Hibikino-Musashi@Home OPL「Go Get It Unknown Environment」の手法

１．提案手法の概要

音声情報から地点・物体の学習を行う．本タスクは学習時間が短いため，目標性能の確保にビックデータが必要な関数ベースの機械学習を避け，プロトタイプとの距離比較で認識を行うことで，データベース構築に必要な学習時間を短縮する．Training Phase では音声情報から場所と物体をデータベース化し，Test Phase では，音声命令から取得した単語とデータベースに格納された単語間の距離を算出することで，尤もらしい場所へ移動し命令を実行する．以下，両 Phase の説明を記す．なお，本タスクで用いた音声認識，形態素解析，SLAM，物体認識等の基本機能は文献[1]を参照のこと．

２．Training Phase

図１に，Training Phase における提案手法の概要を示す．また，図２にフローチャートを示す．まず，話者は，場所と物体の特徴を単語レベルでロボットへと教示する．例えば，TV，Red object などである．次に，教示により入力された英語音声を Google Chrome の Web Speech API を利用し文字起こしする．文字起こしされた文を，Double Metaphone アルゴリズムを用いて音声発音を表すコード（発音記号）に変換する．この時，物体の場所情報（TV などの場所の名前と座標）と特徴情報（Red object など）をそれぞれ別のデータベースに保存する．以上のステップを，物体および場所の特徴を判別できる程度に繰りかえして Training Phase を終了する．例えばリンゴの場合，Red object，foods，ball shape，sweet などの特徴を音声教示する．

