記者会見通訳の二言語並行コーパスの構築～第2報～

山田優[1]　松下佳世[2]　石塚浩之[3]　歳岡冴香[4]　Michael Carl[5]

1 関西大学　2 立教大学　3 広島修道大学　4 近畿大学　5Copenhagen Business School

1 はじめに

本研究プロジェクトは，我が国の通訳翻訳研究の活性化を目指して，研究者が広く利用可能な日英の通訳の対訳コーパスを構築することを目的としたものであり，本年度で2年目を迎えた．今年度は英日の音声の同時通訳からテクストを書き起こす作業工程に IBM Watson Speech to Text の自動音声認識技術を応用するなど，コーパス構築のための技術的な仕様を整備し，試作データの構築を行った．また，同時通訳コーパス構築に用いた応用技術を，サイト・トランスレーション（視訳）の訳出プロセスデータ収集およびコーパス構築に応用する研究環境の開発も副次的に行った．本稿では，同時通訳のコーパス構築の技術的な仕様および構築方法の概観と同技術の応用利用例を紹介する．ここで扱うツール／プログラムおよびデータは，研究使用目的で自由に利用できる．

2 翻訳プロセスリサーチ・データベース（TPR-DB）

翻訳通訳学（Translation Studies）の分野で，その研究対象の射程をマップ化した Holms (cited in [1])は，研究全体を純粋な研究（Pure）と応用研究（Applied）に二分し，さらに前者の純粋研究を，理論的研究（Theoretical）と記述的研究（Descriptive）の下位部門に分割した．これにより研究が実証的な記述的研究へとシフトすることになる．この記述的研究の部門は，さらに訳出物志向（product-oriented），プロセス志向（process-oriented），機能志向（function-oriented）に細分化される．

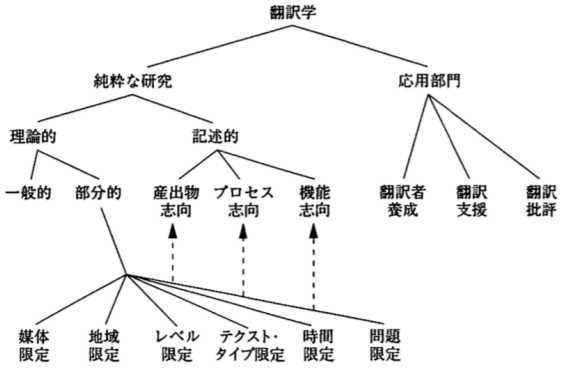


図 1：Holms による翻訳学の「地図」[1]

コーパスに基づく翻訳研究は記述的研究に属するが，コーパスというと，それまでは最終的な訳出物（product）のデータマイニングが中心であった．すなわち原言語と目的言語の組合せの訳文データからズレを探り，そこから翻訳方略やパターンを記述していくような訳出物志向の要素が強かった．しかし近年，訳出プロセス（process）を記録できる方法が確立すると，その訳出過程で記録できるデータも含めて研究の対象となった．例えば，広範に活用されるツールの一つである CRITT TPR-DB[1]は，Translog II[2]とアイトラッキング装置を組合わせて，人間翻訳者の翻訳中のキーボードストロークログと視線計測データを時間経過と共に記録する．翻訳を開始してから何分何秒に，原文のどの単語を読んでいて（視線の位置情報），何を訳しているのか（キーボード情報）を同定し，それらの情報から，どのような翻訳の困難に直面しているのか等を分析することが可能になる．

このように，コーパスベースの研究は，訳出物志向だけでなくプロセス志向の研究として確立した研究部門になった．しかし，書き言葉の翻訳に比べ，音声言語の通訳であるコーパスは，これまであまり存在しなかった．その要因の一つは音声データをテクストデータに変換しなければならないという技術的・人的工数的な壁があったとも言える．このようなそのギャップを埋めることが，本プロジェクトの目的の一つである．

