スピーチレベルコーディングプログラムマニュアル

東京工芸大学工学部　木原諒子，片上大輔

　本マニュアルは，BTSJのコーディングルールに基づき，対話ログに対してスピーチレベルやスピーチレベルシフトのコーディングを行うプログラムについてまとめたものである．なお，作成したプログラムは人と対話システムの1対1の対話におけるログを対象にコーディングを行う仕様となっている．

# **1　プログラムの保存場所**

プログラムはGitHubに保存されており（<https://github.com/KiharaR/SpeechLevelCodingProgram>），以下のファイルが含まれる．

1. BTSJexcelLog.py
2. readme.md
3. スピーチレベルコーディングマニュアル.docx（本マニュアル）
4. SpeechLevelprogramフォルダ

環境構築等については，readmeもしくは本マニュアルの3章に記載している．

また，配布した「SpeechLevelprogram」には「dialogueLog」「rule」フォルダとtemplate.xlsxファイルが含まれており，「dialogueLog」内には実行確認用の対話ログが格納されている．

# **2　プログラムの概要**

　前述の通り，作成したプログラムは，BTSJのコーディングルールに基づき，発話文に対してスピーチレベルのコーディングを行うプログラムである．プログラムを実行すると以下のようにコーディング結果が出力される（表1）．

表1　本プログラムを用いた発話文のコーディング結果例（一部）



# **３　環境構築**

## **３．１　システム要件**

システム要件を見たいしていない場合，以下の方法でインストールしてください．

1. Anaconda

本プログラムはPythonファイルであるため，Pythonの実行環境が必要です．以下のリンクからAnacondaをインストールしてください．

* ダウンロード：<https://www.anaconda.com/products/distribution>
* インストール方法（Windows版）：<https://www.python.jp/install/anaconda/windows/install.html>

1. MeCab

本プログラムでは，発話文のスピーチレベルを判定するためにMeCabを使用します．以下のリンクからMeCabをインストールしてください．

* ダウンロード：<https://github.com/ikegami-yukino/mecab/releases>
* インストール方法（Windows10）：<https://self-development.info/mecab%E3%82%92%E3%82%A4%E3%83%B3%E3%82%B9%E3%83%88%E3%83%BC%E3%83%AB%E3%81%97%E3%81%A6python%E3%81%A7%E4%BD%BF%E3%81%86%E3%80%90windows%E3%80%91/>
  + パス設定は，以下のサイトを参考にしてください．
  + パス設定：<http://realize.jounin.jp/path.html>
* インストール方法（Windows11）：<https://self-development.info/windows-11%E3%81%B8%E3%81%AEmecab%E3%81%AE%E3%82%A4%E3%83%B3%E3%82%B9%E3%83%88%E3%83%BC%E3%83%AB/>

1. Cコンパイラ

* ダウンロード：https://visualstudio.microsoft.com/ja/visual-cpp-build-tools/
  + インストーラー実行後の選択画面では「C++によるデスクトップ開発」をクリックし，インストールしてください．

