多言語機械翻訳システムの高精度化

Improvement of the accuracy of multilingual machine translation systems

徐恵[1)]
指導教員　亀田弘之[2)]，相田紗織[2)]，喜多義弘[2)]，渡邊紀文[3)]

1）東京工科大学　バイオ・情報メディア研究科
　コンピューターサイエンス専攻　亀田・相田研究室
2）東京工科大学 コンピューターサイエンス学部
3）武蔵野大学 工学部

キーワード：日中機械翻訳，精度向上，語順再配列，インターネット投稿

1. はじめに

近年，国際交流が盛んになるにつれ，言語障壁の問題解決のための翻訳研究が重視されている．その中，インターネットの十分な普及により，世界中の人々が投稿やコメントによって意思疎通や情報交換などを盛んに行っている．

しかしながら，インターネット上での投稿やコメントは曖昧で標準的な文型になっていないものが多い．そのため，現在の代表的な Google 翻訳機でも適切に訳すのが困難である．

Google 翻訳機は AI によるニューラル翻訳(NMT)という精度の高い方法を採用しているが，同じ意味の入力文であっても語順が異なると，相互に異なった意味の文が出力されることがある．つまり，翻訳結果が不安定という欠点がある[1][2]．例を図１に示す．

図１の示している通り，主語の位置を移すだけで翻訳の精度が上がっている．文を標準的な文型に直せば翻訳精度の向上が期待できる．

本研究では中日翻訳に注目し，構文を標準的な文型に再配列するアルゴリズムを提案し，Google 翻訳機の翻訳精度を向上することを目指す．

図１　主語の位置による翻訳結果の変化

2. 中国語の一般的な構成

中国語には「主語」「述語」「目的語」「定語（連体修飾語）」「状語（連用修飾語）」「補語」という６種類の文の成分が存在している[3]．

基本的な文型としては下記の３つがある．

- 主語＋述語
- 主語＋述語＋目的語
- 主語＋述語＋目的語１＋目的語２

3. 提案手法

3.1. データセットの用意

人気の社交サイトから標準的な構文になっていない投稿やコメントを収集する．

3.2. 単語の区切り

既存の単語の区切りソフトウェアを選択し自作のデータを図２のように区切る．

```
import jieba

path = '/Users/ouyangchen/Downloads/stanford-corenlp-full-2018-10-05'

nlp = StanfordCoreNLP(path, lang='zh')

sentence = "太偏激了吧这个电影。"

seg_list=jieba.cut(sentence,cut_all=False,HMM=True)

seg_str=' '.join(seg_list)

print(seg_str)
```

```
太 偏激 了 吧 这个 电影 。
```

図２　区切った結果

3.3. 構文解析

構文解析ソフトウェアを用いて単語に区切りしたデータの解析を行う．構文解析によって単語の成分(主語や述語など)と依存関係が示さる(図 3)．

```
token = nlp.word_tokenize(seg_str)

dependencyParse = nlp.dependency_parse(seg_str)

for i, begin, end in dependencyParse:
    print (i, '-',join([str(begin), token[begin-1]]), '-',join([str(end),token[end-1]]))
```

```
ROOT 0-, 6-电影
advmod 2-偏激 1-太
dep 6-电影 2-偏激
dep 2-偏激 3-了
dep 6-电影 4-吧
det 6-电影 5-这个
punct 6-电影 7-。
```

```
print(dependencyParse)
```

```
[('ROOT', 0, 6), ('advmod', 2, 1), ('dep', 6, 2), ('dep', 2, 3), ('dep', 6, 4), ('det', 6, 5), ('punct', 6, 7)]
```

```
print(token)
```

```
['太', '偏激', '了', '吧', '这个', '电影', '。']
```

図３　構文解析の結果

3.4. 提案アルゴリズムによる再配列

単語成分とそれらの相互依存関係を踏まえ単語を標準的な文型に直すアルゴリズムで再配列する．

「主語」を S,「述語」を V,「目的語」を O とし，標準的な文型を Sen とすると，アルゴリズムの一部は図４の通りになる．

- 入力: 原文
- 出力: Sen
- for 文の成分，依存する単語の番号，単語の番号 in 構文分析の結果：
  - if Sが見つかったら，then
    - Sの位置（番号）を獲得し，Sと他の単語の依存関係を調べる
    - if Sの前に依存する単語がなければ，then
      - Sを一番前に移す
    - else
      - 依存する単語を一番前に移し，Sをその後に移動する
  - else if...
  - else...
- 最終的には，Sen＝S＋V＋Oという基本文型に揃える
- print(Sen)
- end

図４　再配列アルゴリズムの一部

3.5. 再配列した文を翻訳機に通し結果を確認

本研究では Google 翻訳機を採用している．再配列した文を Google 翻訳機に通し，結果を確認する．

3.6. 評価方法

個人サイトを作り，文の入力から翻訳機に通すことまで行い，アンケート形式で提案手法の有用性を評価する．サイト画面を図５と図６に示す．

図５　個人翻訳サイトの画面

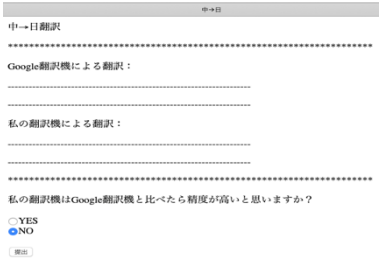

図６　Google 翻訳と比較したアンケートを集計

4. 実験

4.1. 単語区切りソフトの選定

中国語の単語区切りソフトウェア「Jieba」を採用する．Python ライブラリの観点から効率とバランスが最も良い単語区切りツールと検証されているからである[4]．用意した例文を「Jieba」で区切った結果を図２に示す．

4.2. 構文解析

構文解析はスタンフォード大学の自然言語処理グループが提供する Python の NLP ライブラリを採用している．単語区切りしたデータを NLP で構文解析をすると，結果は図３の通りになっている．

4.3. 再配列処理

構文解析したデータを再配列アルゴリズムに通し，文型を直す．結果を図７に示す．

図７　アルゴリズムによる再配列の結果

4.4. 原文と直した文を翻訳機に通し結果を確認

原文と再配列した文を翻訳機に通し，結果を確認した（図 1）．

5. 結果

例文の翻訳精度が向上し，提案した手法は有効であることが分かった．

6. 今後の予定

再配列のアルゴリズムを提案しているが，途中であるため性能は不足している．例えば，主語なしや，目的語が複数ある場合には対応していない．今後は可能な文の組み合わせを考慮しアルゴリズムを完成してから個人サイトによる評価を行う．

参考文献

[1] 後藤功雄，田中英輝，“ニューラル機械通訳での訳抜けした内容の検出”，自然言語処理，Vol.25 No.5 ， pp.577-597,2018.

[2] 田畑文也，“ニューラル翻訳を用いた中国特許機械翻訳精度の検証：中国特許の日本語及び英語への機械翻訳精度の検証”，第 14 回情報プロフェッショナルシンポジウム予稿集，pp.89-93，2017.

[3]https://www.hananiko.com/grammar/mandarin140/

[4]https://blog.csdn.net/adnb34g/article/details/87911995