Vol.2019-IFAT-134 No.3
Vol.2019-DC-112 No.3
2019/3/22

# ニューラルネットワークを用いた 英日・日英機械翻字

加藤 舜介[1,a)]　藤井 敦[2,b)]

**概要：**
翻字とは，ある言語における語を別の言語における音韻体系で綴ることである．本研究は，ニューラル機械翻訳（NMT）との類似性に基づく英日・日英ニューラル機械翻字モデルを実装し，NMT に関する 2 つの既存手法の翻字に対する有効性について調査した．また，日本語の表記にカタカナとローマ字を用いたときの，翻字への影響を調査した．その結果，NMT に関する手法により日英翻字の精度が向上し，日本語のローマ字表記は英日翻字の精度が向上することが示された．

# English-Japanese and Japanese-English transliteration using neural networks

## 1. 序論

科学技術や文化の発展により，多数の専門用語や固有名詞が次々に生まれている．それらの語を外来語として移入するために，発音で文字を当てる「翻字」が多用される．翻字により，例えば英語の “automaton” は，日本語の場合はカタカナで「オートマトン」と表記することができる．近年，ニューラルネットワークを利用した機械翻訳（NMT）に関する研究が盛んに行われている一方で，翻字の研究事例は少ない．そこで本研究は，NMT との類似性に基づく機械翻字モデルを実装する．

翻訳と翻字を比較すると，これらは共に記号列の変換であるが，翻訳は「文」を「単語列」とみなして原言語から目的言語に変換するのに対し，翻字は「単語」を「文字列」とみなして原言語から目的言語に変換する点が異なる．

- 文の翻訳：単語列の変換
  I / like / dogs / . → 私/は/犬/が/好き/です/。
- 単語の翻字：文字列の変換
  l/i/t/e/r/a/c/y → リ/テ/ラ/シ/ー

翻訳は「語順の交代」がない理想的な翻訳である．

NMT において，英日 NMT モデルと日英 NMT モデルの学習時に制約を追加し，2 つのモデルを同時に学習することで翻訳精度が向上する [4]．また，語順が似た言語間の NMT モデルは，原言語を末尾から入力してモデルの学習を行うことにより，翻訳精度が良くなることが知られている [2]．

本研究は，NMT に関するこの 2 つの技法を適用した英日・日英ニューラル機械翻字モデルを実装し，これらの技法の翻字に対する有効性を調査した．また，日本語の表記としてカタカナとローマ字を個別に使用して翻字精度を比較した．

## 2. 関連研究と本研究の位置付け

### 2.1 NMT モデル

本研究におけるニューラル機械翻字モデルは，NMT モデルを元にしたものであるため，まず NMT モデルについて説明する．本研究は，アテンションに基づくエンコーダ・デコーダモデル [1] を翻字に適用した．このモデルは，エンコーダにより入力列をベクトル化し，デコーダによりベクトル化された表現から適切な出力列を求める．エンコーダ・デコーダ間にアテンション機構を追加することにより，デコーダから出力を求める際に入力文内で注目すべき単語の情報を加えることができる．例えば，”I like dogs .” という英語を日本語に翻訳するとき，「私 は」まで出力したとすると，次に出力すべき単語を判断するときに，”dogs”に

1 東京工業大学工学部情報工学科
2 東京工業大学情報理工学院
a) kato.s.aw@m.titech.ac.jp
b) fujii@cs.titech.ac.jp

ⓒ 2019 Information Processing Society of Japan

注目すべきである，という情報がアテンション機構によりわかる．

原言語の文を $\mathbf{x} = x_1, x_2, ..., x_n$ とすると，目的言語の文 $\mathbf{y} = y_1, y_2, ..., y_m$ は以下の式から導かれる：

$$\mathbf{y} = \arg\max P(\mathbf{y}|\mathbf{x};\theta) \quad (1)$$

ただし，$\theta$ は NMT モデルのパラメータ集合を表す．

ここで，$\mathbf{y}$ の出力確率は以下の式で表される：

$$\begin{aligned} P(\mathbf{y}|\mathbf{x};\theta) &= \prod_{t=1}^{m} P(y_t|y_{<t},\mathbf{x};\theta) \\ &= \prod_{t=1}^{m} softmax(W * h_t) \end{aligned} \quad (2)$$