上記3つをインストールしたら環境構築は完了です．

# **3　プログラムの実行方法**

　以下，環境構築済みであることを前提に，プログラムの実行方法について説明する（本マニュアルではwindows11における手順を示す）．

1. 以下のいずれかの方法でソースコードをダウンロード

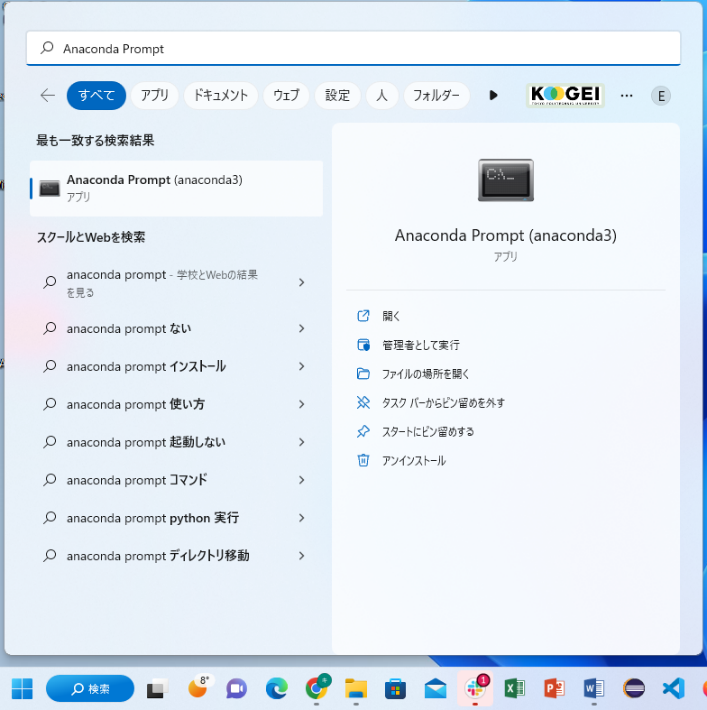
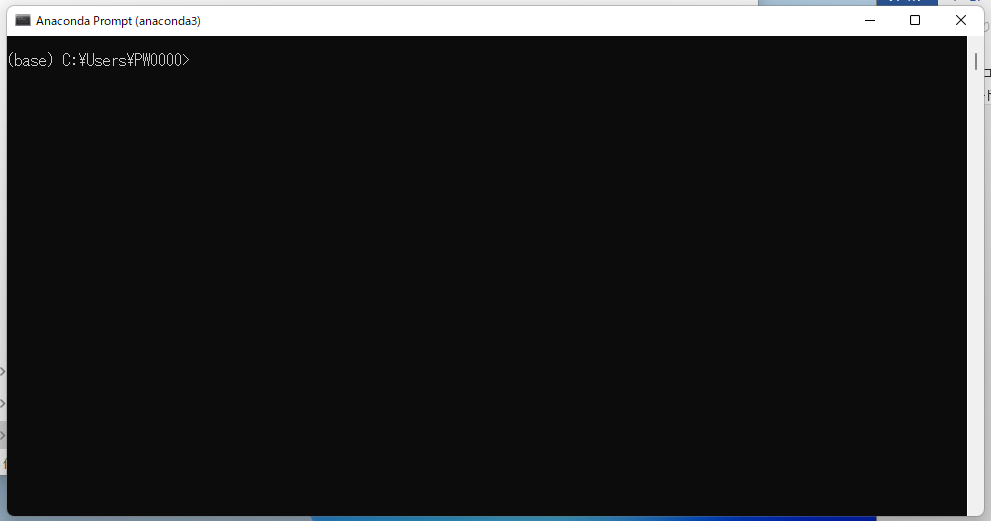
* cloneする場合：<https://codelikes.com/git-clone/>
  + git cloneの実行にはgitが必要です．
  + Anaconda Prompt上で実行する場合，gitがインストールされていなければ

|  |
| --- |
| conda install git |

と入力し，Gitをインストールしてから行ってください．

* 「Download ZIP」からダウンロードする場合
  + 「Download ZIP」を選択することでzipファイルがダウンロードされるため，PC側で解凍してください．

1. 「スタート」メニューから「Anaconda Prompt」と入力し，Anaconda Promptを起動する．



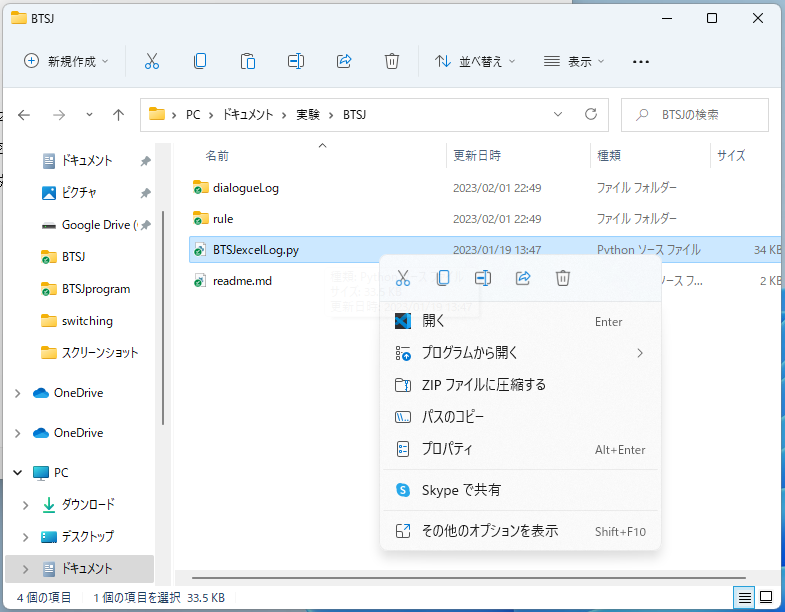
1. Anaconda Prompt上で以下のコマンドを入力し，Enterを押す．

|  |
| --- |
| cd BTSJexcelLog.pyのパス |

※ パスはBTSJexcelLog.pyを右クリックし，プロパティを開くことで確認できる．

※ BTSJexcelLog.pyの保存先ドライブとAnaconda Promptに表示されているドライブが異なる場合は，まず以下のコマンドを入力しドライブ移動する．