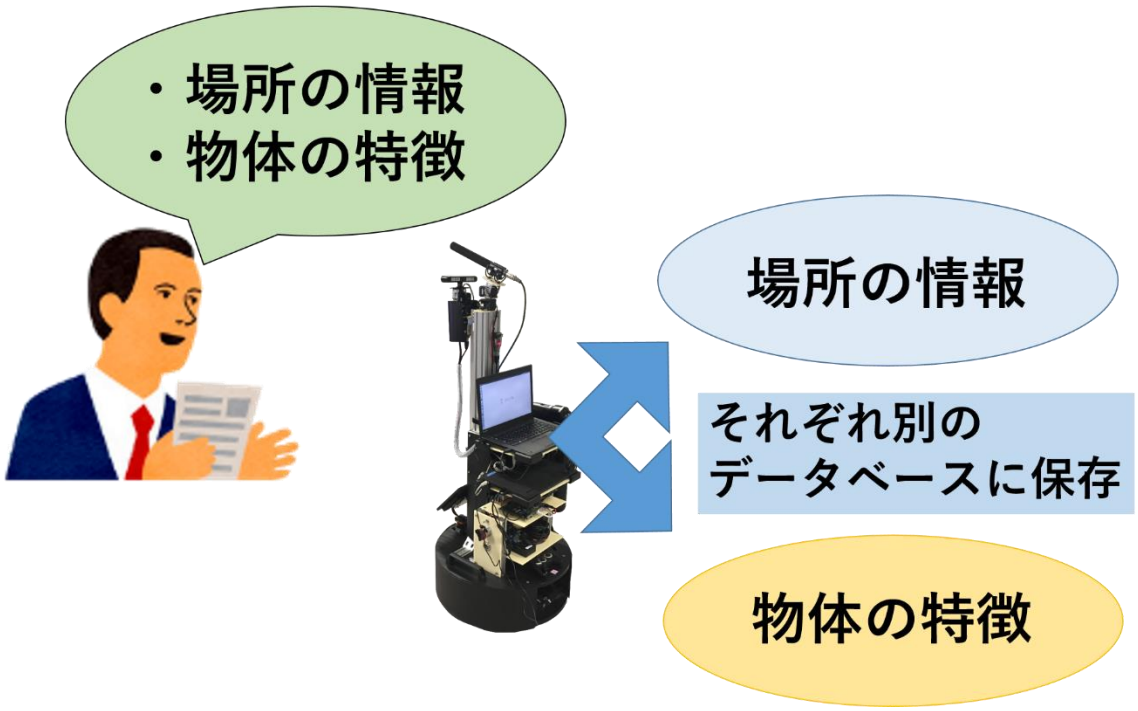

図１：Training Phase における提案手法の概要

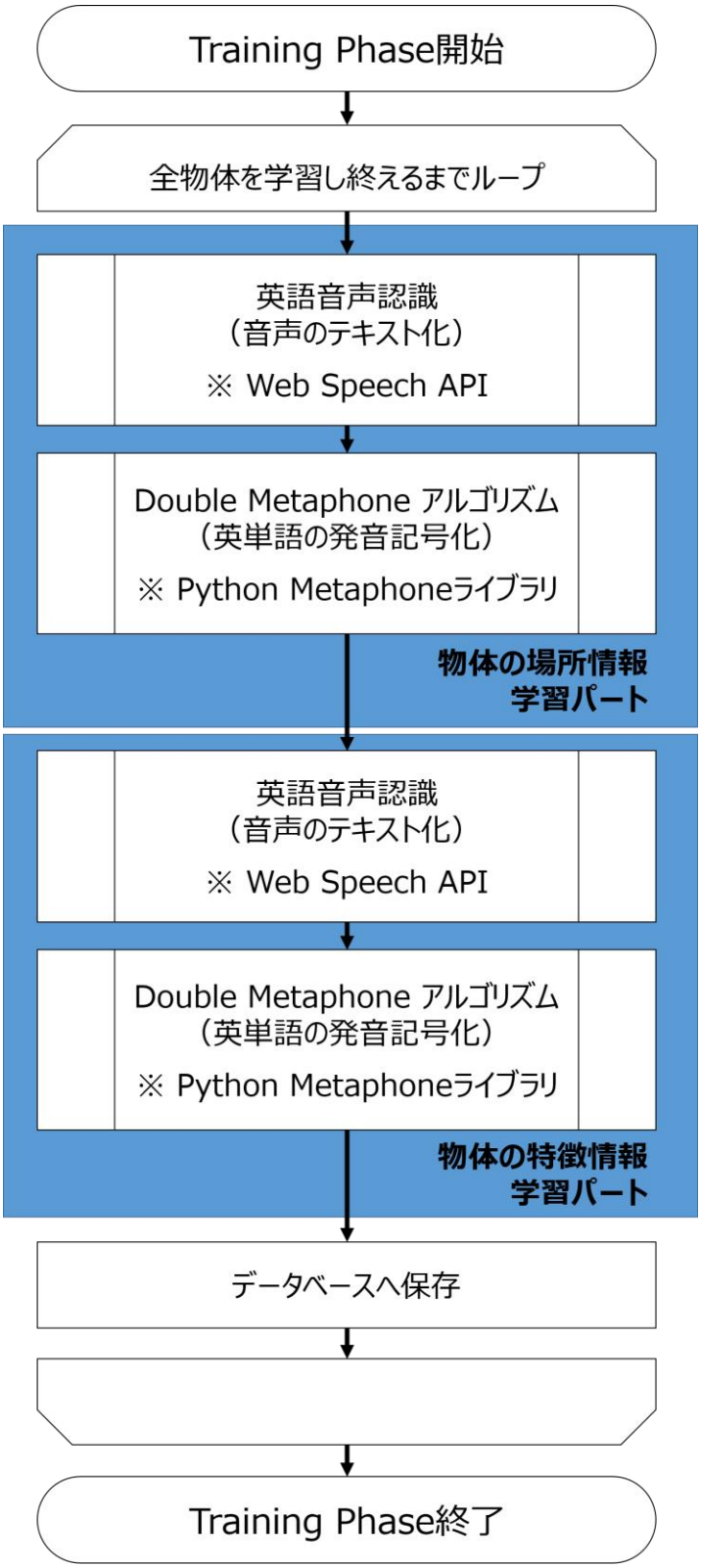

図 2：Training Phase のフローチャート

３．Test Phase

図 3 に，Test Phase における手法の概要を示す．図 3 左上に示すように，まず，形態素解析を用いた品詞分けにより，名詞（NN）と形容詞（JJ）を抽出する．この時，2 単語以上連続して入力があった場合，まとめて 1 語とする．例えば，Red object は形容詞＋名詞ではなく，“Red object”と 1 語にまとめて名詞とする．品詞分けされて抽出された単語を，学習時と同様に，発音記号に変換する．次に，学習時の発音記号とのレーベンシュタイン距離[2]を算出し，あらかじめ定めておいた閾値以下になった場合，命令文にキーワードが含まれているとカウントする．最もキーワードが一致した地点を候補とし，場所の情報を格納したデータベースより必要な情報（場所の名前と座標）を参照すると共に，オペレータに移動許可を求める．

