定しているが，元データの大部分は方言辞典であり，対訳文そのもの以外にも利用可能なデータ（例えば単語そのもの）が含まれている．これらをどこまで，どのように利用するかについては議論の余地がある．

本研究では小規模の対訳コーパスを用いた日琉対訳モデル構築を行い，今後の方向性を検討するための予備実験を行った.データは「動詞活用調査票　読谷村儀間」から約 200 件の対訳文を用意した．翻訳モデルは日本語を含む多言語事前学習済みモデルである google/mt5-base [4] をベースとし，以下の 3 つのパターンでファインチューニング（以降学習と呼ぶ）を行った．基本的にはそれぞれ処理を追加することで BLEU スコアの改善を確認することができたが，単語トークンの追加についてはまだ検討を要する結果となった．

- ケース 1：日琉対訳コーパスのみで学習.
- ケース 2：日本語を含む別の対訳コーパス（Asian Language Treebank; ALT [5]）で学習後，日琉対訳コーパスで学習.
- ケース 3：一部単語をトークン設定後，ケース 2 にて学習.

2 実験設計

2.1 日琉対訳コーパスの構築

狩俣から提供を受けた対訳コーパスの例を表 1 の元文に示す．基本的には方言辞典に掲載している例

文抜粋であり，多くは chat ドメインに近い．ただし対話文とはなっておらず，単語を用いた話し言葉の例となっている．大半は単文だが，例示したように複数文が含まれることもある．また，「弟がこそ食べよったんだ」という言い回しからはやや古い言い回しが用いられている印象を受ける．

表記上の特徴としては，日本語文・琉球諸語文のどちらにも句に準ずる箇所で区切り文字(スペース)が挿入されている．これにより句単位のマッチングについては対応関係を観察しやすいコーパスとなっているが，区切り文字の挿入誤り（表 1 では琉球諸語文の 2 文目冒頭に挿入がない)も見られた．一方，日本語文→琉球諸語文への翻訳への入力文として考えた場合，ユーザに適切なスペース挿入を要求することは難しいと考えられるため，日本語文においてはスペースを削除することとした．

表 1　日琉対訳文の例（スペースを__に変換して掲載）

	日本語文	琉球諸語文
元文	残った__テンプラは、全部__弟がこそ__食べよったんだ。__俺じゃないよ。	ヌクトーヌ__ティンプラーヤ、ムル__ウットゥヌル__カムタル。ワンネーアランドー。
処理後	残ったテンプラは、全部弟がこそ食べよったんだ。俺じゃないよ。	（同上．そのまま利用）

表 2　特殊表記の例

記号種別	例文	備考
/	太郎も__食べるだろう／食べるはず。好きだから。	日本語文、琉球諸語文の双方で異なる言い回しを掲載する際に用いている模様。
・	ワンガ__カムヌ__メーニ__ネーン・ネーラン__ナトー__タン。	琉球諸語で2通りの表現がある場合に用いている模様。（日本語文では 1 つの表記のみ）
()	そうか。来よったんだ。（私は見なかったけど）	状況説明のために用いている模様。（カッコ内の文に対応する訳文なし）

スペース以外にも表 2 のように記号を用いた表記があり，事前処理が必要な例文も存在することを確認している．これらについては仮処理として「/」を含む文を除外し，残りについてはそのまま用いることとした．これにより合計 211 件の日琉対訳コーパスを構築した．

2.2 実験手順詳細

google/mt5-base を用いた 3 種類の検証を行った．ケース 1〜3 に共通する前処理を以下に列挙する．

- 日英翻訳，英日翻訳，日琉翻訳それぞれに対応する合計 3 個の専用トークンを追加した．
- 日琉対訳コーパスをシード値により固定した上で train : val : test = 8 : 1: 1 で分割して利用した．
- 検証データに対する損失が 3 エポック連続で更新されない場合には EarlyStopping により停止させた．テストデータによる評価は停止時点でのモデルにより行った．
- 初期学習率を lr = 5e-4 とし，get_linear_schedule_with_warmup を用いて num_warmup_steps = int(total_steps * 0.01) により減衰させた．
- 最大トークン数を 50 とし，超える場合には打ち切り処理をした．（該当サンプルを除外することはしていない）