3 対象データ

本研究でコーパス化する英語・日本語の通訳データ（音声および動画）は，昨年度の第 1 報[2]でも記した通り，公益財団法人・日本記者クラブの『日本記者クラブチャンネル』で公開されている映像素材で，記者会見データのうち，通訳付きの会見を行っているものが対象となる．同チャンネルでは過去6年分相当の通訳音声が含まれた会見映像が公開されており，そのうち，英語通訳を介して行われた 300 件ほどの映像・音声データが対象で，毎週数件の会見が新たに行われては，データは蓄積され続けている．会見は平均 1 時間程度．本プロジェクトでは，このうち約 100～200 件ほどのコーパス化を目標としている．

[1] The Center for Research and Innovation in Translation and Translation Technology Translation Process Database

[2] 以下よりダウンロード可：
https://sites.google.com/site/centretranslationinnovation/translog-ii

4 コーパスの仕様とデータ化の方法

本通訳コーパスで提供するデータの仕様は次の通りである. (1) 動画(MP4 形式), (2) 音声(wav:L チャンネル:英語, R チャンネル:日本語), (3) Speech to Text を使って書き起こしたテクストを音声波形に載せ, 単語レベルでのタイムスタンプ情報を含むデータ(フリーソフト ELAN 形式), また一部のデータは, (4) 原言語と目標言語を単語レベルで対応付けしたデータ(CRITT-DB の形式)を提供する.

本コーパス仕様の特徴は, 先述したように, 通訳コーパスを翻訳プロセス研究と比して研究できる素材にすることである. つまり, 翻訳プロセス研究で使われてきた CRITT TPR-DB ツールのデータとの互換性を, ある程度保てるようにしておくことである.

そのためには, コーパス情報の中に, タイムスタンプ情報が含まれていなければならない. これを実現するために, 本プロジェクトでは IBM Watson(Bluemix)の Speech to Text 技術を使う. また後に ELAN にインポートするための変換スクリプト(wav2Elan. py)[3]も用意した.

4.1. データ化の方法 (Speech-to-Text)

(1)動画と(2)音声データから, (3)のテクストへの書き起こし(トランスクリプト)にタイムスタンプ情報を付与し, フリーソフトウェアの ELAN にインポートするために, 上述したように IBM Watson(Bluemix)の Speech-to-Text を利用する. 同技術は IBM が提供する Web API サービスである。あらかじめダウンロードしておいた動画から, 音声チャンネルを分離した wav 形式音声ファイルを準備しておき, これらを専用のスクリプトを使って IBM サーバに API 経由で送ると, トランスクリプトと単語毎にタイムスタンプ情報が付与された情報が戻り, ELAN に直接インポートできる「タブ区切り」形式にまで変換できる(下記参照).

```
en  1910  2050  my
en  2050  2700  friendship
en  2700  3080  and
en  3250  3480  my
en  3480  4120  acquaintances
en  4120  4550  with
en  4870  4990  the
en  4990  5570  leadership
en  5780  5890  of
en  5890  5990  your
en  5990  6420  country
en  6420  6530  and
en  6530  6660  what
```

図 2 : IBM Speech to Text からの情報 (タブ区切り形式)

なお, 上のスクリプトは研究目的であれば無料で配布するが, API 本サービスを利用するには, IBM Bluemix と契約しておく必要がある(ID とパスワードが必要).

4.2. エラー修正

上記の Watson Speech to Text のデータには, 記者会見と実際の通訳音声であるという性質上, 音声認識エラーも含まれる. このため, プロジェクトでは, ある程度の品質精度を担保するために, ELAN にインポートしてから人手によってエラーの修正を行なう必要がある. 一つのデータにつき 2 名の作業者が確認・修正を行っている. 自動音声認識エラー率は, セグメント(分節)単位で 10%前後であるが, これをほぼゼロの状態まで向上させる.