ただし，$y_{<t} = y_1, y_2, ..., y_{t-1}$ であり，$y_{<1} = \phi$ である．また $softmax$ は softmax 関数を表し，$W$ は NMT モデルのデコーダにおけるパラメータ，$h_t$ は入力文・アテンション機構・直前に出力した単語から計算されるデコーダの隠れ層の状態を表す．

NMT モデルの目的関数 $L$ は以下の式で表される：

$$L = -\sum_{<\mathbf{x},\mathbf{y}>} \log P(\mathbf{y}|\mathbf{x};\theta) \quad (3)$$

ただし，$<\mathbf{x},\mathbf{y}>$ は学習データ中の全ての対訳関係にある文 $\mathbf{x}$，$\mathbf{y}$ の組を表す．この $L$ を最小化するように，モデルのパラメータ集合 $\theta$ が更新される．

### 2.2 NMT に関する研究

小林ら [4] は，2 言語間の双方向の翻訳を考え，2 つのエンコーダの隠れ層の最終状態（中間表現）が一致するように学習することで，翻訳モデルの整合性が取れて翻訳精度が向上することを示した．Sutskever ら [2] は，語順が似ている言語の NMT モデルにおいて，原言語を逆順で入力してモデルを学習すると，原言語の原文における先頭の単語について目的言語における対応する単語との距離が近くなり，最適化問題が解きやすくなるため翻訳精度が上がることを示した．

### 2.3 機械翻字に関する研究

藤井ら [5] は，ローマ字を介して日本語・英語・韓国語の 3 ヶ国語の翻字辞書を既存の対訳辞書から自動で作成し，一つの枠組みで 3 ヶ国語の翻字を行った．鈴木ら [7] は統計的機械翻訳の手法を翻字タスクに応用することで翻字の精度を向上させ，またその翻字モデルを英日機械翻訳で用いることで，翻訳の質が向上することを示した．Liu ら [6] は，Liu ら [3] で提案された，NMT における出力を正順と逆順で行う 2 つのモデルを学習し，それらの出力単語列が一致するように調整することで，出力が安定し翻訳精度が向上する，という手法を翻字に対しても適用できることを示した．

### 2.4 本研究の位置付け

Liu ら [6] が NMT に関する技法は翻字に対して有効であることを示したように，本研究は小林ら [4]，Sutskever ら [2] の手法を翻字に適用することで，翻字精度を向上させることを試みた．また，藤井ら [5] がローマ字を介して翻字辞書を作成したのに対して，本研究はニューラル機械翻字モデルにおいて，学習時に日本語の単語表記にカタカナを使用した場合とローマ字を使用した場合の翻字精度を比較した．

## 3. ニューラル翻字モデル

### 3.1 基本モデル

本研究は，2.1 で説明した NMT モデルにおいて，文を単語，単語を文字に置き換えたモデルを基本モデルとした．この基本モデルに，3.2 で詳述する 3 つの手法を適用し，翻字精度の変化を調査する．

### 3.2 適用手法

#### 3.2.1 同時学習

2.1 で述べたように，基本モデルの目的関数 $L$ は以下の式で与えられる：

$$L = -\sum_{<\mathbf{x},\mathbf{y}>} \log P(\mathbf{y}|\mathbf{x};\theta) \quad (4)$$

小林ら [4] の手法は，英日・日英翻訳における 2 つの NMT モデルの中間表現が一致するように制約を加え，2 つのモデルを同時に学習する（ここでは便宜上「同時学習」と呼ぶ）．すなわち，英文を $\mathbf{x}$，日文を $\mathbf{y}$ とし，英日 NMT モデルの中間表現を $p_n$，日英 NMT モデルの中間表現を $p'_m$ とすると，目的関数は以下の式で表される：

$$\begin{aligned} L = &-\sum_{<\mathbf{x},\mathbf{y}>} \log P(\mathbf{y}|\mathbf{x};\theta_1) \\ &-\sum_{<\mathbf{x},\mathbf{y}>} \log P(\mathbf{x}|\mathbf{y};\theta_2) + ||p_n - p'_m||^2 \end{aligned} \quad (5)$$

ただし，$\theta_1$ は英日 NMT モデルにおけるパラメータ集合，$\theta_2$ は日英 NMT モデルにおけるパラメータ集合である．$||p_n - p'_m||^2$ が中間表現制約を表す．