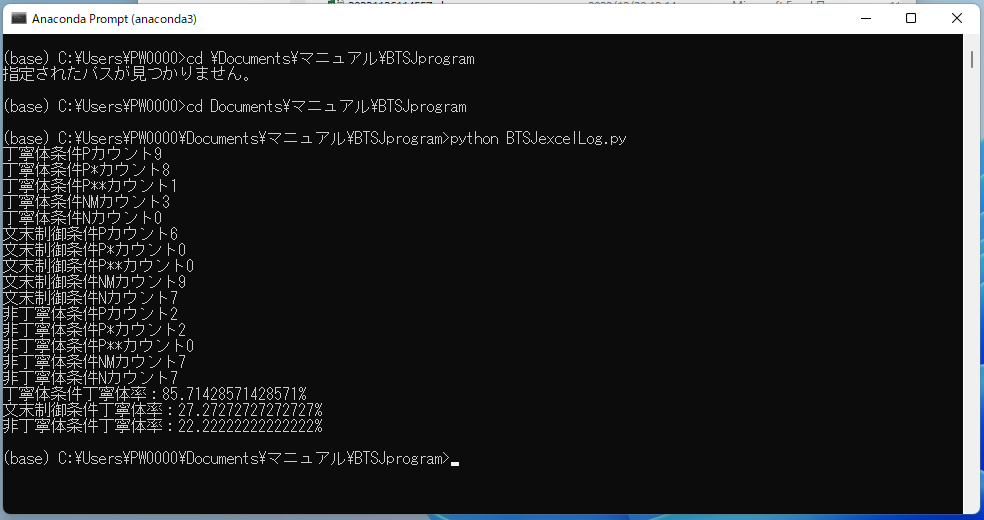
|  |
| --- |
| BTSJexcelLog.pyが保存されているドライブ: |

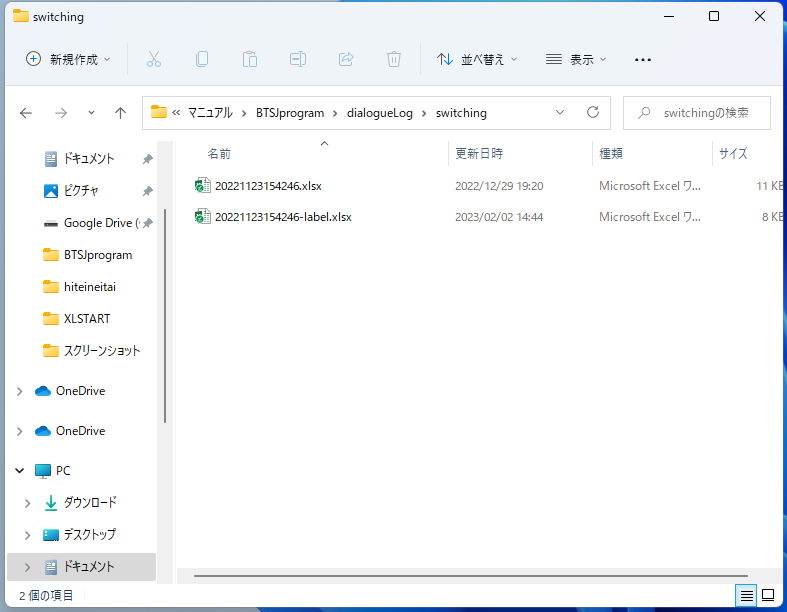


1. Anaconda Prompt上で以下のコマンドを入力し，Enterを押すとプログラムが実行される．

|  |
| --- |
| python BTSJexcelLog.py |

1. プログラムの実行が終了後，「dialogueLog」フォルダ内の各フォルダにコーディング後の出力結果（ファイル名に「-label」が含まれるファイル）が保存される．また，Anaconda Prompt上には各フォルダのスピーチレベルの合計や平均の丁寧体率（丁寧体の使用割合）が表示される．





注：プログラムを再度実行する場合，実行前に必ず「-label」を含むファイルを削除する必要がある（削除せずに実行すると，コーディング済みのファイルも再度コーディングされてしまうため）