4.3. アラインメントから分析データ生成

研究用に提供するデータのほとんどは, 上の ELAN データ+人手修正までで完成とするが, 一部のデータは, CRITT TPR-DB のプラットフォームに移管し, 付属のツール(YAWAT)を使って原文と訳文を言語単位でアラインメント(対応付け)する. この工程は, 従来の翻訳プロセスを CRITT TPR-DB のアップロードする手順と同じであるが, 今回のプロジェクトでは, 通訳の音声データを扱うため, アップロードをする前に, 専用のプログラム(StudyAnalysis. pl)で事前に変換する必要がある。こちらのプログラムも無料で提供する。

5 サイト・トランスレーションのデータ収集への応用

通訳コーパス構築プロジェクトを進めている過程で, 技術的に他の訳出モードでのデータ収集も対応が可能であることに気づき, 副次的にサイト・トランスレーションでの収集環境も整えることができた.

CRITT TPR DB は, これまでは書記言語の翻訳のプロセスデータ収集に主眼が置かれていた. つまり, 書かれた文字の原文→書かれた文字の訳文への訳出モードである(text to text). これに対して, 通訳は, 音声言語→音声言語への訳出のモードである(speech to speech). 2 つの中間に位置するのがサイト・トランスレーションで, 書かれた文字→音声に訳出するモードである(text to speech). サイト・トランスレーションは, 通訳現場で実際に用いられたりする. 例えば, 社内通訳で, メールの内容(書記言語)を口頭(音声言語)で訳して相手に伝えるといったケースがある. 通訳や翻訳の訓練法として用いられたりすることもある. キーボードを打つかわりに音声認識を使うのとは異なる. この場合は訳文が書かれた文字になるが, サイト・トランスレーションの場合は, 通訳のように口頭での訳出が最終的なプロダクトとなる点に注意されたい.

今回, 通訳コーパスの構築・ツール開発の過程で, この

[3] 本プログラムを希望する場合は、筆者に連絡のこと.

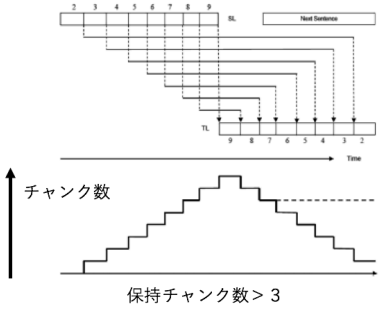
サイト・トランスレーションの訳出プロセス記録が CRITT TPR-DB でも可能になった．原言語がパソコン画面に表示されると，作業者はパソコンに接続されたマイクロフォンを通して口頭で訳出する．アイトラッキング装置も併用できるので，サイト・トランスレーションを行う時に，原言語のテクストのどの部分を見ているのか等の情報も記録できるというわけだ．

6 コーパスを使った分析例

では，今回の一連のデータ収集が可能になったことで，どのような分析ができるようになるのか，一例を記す．第 1 報でも述べたように，同時通訳の難しさは，原発話を聞きながら訳出を行わなければならない同時性にある．書かれた文字の翻訳の場合は，文の終りまで読んでから，逆送りで訳出することができるかもしれないが，同時通訳ではそうはいかない．この制約は，英語と日本語のように統語構造が鏡面関係にあるような言語ペアでは特に大きくなる．下の原文を和訳することを例に考えてみる．

【原文】
(1) The relief workers (2) say (3) they don't have (4) enough food, water, shelter, and medical supplies (5) to deal with (6) the gigantic wave of refugees (7) who are ransacking the countryside (8) in search of the basics (9) to stay alive.

【訳文 1〈逆送り〉】
救援担当者は (9) 生きるための (8) 食料を求めて (7) 村を荒らし回っている (6) 大量の難民達の (5) 世話をするための (4) 十分な食料や水，宿泊施設，医療品が (3) 無いと (2) 言っています



【訳文 2〈順送り〉】
(1) 救援担当者達の (2) 話では (4)食料，水，宿泊施設，医薬品が，(3) 足りず (6) 大量の難民達の (5) 世話が出来ないとのことです．(7) 難民達は今村々を荒らし回って，(9) 生きるための (8) 食料を求めているのです．