#### 3.2.2 逆順入力

Sutskever ら [2] によると，語順が似ている言語間（例えば英仏）の NMT では，原言語を末尾から入力して学習した NMT モデルの方が，先頭から入力して学習した NMT モデルよりも精度が向上する（ここでは便宜上「逆順入力」と呼ぶ）．日英の場合，文単位で考えると語順が似ているとはいえないが，単語単位で考えると文字順が似ているといえる（図 1，図 2）．よって本研究は，ニューラル機械翻字における逆順入力の有効性を調査する．

ⓒ 2019 Information Processing Society of Japan

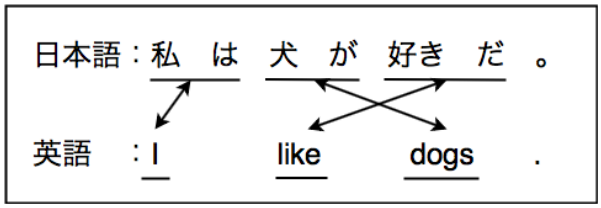


図 1　文単位で単語アライメントをとる場合

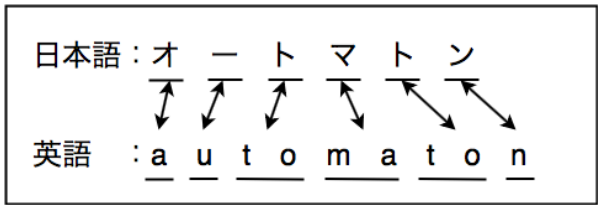


図 2　単語単位で文字アライメントをとる場合

#### 3.2.3　ローマ字表記

日本語はカタカナだけでなくローマ字を用いて表すこともできる．ローマ字を用いることにより，日本語の文字を母音と子音に分けて細かく表すことができるため，カタカナを使うよりも精度が上がるのではないかと考えた．そこで本研究は，日本語の単語表記にカタカナを用いた場合とローマ字（ヘボン式）を用いた場合を比較する．ただし，日本語の長音（ー）については，ローマ字ではハイフン（-）を用いて表記した．以下に例を示す．

- オートマトン → o-tomaton
- リテラシー → riterashi-

## 4.　実験

### 4.1　データセット

本研究の実験には，英日・日英機械翻訳用の専門用語辞書を使用した．この辞書から対訳関係にある日本語と英語の単語対が得られるため，そこから翻字関係にある単語対を抽出したものをデータセットとして使用した．単語対抽出の手順は以下の通りである．

(1) 全ての単語対のうち，日単語がカタカナで書かれている単語対を抽出する．
(2) 抽出した単語対の日単語をローマ字表記に直す．
(3) ローマ字表記に直した日単語とそれに対応する英単語の編集距離が 10 以下のものを翻字関係にある単語対として使用する．

ただし，置換の重みを 2 とすることで，例えば「トーチカ（to-chika）」と「pillbox」のように翻訳関係にあっても翻字関係にはない単語対を抽出することを減らした．

このように作成したデータセットから，学習データを 20,000 語，テストデータを 2,000 語として実験を行った．

### 4.2　評価手法

評価には正解率を用いた．ここで，出力単語と参照単語が完全に一致した場合のみを正解とする．ただし，このように評価を行うと日本語における「表記ゆれ」を考慮することができない．表記ゆれとは，”computer”を「コンピュータ」と「コンピューター」のどちらでも表すことができるというように，表記にばらつきが生じることである．そこで，5. では目視に基づく分析結果について考察する．

### 4.3　実験設定

日単語をローマ字に直すために pykakasi モジュール[*1]を利用した．翻字モデルにおける文字分散表現の次元数，隠れ層の次元数は共に 256 とした．

実装したモデルは以下の通りである．

英日翻字モデル・日英翻字モデルについて

(A) 適用手法なし（基本モデル）
(B) 同時学習のみ適用
(C) 逆順入力のみ適用
(D) ローマ字表記のみ適用
(E) 同時学習・逆順入力を適用
(F) 同時学習・ローマ字表記を適用
(G) 逆順入力・ローマ字表記を適用
(H) 同時学習・逆順入力・ローマ字表記を全て適用