2.2.1 ケース 1：日琉対訳コーパスのみで学習

ケース 1 では，最も素朴に日琉翻訳をファインチューニングにより学習させた．

2.2.2 ケース 2：ALT で学習後，日琉対訳コーパスで学習

ケース 2 では，2 段階のファインチューニングを行った．

1 段階目の学習では，ALT の学習用データから約 1.8 万件の日英・英日対訳コーパスを用い，1 エポックのみ学習させた．学習時にはランダムに日英・英日いずれかを選ばせたため，大凡 9 千件ずつ学習を行ったことになる．この時点では日英翻訳，英日翻訳の専用トークンのみを用いており，日琉翻訳の専用トークンを用いていないが，日琉翻訳を包含する「翻訳タスク」を一度学習させることで 2 段階目の学習がより効率良く行われることを期待して実施した．

2 段階目の学習では，ケース 1 の日琉翻訳を学習させた．なおこの際の学習率については再び初期値（5e-4）から行った．

2.2.3 ケース 3：一部単語をトークン設定後，ケース 2 で学習

事前学習済みモデルで用いられているトークナイザはそこで用いられたコーパスで最適となるように設計されていることから，本来であれば専用のコーパスで設計し直すことが望ましい．一方で事前学習をするだけの大規模コーパスを用意することは難しい．折衷案として，今回は方言辞典が元となっていることを踏まえ，一部の単語を通常トークンとして新規追加することの影響を観察することとした．

表 3 に追加した琉球諸語単語を示す．すべての単語が方言辞典の項目として掲載されているとは限らないため，今回は一部の名詞を登録候補とした．このうち細字の 3 つの単語（ワン，ヤー，チュー）については既存トークナイザに登録されていたため登録除外し，太字の 6 つの単語（イャー，ワッター，ナチ，メーナチ，ゴーヤー，ヒティ）を追加した．これにより表 4 のように新規トークンを用いてトークナイズされることを確認した．

表 3　登録候補と追加した単語一覧（太字が実際に追加した単語）

日本語単語	琉球諸語単語
あなた	**イャー**
私	ワン
我が	**ワッター**
家	ヤー
夏	**ナチ**
毎日	**メーナチ**
ニガウリ	**ゴーヤー**
今日	チュー
朝	**ヒティ**

表 4　トークン追加により期待する効果

元の文章	イャーヤ　ナスビ　カムンナー？
トークン追加前	['**イ**', '**ャ**', '**ー**', 'ヤ', '_', 'ナス', 'ビ', '_', 'カム', 'ン', 'ナー', '?']
トークン追加後	['**イャー**', '_', 'ヤ', '_', 'ナス', 'ビ', '_', 'カム', 'ン', 'ナー', '?']

なお，新規追加した 6 つのトークンの埋め込みベクトルは初期状態であり，このままでは翻訳学習がスムーズに行われない可能性が高い．そこで今回は「該当する日本語単語トークン列の埋め込みベクトルから求めた平均ベクトル」を新規トークンの埋め込みベクトルとして設定した．

新規トークンの追加とそれらの埋め込みベクトルを設定し終えた後は，ケース 2 の 2 段階のファインチューニングを行った．

2.3 評価指標

機械翻訳系の論文でよく見かける指標である BLEU [6], ROUGE（ROUGE-1, ROUGE-2, ROUGE-L）, TER [7] により評価を行った．なお，ROUGE については「トークン n-gram の一致率」として利用するため，独自実装した[i]．

3 実験結果と考察

3.1 損失推移の観察

図 1 にケース 1 とケース 2（2 段階目）の学習中における損失推移を示す．ケース 1 では初期損失が 10 前後から減り始めるのに対し，ケース 2 の 2 段階目では 1 前後からのスタートとなっている．図には掲載していないが，ケース 3 もケース 2 と同様の傾向であった．これらのことから，対象言語を含まなくとも，原言語を含む日英・英日翻訳を行うことで対象言語翻訳の学習を効果的に進めることが可能であることが分かる．