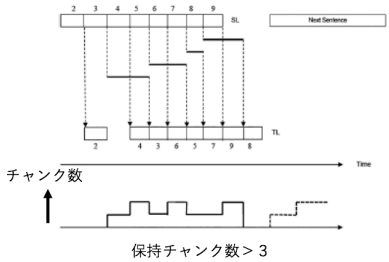


図 3：英日通訳における統語的鏡面関係と作動記憶のチャンク保持数[3]

便宜的に，訳出の単位（チャンク）を決めて，(1)(2)…のように番号を振ってある．〈逆送り〉の例では，(1) *The relief workers* に対応する部分の訳出(1)の救援担当者は が出た後に，続く原文の(2) *say* が，訳出されることなく，原文(9)の *to stay alive* に対応する *生きるための* が翻訳されている．これは〈逆送り〉の例であり，必ずしも訳として間違っていないかもしれないが，これを同時通訳で行うとなると，チャンク(1)の次に(9)を訳すまでの間の(2)〜(8)のチャンクを脳内の作動記憶に保持しておかなければならず，蓄積されたチャンク数が容量制限（通常は 2 から4チャンクと言われる）を超えてしまう可能性がある．つまり，同時通訳が失敗してしまうリスクが高まるのである．

これに対して〈順送り〉の訳では，〈逆送り〉に比べると，作動記憶内に蓄積されるチャンク数が平準化しているのがわかる．おおよそ原文の語順にそって，訳出が行っているのである．これにより同時通訳においても，作動記憶への負荷を回避でき，失敗のリスクを減らせるというわけだ．この点が，翻訳と通訳の方略の違いの一つである．

これらの訳出方略の違いを定量分析するには，原文と訳文の統語的な交差量（crossover）を見ることで算出できる（下図参照）．上の場合だと，〈逆送り〉と〈順送り〉の比較では，〈順送り〉のほうが統語的交差量は少なくなると予測される．すなわち書き言葉の翻訳と音声言語の同時通訳では，同じ原文を訳出した時でも，統語的交差量は，同時通訳のほうが小さくなるのである．さらに，CRITT TPR-DB ツール分析できる訳出プロセスデータも合わせてみると，訳出過程で困難な箇所などの同定も可能になろう．

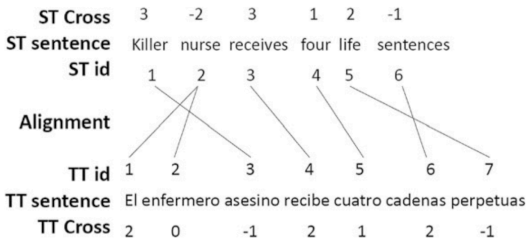


図 4：原文（ST）と訳文（TT）の交差値の考え方[4]

7 分析データの例

ここでは，パイロットで収集したサイト・トランスレーションのデータを例に訳出プロセスにおける「ポーズ（口頭の訳出が中断する時間）と「交差量」の関係を見る．同時通訳のコーパスでも同様の分析ができる．下図は【原文】から【訳文】を行っている様子を示している。

【原文】
Hospital nurse Colin Norris was imprisoned for life today for the killing of four of his patients.

【訳文】
病院勤務看護師コリン・ノリスは今日終身刑となりました．四人の患者を殺したためです．

開始時間	終了時間	訳文	ポーズ	原文	交差量
21490	21930	病院	0	Hospital	1
21930	22280	勤務	350	Hospital	1
22630	23000	看護	0	nurse	1
23000	23200	師	0	nurse	1
23200	23800	コリン・ノリス	249	Colin_Norris	2
24049	24309	は	691	Colin_Norris	2
25000	25300	今日	1780	today	5
27080	27500	終身	0	was_imprisoned_for_life	-4
27500	27900	刑	0	was_imprisoned_for_life	-4
27900	28000	と	0	was_imprisoned_for_life	-4
28000	28630	なりました。	1400	was_imprisoned_for_life	-4
30030	30320	[ええ]	740	---	0
31060	31580	四人	0	four_of	6
31580	31960	の	89	four_of	6
32049	32490	患者	0	his_patients	2
32490	32619	を	0	his_patients	2
32619	33210	殺した	0	the_killing_of	-6
33210	33520	ため	0	for	-1
33520	33750	です。	520	for	-1