これらのモデルの精度を比較する．

### 4.4　実験結果

実験結果を表 1 に示す．

表 1　テストデータに対するそれぞれのモデルの正解率 (%)

| モデル名 | 英日 | 同時学習 | 逆順入力 | ローマ字 | 日英 |
|---|---|---|---|---|---|
| A | 39.20 | | | | 23.95 |
| B | 40.30 | ● | | | 25.20 |
| C | 39.35 | | ● | | 24.70 |
| D | 44.15 | | | ● | 22.85 |
| E | 37.75 | ● | ● | | 25.10 |
| F | 41.40 | ● | | ● | 23.90 |
| G | 40.20 | | ● | ● | 25.70 |
| H | 42.00 | ● | ● | ● | 24.45 |
| | 40.54 | | 平均 | | 24.48 |

●は，モデルに適用した手法を表す．

結果をまとめると以下のようになる．

- 平均正解率は英日翻字で 40.54%，日英翻字で 24.48% であり，全体的に日英翻字よりも英日翻字の方が精度が高い．
- 同時学習の有無による精度の変化を調査する．すなわち，表 1 において，モデル A と B，C と E，D と F，G と H を比較すると，英日で平均 0.36% 精度が低下，日英で平均 0.36% 精度が向上している．
- 同様に逆順入力の有無による精度の変化を調査する

[*1] https://github.com/miurahr/pykakasi

ⓒ 2019 Information Processing Society of Japan

と．英日で平均 1.44% 精度が低下，日英で平均 1.01% 精度が向上している．

- ローマ字表記の有無による精度の変化を調査すると，英日で平均 2.79% 精度が向上，日英で平均 0.51% 精度が低下している．

## 5. 考察

### 5.1 英日翻字と日英翻字の比較

英日翻字の平均正解率は 40.54%，日英翻字の平均正解率は 24.48% となり，全体的に日英翻字よりも英日翻字の方が精度が高くなった．これは，日本語の文字と英語の文字の対応が一対多であることが理由にあげられる．

**表 2** 日英翻字の出力例：モデル A

| 入力 | 正解 | 実際の出力 |
|---|---|---|
| アクロース | acrose | a**ch**rose |
| ステライト | stellite | ste**ri**te |

表 2 は日英翻字モデルの出力の一例である．この例では「ク」を'c' と翻字するところを'ch' と翻字していて，下の例では「ライ」を'lli' と翻字するところを'ri' と翻字している．これは，「ク」は'c'，'ch'，「ライ」は'lli'，'ri' とそれぞれどちらでも表すことがあるためであるが，英単語として考えると'achrose'，'sterite' という単語は存在しないため翻字としては間違いである．

**表 3** 英日翻字の出力例：モデル A

| 入力 | 正解 | 実際の出力 |
|---|---|---|
| acrose | アクロース | アクロース |
| stellite | ステライト | ステライト |

表 3 は英日翻字モデルの出力の一例である．この例では'acrose' を「アクロース」と正しく翻字することができている．'c' は単語によって「ク」「シ」など読み方が変わるが，これは'c' の前後に続く文字を見ることで読み方を判断することができる．この場合，'c' の後に'r' が続くため「ク」と読めば良いとわかる．同様に，下の例でも'lli' を「リ」ではなく「ライ」と読むことを後の'te' を見ることで判断できる．

日英翻字の場合は，「〜ライト」で終わる単語は'-lite' と'-rite' のどちらで表すこともあるため，判断が難しい．このため，日英翻字は英日翻字に比べて精度が下がるのだと考えられる．

### 5.2 同時学習の影響

正解率を見ると．4.4 で示した通り，同時学習により英日で平均 0.36% 精度が低下し，日英で平均 0.36% 精度が向上している．しかし，同時学習を適用していないモデル A と同時学習を適用したモデル B における，テストデータに対する英日モデルと日英モデルの中間表現の平均距離を比較すると表 4 のようになった．

**表 4** 英日モデルと日英モデルの中間表現の平均距離

| モデル名 | 距離 |
|---|---|
| A | 12.95 |
| B | 13.07 |

これを見ると，同時学習を適用しても中間表現の距離はほとんど変化していないことがわかる．すなわち，同時学習の効果が現れていない．これは，目的関数における中間表現制約が効いていないためであると考えられる．