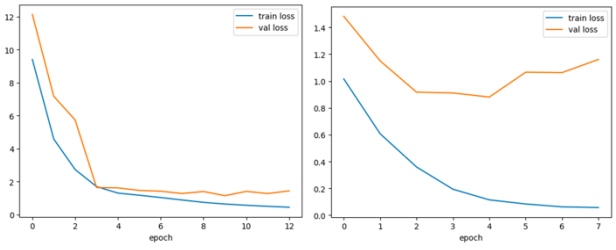

図 1　損失推移（左：ケース 1，右：ケース 2）

[i] いくつかのライブラリで検証した限りでは，内部でステミングやレマタイズ等の処理が行われるため，意図したトークンを用いていない（表記上も異なる）にも関わらずスコアが高く評価されることがあった．

3.2 評価指標の観察

表 5 に評価指標に基づくスコアを示す．2 段階ファインチューニングを行ったケース 2 とケース 3 は BLEU スコアを 2 桁改善，ROUGE スコアを 3 倍強に改善, TER を 25 ポイント改善することができた．殆どの指標においてケース 3 がベストであったが，ROUGE-2 スコアにおいては新規トークンを追加しないケース 2 の方が上回る結果となった．

ケース 3 で ROUGE-2 スコアが下がった要因は，新規トークンを追加したこととそれらの埋め込みベクトル初期値を平均ベクトルとして設定したことの影響だと思われる．しかしながら，テストデータにおいて新規トークンは原言語，目的言語いずれにも出現していなかった．現時点ではこれ以上の分析は行えていない．

表 5　評価指標に基づくスコア

評価指標	ケース 1	ケース 2	ケース 3
BLEU	0.00226	0.14351	**0.15066**
ROUGE-1	0.11738	0.39476	**0.42449**
ROUGE-2	0.00294	**0.21125**	0.20231
ROUGE-L	0.11062	0.39256	**0.41236**
TER	95.40816	72.44898	**70.10309**

4 議論

翻訳モデルの性能検証に用いられることの多い代表的な指標は，複数指標を組み合わせたとしても目安として機能するに留まり，今後どのような改善をしていくべきかを検討するにはドメインに応じた個別指標の検討が必要であることがわかった．

今回の日琉翻訳タスクにおいては，文例が句に準ずる形で区分けされていることと，文例以外の情報源として単語訳が利用可能である可能性が高い．これらを活用することで，名詞のように活用することのない体言を主体とした名詞句抽出に基づいた細かな評価指標の設計を検討中である．なお，琉球諸語に限った話ではないが，名詞には複数候補があり得ることから多対多を踏まえた評価指標の検討も必要だと思われる．

もし，名詞句に基づく分析を高品質で行えるならば，ここで用いたルールに合致する範囲で部分翻訳した中間言語文を作成し，「日本語→中間言語」をルールベース翻訳で行い，「中間言語→琉球諸語」を機械翻訳対象とすることでルール化困難な部分に焦点を当てて学習させることがしやすくなると思われる．

一定程度の翻訳品質を確保できた後は逆翻訳 [8,9] による疑似コーパス生成により頑健性向上を検討したい．しかし今回の設定では，日本語文からは区切り文字を除去し，琉球諸語文からは区切り文字を残したまま扱っていることの影響（区切り文字を含まない琉球諸語文を扱うことの影響）を検討できていない．まずは日琉コーパスのみで琉日翻訳するところから取り組みたい．

新規追加したトークンの初期埋め込みベクトルについては単純に平均ベクトルとして設定したが，これがベストであるかは議論の余地がある．また既に登録されていたトークンについては琉球諸語と異なる語彙として学習されている可能性が高い．語彙分析や異なる語彙の共存方法等を検討する必要があるだろう．

これらに加えて，検証用・テスト用コーパスの設計が必要だと思われる．文例が方言辞典における例文であることから，全ての単語を学習させるにはすべての例文を学習させる必要がある. このためには，（分量はさておき）均衡コーパス [10] のように全体像を把握できるような複数観点からの例文を含む形で検証用・テスト用コーパスを構築することが望ましいだろう．