# **4　本マニュアルにおけるコーディング基準**

## **4.1　スピーチレベルのラベル**

本プログラムでは，スピーチレベルのラベルを以下のように定義する（表2）．

表2　スピーチレベルのラベル

|  |  |
| --- | --- |
| ラベル | 定義 |
| P | 挨拶などの定型表現，「です。/ます。」，「ください。」 |
| P\* | 「です。/ます。」+α（※1），「ください。」+α（※2） |
| P\*\* | 「です、/ます、」+α（※1） |
| NM | あいづち・応答，形容動詞終了，名詞止め，言い淀み，名詞+助詞，中途終了型発話 |
| N | 上記以外 |

※1　αには「ね/よ/か/けど/から/って」が入る

※2　αには「ね/よ」が入る

なお，Pにおける挨拶などの定型表現は宇佐美[1][2]のスピーチレベルのコーディングに基づく（表3）．ただし，表3の赤文字で示した文章は独自に定義したものである．

また，NMにおけるあいづち・応答については，プログラムの仕様上すべてのあいづち・応答をコーディングすることができないため，本マニュアルでは，宇佐美[1][2]のスピーチレベルのコーディングと吉田ら[3]のあいづち表現認定基準に当てはまる発話とする．表4にNMとコーディングする文章を示す．

表3　Pにおける定型表現

|  |
| --- |
| P（定型表現） |
| よろしくお願いいたします　よろしくお願いします　申し訳ございません　申し訳ありません　おはようございます　こんにちは　ありがとうございます　ごめんなさい　すみません　はじめまして（初めまして）　こんばんは |

表4　NMにおけるあいづち・応答

|  |
| --- |
| NM（あいづち・応答） |
| うん　そう　はい　あ　え　うーん　ね　どうぞ　まあね　よろ　おう　ありあり　どうも　いえいえ　もちろん　あー　えー　いやいや　そだねー　そうだねぇ　そうだね　そうだな　せやな　はーい　ぜひ　まあ　いえ　はいよ　うい　ううん　そだね　うむ　そうね　うんうん　そうだねー　そうだなー　いいえ　だなー　だね　いや　なるほど　わかる　わかった　わかったよ　いいよ　いいよー　いいよね　ええんやで　いいだろう　なんだろう　とんでもない　なる　そんなことないよ　とくには　いいけど　まあ―まあ―　もちろん　それほどでも　あまり　また　特に　うふふ　たまに　そうかも　いますぐ |

※表4は，我々の実験にて取得した対話ログで見受けられたものであるため，すべてのあいづち・応答は網羅できていない

## **4.2　スピーチレベルシフト**

　本プログラムでは，宇佐美[2]の定義に基づき，スピーチレベルシフトを表5のように定義する．

表5　スピーチレベルシフト

|  |  |
| --- | --- |
| シフト | 定義 |
| ダウンシフト | P/P\*/P\*\*からNへのシフト |
| アップシフト | NからP/P\*/P\*\*へのシフト |

## **4.3　コーディング方法**

### **4.3.1　スピーチレベルのコーディング**

　発話文のスピーチレベルのコーディングは，1文ごとに行う（1発話ターンに複数文含まれている場合でも，1文ずつコーディングする）．なお，表6に示す記号が含まれている場合に1文としてコーディングするため，適切な1文をコーディングできない可能性がある．