これを解消するには，目的関数に重みづけをして，中間表現制約を重要視すればよいと考えられる．例えば，式 5 を以下のように修正して学習を行う：

$$L = -\sum_{<\mathbf{x},\mathbf{y}>} \log P(\mathbf{y}|\mathbf{x};\theta_1) - \sum_{<\mathbf{x},\mathbf{y}>} \log P(\mathbf{x}|\mathbf{y};\theta_2) + W * ||p_n - p'_m||^2 \quad (6)$$

ただし $W$ は 1 より大きい実数である．

これにより中間表現が等しくなるような学習を行うことで，小林ら [4] はピボット翻訳への応用が期待できると述べている．翻字においても，第 3 言語に対する翻字モデルを作成する場合に，ピボット翻字が可能になると考えられる．

### 5.3 逆順入力の影響

3.2.2 で述べた通り，翻字関係にある日本語と英語を単語単位で考えると文字順が似ている．本研究の実験では，逆順入力を適用することにより英日で平均 1.44% 精度が低下し，日英で平均 1.01% 精度が向上した．よって，日英翻字で逆順入力の効果が出ている．

**表 5** 英日翻字の出力例：モデル G

| 入力 | 正解 | 実際の出力 |
|---|---|---|
| fabless | faburesu（ファブレス） | fu**reba**su（フレバス） |
| cineol | shineo-ru（シネオール） | shi**noe**ru（シネオル） |

英日翻字において，逆順入力を適用したことにより最も精度が低下しているのはモデル D，G の組（3.95% 低下）であるためこれらの出力を見てみる．表 5 は，モデル D は正解であるがモデル G は不正解である出力の一部である．これを見ると，文字順が一部バラバラになって出力されるという間違いを犯している．モデル D，G で不正解であった出力からそれぞれ 100 個抽出して見てみると，文字順がバラバラであるという間違え方が，モデル D で 4 個に対し，モデル G で 7 個であった．これは，逆順入力を適用したことによる悪影響が現れているためである．

これを解消するためには，単語の分割単位を変更すればよいと考えられる．本研究は，英語・日本語共に単語を一

ⓒ 2019 Information Processing Society of Japan

文字ずつに分割したが，英単語は部分文字列で分割することも考えられる．

- 一文字ずつ分割　：fabless → f/a/b/l/e/s/s
- 部分文字列で分割：fabless → fa/b/le/ss

このように単語を分割することで，英語と日本語の単語における文字アライメントが一対一となるため，文字順がバラバラになるという間違いは少なくなると考えられる．

また，本研究では，単語を先頭から入力，または末尾から入力のどちらか一方のみを考慮したモデルを実装したが，Bidirectional RNN を用いることにより，先頭・末尾からの両方向の入力を同時に考慮したモデルを実装することができる．これにより，モデルへの入力情報が増えるため，翻字精度はさらに上がると考えられる．

### 5.4 ローマ字表記の影響

英日翻字においては，日本語表記にローマ字を用いることで，平均 2.79% 精度が向上した．ここで，モデル A，D のどちらか一方のみ正解した英単語と，それに対応する日単語（ローマ字表記）の平均編集距離を比較すると表 6 のようになった．

**表 6** 英日翻字において，モデル A とモデル D で正解した英単語・日単語の平均編集距離

| モデル名 | 平均編集距離 |
|---|---|
| A | 4.561 |
| D | 4.337 |

表 6 を見ると，モデル D で正しく翻字した単語はモデル A で正しく翻字した単語に比べ，英日間の編集距離が小さいものが多いことがわかる．これは，日本語の文字を母音と子音に分けて細かく表すことで，日本語表記と英語表記が近いものになり，翻字しやすくなるためであると考えられる．すなわち，日本語のローマ字表記の翻字における有効性が現れている．

日英翻字においては，日本語表記にローマ字を用いることで，平均 0.51% 精度が低下した．表 7 に，モデル A は正解であるがモデル D は不正解である出力の一部を示す．

**表 7** 日英翻字の出力例：モデル D

| 入力 | 正解 | 実際の出力 |
|---|---|---|
| emyure-ta-（エミュレーター） | emulator | emu**r**ator |
| karushitonin（カルシトニン） | calcitonin | ca**r**citonin |