5 まとめと今後の課題

琉球諸語辞典に含まれる対訳文を前提とした日琉翻訳モデルの構築に向けた検証を行った．実験結果から小規模対訳コーパスであったとしても，原言語を含む異なる対象言語（英語）に対する翻訳学習を行っておくことが、本来の対象言語（琉球諸語）に対する翻訳品質向上に寄与することを確認した．今後は名詞句分析を中心とした評価指標構築，ルールベース翻訳した中間言語を介した翻訳、検証用・テスト用コーパスの設計を検討していきたい．

参考文献

[1] −. **危機言語・危機方言の記録と記述と復興**. 国立国語研究所. (オンライン) (引用日：2023 年 12 月 18 日.)
https://www2.ninjal.ac.jp/openhouse/2020/pdf/b-03.pdf.

[2] **消滅危機言語の教育可能性を考える－多様な琉球諸語は継承できるか－.** 狩俣繁久. 島嶼地域の新たな展望－自然･文化･社会の融合体としての島々, 九州大学出版会, 2014 年, 263-279.

[3] −. **消滅危機言語としての琉球諸語・八丈語の文法記述に関する基礎的研究**. 基盤研究(A), 研究代表者: 狩俣繁久. (オンライン) (引用日 : 2023 年 12 月 18 日.)
https://kaken.nii.ac.jp/grant/KAKENHI-PROJECT-24242014/.

[4] **mT5: A massively multilingual pre-trained text-to-text transformer.** Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, Colin Raffel. Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2021, 483–498.

[5] **Introduction of the asian language treebank.** iza, Hammam and Purwoadi, Michael and Uliniansyah, Teduh and Ti, Aw Ai and Aljunied, Sharifah Mahani and Mai, Luong Chi and Thang, Vu Tat and Thai, Nguyen Phuong and Chea, Vichet and Sam, Sethserey and others. IEEE, 2016 Conference of The Oriental Chapter of International Committee for Coordination and Standardization of Speech Databases and Assessment Techniques (O-COCOSDA), 2016, 1–6.

[6] **BLEU: a Method for Automatic Evaluation of Machine Translation**. Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL), 2002, 311-318.

[7] **A Study of Translation Edit Rate with Targeted Human Annotation**. Matthew Snover, Bonnie Dorr, Rich Schwartz, Linnea Micciulla, John Makhoul. Proceedings of the 7th Conference of the Association for Machine Translation in the Americas: Technical Papers. 2006, 223-231.

[8] **Improving Neural Machine Translation Models with Monolingual Data**. Rico Sennrich, Barry Haddow, Alexandra Birch. Proc. 54th Annual Meeting of the Association for Computational Linguistics, Vol. 1, Long Papers, 2016, 86-96.

[9] **サンプリング生成に基づく複数逆翻訳を用いたニューラル機械翻訳**. 今村 賢治, 藤田 篤, 隅田 英一郎. 人工知能学会論文誌, 2020, 35 巻, 3 号, p. A-JA9_1-9.

[10] **KOTONOHA『現代日本語書き言葉均衡コーパス』の開発(<特集>資料研究の現在)**. 日本語の研究, 2008, 4 巻, 1 号, 82-95.

A 翻訳例

ケース 1〜3 のモデルを用いて，テストデータを与えた際の翻訳例を表 6 に示す．**太字**は対象言語と一致していると思われる箇所である（目視確認）．1 文のみ完全一致した生成文（no.9 の 2 文目）があり，初めて覚えた文は「俺じゃないよ（ワンネーアランドー）」であった．ただしこれは学習データにも含まれていた文例であった．ケース 3 は最後に句読点を複数書くことが多いようだ．