表6　1文のコーディング基準

|  |
| --- |
| 記号 |
| 。　！　？　☺　（笑）　笑 |

※表6は我々の実験にて取得した対話ログにて見受けられたものを基準としている

### **4.3.2　スピーチレベルシフトのコーディング**

　スピーチレベルシフトのコーディングは，対象の発話文とその前ターンの発話文のスピーチレベルを比較することによりコーディングする（表7）．

表7　スピーチレベルシフトのコーディング例



ただし，上記以外の場合におけるスピーチレベルシフトのコーディングを以下に示す通りとする．

1. 発話文のスピーチレベルコーディングにP/P\*/P\*\*とNが含まれる場合は，必ずダウンシフトかアップシフトとコーディングする（表8）．

表8　 発話文にP/P\*/P\*\*とNが含まれる場合のスピーチレベルシフトのコーディング例



1. 1発話のスピーチレベルのコーディング結果が複数かつNMを含む場合は，NMを考慮せずスピーチレベルシフトをコーディングする（表9）．

表9　発話文にNMを含む場合におけるスピーチレベルシフトのコーディング例



1. 前ターンの発話文においてスピーチレベルのコーディング結果がNMの場合は，その前ターンのコーディング結果を参照してコーディングする（表10）．

表10　前ターンの発話文にNMが含まれる場合におけるスピーチレベルシフトのコーディング例



# **5　プログラムの注意点**

　作成したプログラムは，我々の実験で使用することを前提としているため，以下のように開発している．

* コーディングに使用する対話ログは，xlsxファイルのみに対応している．
* 現行バージョンでは，各15ターンまでのコーディングにのみ対応している．
* 現行バージョンでは，対話ログに含まれる「ユーザ年齢」の入力がなければプログラム実行時にエラーが出る可能性がある．
* 対話ログファイルに含まれる「system」の「発話内容」におけるスピーチレベルのコーディングは，BTSJのコーディングルールに基づくものではなく，「rule」フォルダ内のtxtファイルとの比較によりコーディングしている．
* 「rule」フォルダ内のtxtファイルと「system」の「発話内容」を書き換えることで，スピーチレベルのコーディングをすることが可能である（ただし，コーディングは「丁寧体，混在（丁寧体と非丁寧体），非丁寧体」の3つである）．
* スピーチレベルのコーディングは，BTSJのコーディングルールが基本となるが，我々の実験にて取得した対話ログにおける発話文が正しくコーディングできないものがあったため，一部独自に定義している．
* プログラム実行後に出力されるものは，それぞれ以下の意味を示す．
  + 〇〇条件〇カウント  
    各条件のスピーチレベルの合計を出力している．

なお，条件とはシステム発話のスピーチレベルのことを指している．以下に具体的に示す．

* + - 丁寧体条件（teineitai）：常に丁寧体
    - 文末制御条件（switching）：丁寧体と非丁寧体を使い分ける
    - 非丁寧体条件（hiteineitai）：常に非丁寧体

上記の（）に示すとおり，各条件は「dialogueLog」フォルダ内の各フォルダと対応している．

* 〇〇条件丁寧体率

各条件におけるuserの丁寧体率の平均を示している．

なお，丁寧体率は以下のように算出している．

# **6　プログラムの改良方法**

　開発したプログラムは，BTSJexcelLog.pyにおいて以下に示す行のコードを書き換えることで，様々な対話ログのコーディングを行うことができる．なお，プログラムを書き換える際には，VScodeを使用することを推奨する（<https://code.visualstudio.com/download>）．

1. スピーチレベルにおけるコーディング基準の変更

114-132行目は，定型表現やNMにおけるあいづち・応答のコーディングリストである．以下にそれぞれのリストの説明を示す．

* teikeiList1：Pにおける定型表現のリスト（「。」で終わる定型表現）
* teikeiList2：Pにおける定型表現のリスト（「！/？」で終わる定型表現）
* NMlist：NMにおけるあいづち・応答のリスト

コーディングリストの中身を変更したい場合，[ ]内を書き換えることで変更することができる．ただし，以下の2点に注意する必要がある．

* teikeiList1は「。」で終わる表現，teikeiList2は「！/？」で終わる表現として分類しているため，適切なリストを書き換えなければスピーチレベルのコーディングに誤りが生じる可能性がある
* teikeiList2とNMlistにおいて新たに文章を追加する場合，記号（「。/！/？」）の数だけ文章を追加する必要がある．

1. dialogueLogに含まれている対話ログ以外の対話ログ（独自に用意した対話ログなど）のコーディング

配布時にdialogueLogに含まれている対話ログ以外の対話ログもコーディング可能である．ただし，以下の要件を満たしていない場合はコーディングすることができない．

* template.xlsxの「発話内容」（C2-C33セル）※と「ユーザ年齢」（I1セル）に入力後，「名前を付けて保存」から適切なファイル名で保存されていること  
  ※「話者」が「system」となっている「発話内容」は書き換えても正しくコーディングされないため，入力済みのものからの書き換えは不要である
* 上記ファイルを「dialogueLog」フォルダ内のいずれかのフォルダに格納されていること

# **参考文献**

[1] 宇佐美まゆみ：スピーチレベルのコーディング\_ 修正版\_110128，2011

[2] 宇佐美まゆみ：【最新版】スピーチレベルとシフトのコーディングのルール 宇佐美 13

[3] 吉田奈央，高梨克也，伝康晴：対話におけるあいづち表現の認定とその問題点について，言語処理学会第15回年次大会発表論文集，2009

# **ライセンス**

本プログラムは，MIT Licenseで公開されています．詳細については，

<https://github.com/KiharaR/SpeechLevelCodingProgram/blob/main/LICENSE>

を参照してください．