これらはどちらも”l”と翻字すべきところを”r”と翻字してしまった例である．このような間違いは，モデル A で 170 個あるのに対し，モデル D では 193 個あった．これは，ローマ字表記に”l”は用いられず，日本語のラ行の文字は全て”r”を用いて表されるため，日本語の”r”は英語においても”r”と翻字する傾向が強くなるように学習してしまったためであると考えられる．他にも，「ファ」を’pha’ ではなく’fa’ と翻字するような，ローマ字に近づけすぎる間違いが多くあった．具体的には，正解に比べてローマ字に近すぎる単語を出力してしまう場合が，モデル A は 384 個であるのに対し，モデル D は 536 個であった．それゆえ，日英翻字においては，日本語表記にローマ字を用いることで精度が低下したのだと考えられる．

## 6. 結論

### 6.1 まとめ

本研究ではニューラルネットワークを用いた英日・日英機械翻字モデルを実装し，NMT で精度が向上した手法を翻字に適用したときの影響を調査した．さらに，日本語の表記をカタカナにした場合とローマ字にした場合の翻字への影響を調査した．

小林ら [4] で提案された，両方向の翻訳モデルの中間表現が一致するように学習する手法を翻字に適用したところ，翻字の精度が向上することは確認できたが，2 言語のエンコーダの最終ベクトルの距離が小さくなることは確認できなかった．

Sutskever ら [2] で提案された，語順の近い言語間の NMT は原言語の入力を逆順にすると精度が向上するという手法は，日英翻字において有効であることが確認できた．

日本語の表記をカタカナからローマ字に変換して翻字モデルの学習を行うと，英日翻字では精度が向上したが，日英翻字では精度が低下してしまうことが確認できた．

### 6.2 残された課題

残された課題としては，以下の点が挙げられる．

#### 6.2.1 目的関数の重みづけ

本研究は，小林ら [4] の手法を用いても，2 言語双方向翻訳モデルの中間表現の距離が小さくなることは確認できなかった．しかし，目的関数に重みづけを行い，中間表現制約を重要視することで，中間表現が近い値になり，より翻字精度が上がると考えられる．

#### 6.2.2 単語の分割単位の変更

本研究は，英単語・日単語共に単語を一文字ずつに分割したが，英単語は部分文字列で分割することも考えられる．これにより，英語と日本語の文字アライメントを取りやすくなるため，翻字精度が向上することが考えられる．

#### 6.2.3 Bidirectional RNN の利用

本研究は，翻字の入力を正順または逆順の一方向しか考えなかったが，Bidirectional RNN を用いることで，両方向の入力情報をエンコーダに与えることができる．そのため，Bidirectional RNN を用いれば，本研究の結果に比べてより良い精度の翻字モデルを実装することができると考えられる．

ⓒ 2019 Information Processing Society of Japan

## 参考文献

[1] Dzmitry Bahdanau, KyungHyun Cho and Yoshua Bengio. "Neural machine translation by jointly learning to align and translate", Proceedings of ICLR 2015
[2] Ilya Sutskever, Oriol Vinyals and Quoc V. Le. "Sequence to Sequence Learning with Neural Networks", Proceedings of NIPS 2014
[3] Lemao Liu, Masao Utiyama, Andrew Finch and Eiichiro Sumita. "Agreement on Target-bidirectional Neural Machine Translation", Proceedings of NAACL-HLT 2016, pp. 411-416
[4] 小林尚輝, 田村晃裕, 二宮崇, 高村大也, 奥村学. "双方向翻訳のための中間表現制約を用いたニューラル機械翻訳", 言語処理学会 第 24 回年次大会 発表論文集 (2018 年 3 月), pp. 300-303
[5] Atsushi Fujii and Tetsuya Ishikawa. "Cross-Language IR at University of Tsukuba: Automatic Transliteration for Japanese, English, and Korean", Proceedings of NTCIR-4, April 2003-June 2004
[6] Andrew Finch, Lemao Liu, Xiaolin Wang and Eiichiro Sumita. "Target-Bidirectional Neural Models for Machine Transliteration", Proceedings of the Sixth Named Entity Workshop, joint with 54th ACL, pp. 78-82, 2016
[7] 鈴木久美, Colin Cherry. "カタカナ語から英語への翻字翻訳", 言語処理学会 第 16 回年次大会 発表論文集 (2010 年 3 月), pp. 202-205

ⓒ 2019 Information Processing Society of Japan