表 6　翻訳例

no	日本語文	琉球諸語文	ケース 1	ケース 2	ケース 3
1	姉さんが煮ないなら、私たち二人で一緒に煮よう。	_ アバーガ_ ニランラー、ワッタータイッシ_ マジョーン_ ニラ。	_ **アバー**ヤ_ チンヌ_ ヌ_ ヌ_ ヌ_ ヌ_ ヌ_ ヌ	_ **アバー**ターヤ_ **ニ**チェーグトゥヌ_ アバーガル_ チャクガ	_**アバー**ター**ガ**_**ニ**ーンナー、**ワ**ンガ_カム,...
2	今日はおまえが煮ろ。	_ チューヤ_ イャーガ_ ニレー。	_ ヌ_ チンムヲゥトゥルムンヌ _ ヌランヌ_ ナタン	_ **チューヤ**_ クーンネー・クーランネー_ **ニ**ラン。	_**チューヤ**_**ヤ**ーヲゥティ_**ニ**ーン。
3	お前が煮ないなら、私が煮よう。	_ イャーガ_ ニランラー_ ワンガ_ ニーサ。	_ ヌンナー?ゥンムンナー?	_ **イャーガ**_ **ニ**ーガ_ ゥンジャサンルアテーサニ・	_ヰキ**ガ**_ナーマジョーン、**ワンガ**_**ニー**...
4	せっかく、私がサトイモを煮たのに誰も食べなかった。	_ セッカク、ワンガ_ チンヌク_ ニチャシガ_ ターン_ カマンタン。	_ **カマン**サニ?ゥンナー?	_ ボーヤ_ ゥンム、**ワンガ**_ **ニ**ール_ **カマン**ル。	_ゥンニー、**ワンガ**_**カムタン**。
5	魚ぐらいたまには自分で煮てみろ。	_ イユヌアタイ_ マルケーテー_ ドゥークル_ ニチンデイ。	_ ヌーンヌーン_ イヨールヌ _ トーテーサ**ニ**ラン。。	_ **イユ**_ イチュナサテーグトゥ_ カラシガ_ **ニ**ール	_**イユヌ**_グニヤ_ヤーノー_キッサ...
6	魚を煮るなら、塩を少し入れろ。	_ イユ_ ニーラー、マース_ イフェー_ イリレー。	_ ムシテーサニ??????????????	_ **イユ**_ **ニ**チェーグトゥ、カラス_ グトゥ、カラス	_**イユ**ヌ_**ニ**チャガ_コーテーサグトゥ,...
7	魚を全部煮るまで待ってくれ。	_ イユ_ ムル_ ニールマディ_ マッチトゥラシェー。	_ ムシカラ_ ウッサ_ **ニ**ヤーヤ_ ニランバーイ・ヌ_ ヌ	_ **イ**ヨー_ **ニ**ラン_ **マ**ジョーン_ カマン。	_**イユ**ヌ_ワラベー_**マ**ジョーン_マ...
8	全部煮なくても家族の分はあるよ。	_ ムル_ ニランティン_ ヤーニンジュヌ_ ブノー_ アンドー。	_ **ヤ**_ ブラヤ_ ニ??	_ **ヤ**シェー_ **ニラン**_ モナコヤ_ ヌクヌ_ **ア**シバ。	_シル**ティン**_グ**ニ**ラン_マール_クーン...
9	全部兄さんがこそ 煮よったんだ。俺じゃないよ。	_ ムル_ アッピーガル_ ニータルヨ。ワンネーアランドー。	_ マンテ**ータ**ン。クンタンタンタンタンタンタンタンタンタンタンタン	_ すべてアシガ_ **ニール**_ッチー。**ワンネーアランドー。**	_アバーンディ_カムルムンヌ_ナービク...
10	魚は兄さんがこそ煮るんだ。ほかの人は煮ない。	_ イヨー_ アッピーガル_ ニール。ビチヌ_ チョー_ ニラン。	_ ムシゥーサグトゥ**ガル**_ カマンヤー。	_ **イヨー**_ カムサニ_ カム**ル。**アワリ_ **ニ**チェー	_**イ**ユヌ_カムル_クイチャヤ_シガ...

B 計算機資源を踏まえた見積もり

今回の検証は M1 MacBookAir 16GB で行った．日琉対訳コーパス 211 件を train:val:test=8:1:1 で分割することで，学習データ 168 件を学習時に用いた．1 エポックあたりにかかった実行時間は 2 分弱であり，大凡データ 100 件あたり 50 秒程度である．最終的にコーパスを 50 万件用意することができたとすると，同環境では 1 エポックあたり約 69 時間（約 2.8 日），10 エポックならば約 28 日かかることとなる．GPU 環境下で 10 倍程度に高速化できたとすると 10 エポックを約 2.8 日にて終了できそうだ．