図 5：サイト・トランスレーションのプロセス例

例えば，表内の訳文の*病院*は，原文の *Hospital* に対応する．この訳文が発話されたのは，開始時間 21490ms（ミリ秒）から 21930ms の間である．そして *Hospital* の訳語の*病院*は，文頭の語がそのままの位置で変わらないので統語的交差量は「1」（つまり交差していない）のがわかる．

それに続く訳文の*勤務*は，*病院勤務*というように一連の語として訳出されたので，*病院*と*勤務*の間に時間的ポーズは無く（ゼロで）発話されていた．しかし，次の*看護師*の訳出の前で，350ms ほどのポーズがある．これは，*hospital nurse* を*病院勤務看護師*と訳出しようとしたため，訳者が若干の困難に直面し，*病院勤務*と*看護師*の間に，ポーズが入ったのだと想像できる．とはいえ，350ms（0．35 秒）程度のポーズなので概ね訳出プロセスとしては，この時点では順調である．

問題は，*today* に対応する*今日*が訳出されてから，*終身刑*が出て来るまでに 1780ms ほどの空白がある．これは，*today* にたどり着くまでに *was_imprisoned_for_life* を超えて（交差量 5）の訳出となったので，統語処理に要した認知的困難（負荷）が高くなったと考えられる．

さらに，*was_imprisoned_for_life* に対応する*終身刑となりました*を訳出したあと，一旦，訳文を切る（終止符）と、*[ええ]*の淀みが挿入され，訳文の *4人の* が発話されるまでに，1400+740ms のポーズが生じている．1 文の原文を 2 文に分割して訳出するなどの工夫をして，〈順送り〉の方略を行っているのだが，一連のプロセスの節々に見られるポーズが思考（≒認知負荷）の痕跡となり，その付近の統語的な交差量も高くなっていることから，お互いの間に，何らかの関係（相関）があると想像できる．すなわち，〈順送り〉の方略は、英日の統語的な違いを乗り越えるためのものであり、それが通訳というタスクの困難の 1 つになっていることを同定する手がかりになると考えられるわけだ．

上でみた例は，単に訳出というプロセスの現象を記述したにすぎないが，通訳コーパスが大量に蓄積できれば，統計分析などもできるようになると考える．

8 まとめ

本稿では，記者会見通訳の二言語並行コーパスについて，その第2報として，技術的な仕様，構築法，そして分析例を報告した．本コーパスの作成目的は，通訳研究の向上と，通訳教育への貢献，そして分野を横断して，自然言語処理，認知科学，脳科学など多面的な活用を目論んでいると第 1 報でも述べたが，上記のような大量のデータを分析することにより，人間の通訳者の方略や困難についての同定をするプロセス研究に貢献するだけでなく，機械学習やニューラル通訳・翻訳機へのアプリケーションにも資するのではないかと考える．今後も引き続き広範な研究分野で多面的に活用を促し，研究領域を超えた意見交換を行い，総合的に進展していくことが望ましい．

謝辞

本研究は，JSPS 科研費 16H02915 の助成を受けている．

参考文献

[1] 鳥飼玖美 編（2009）『翻訳学入門』みすず書房

[2] 山田優，松下佳世，石塚浩之，歳岡冴香，Carl, M. (2017). 記者会見通訳の二言語並行コーパスの構築．言語処理学会第 23 回年次大会 発表論文集 (NLP2017), pp. 1168-1171.

[3] 水野的（2015）『同時通訳の理論―認知的制約と訳出方略』朝日出版

[4] Carl, M., Bangalore, S., and Schaeffer, M. (2016). New Directions in Empirical Translation Process Research: Exploring the CRITT TPR-DB. New Jersey:Springer.