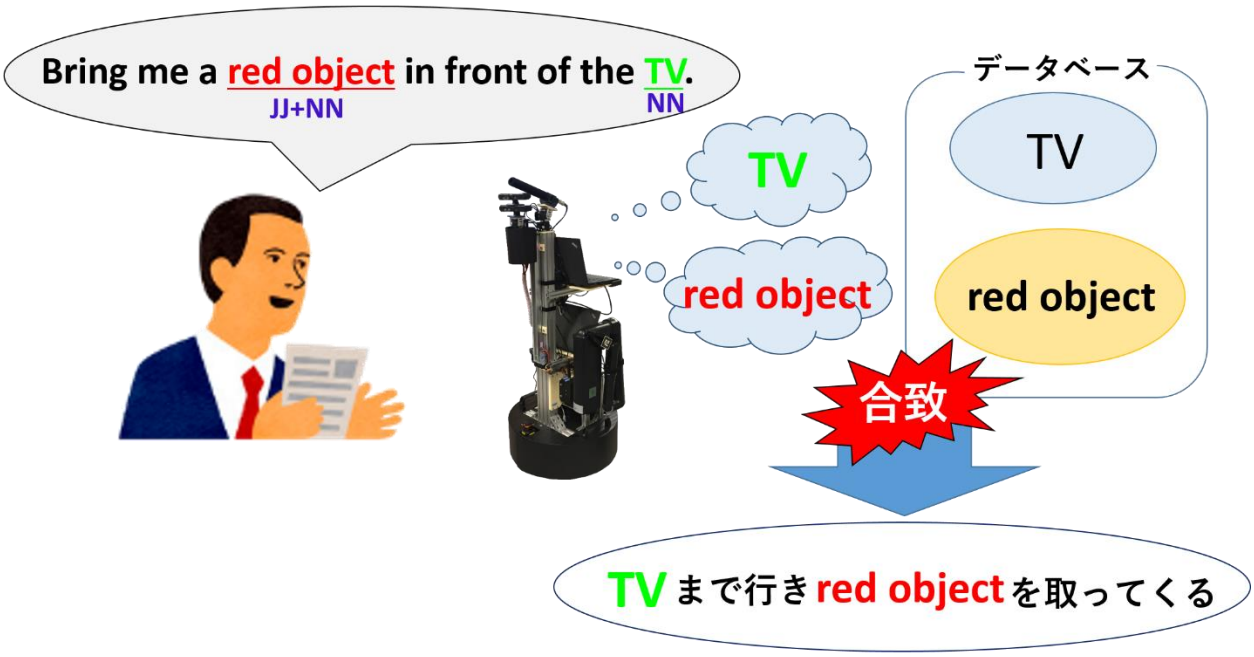

図 3：Test Phase における提案手法の概要

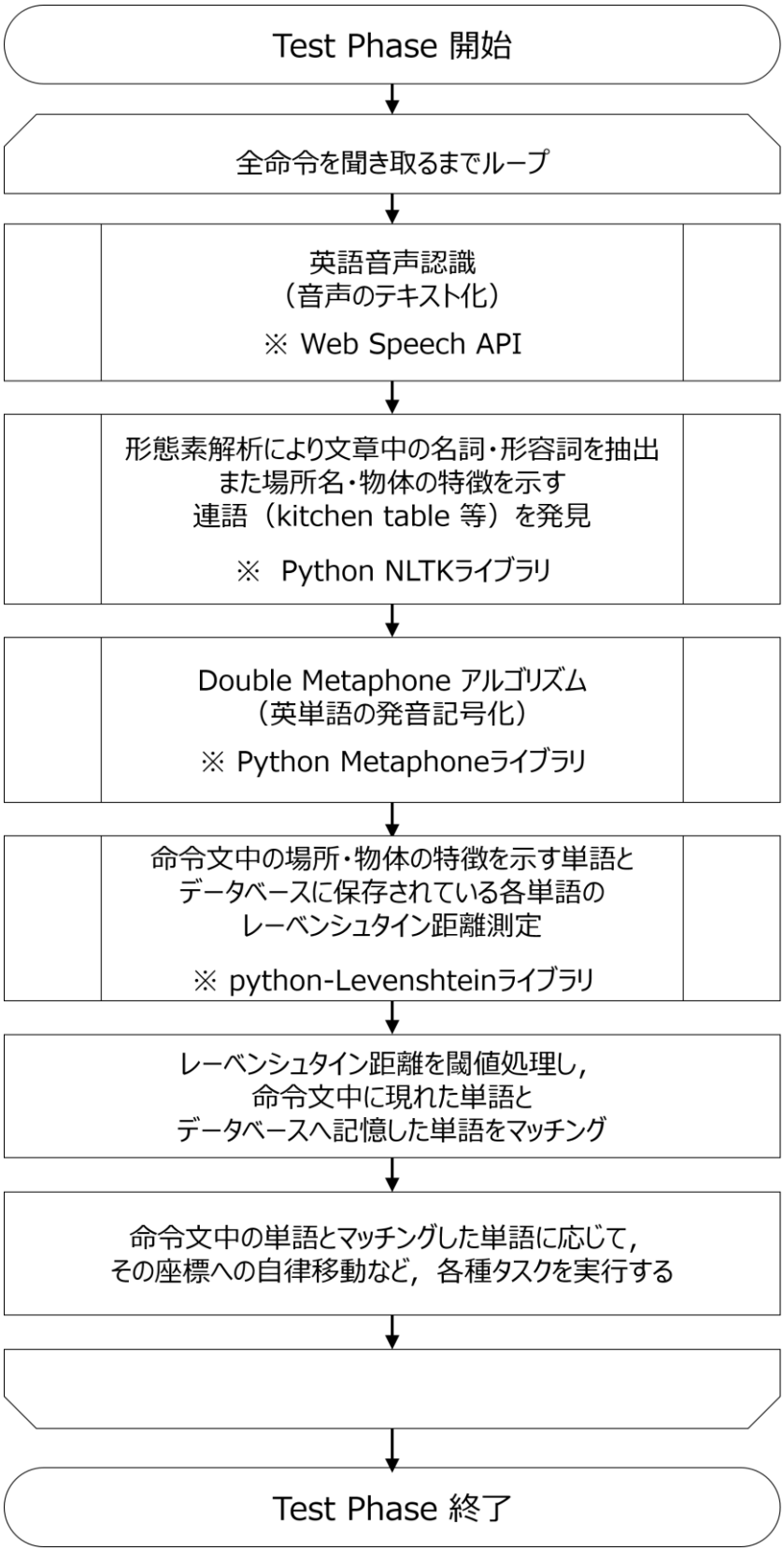

図 4：Test Phase のフローチャート

４．Double Metaphone アルゴリズム

Double Metaphone [3]は，アルファベットの発音に拘るのではなく，“英語の発音ルール”に基づき，英語の発音を記号へ変換するアルゴリズムである．綴りから母音を取り除き，子音だけで発音を近似する．発音ルールによる記号化により，本タスクにおいて，例えば，環境ノイズや日本人話者による不適切な発音での音声認識部の認識間違いの影響低減が見込まれる．

図５に，本アルゴリズムの概要を示す．例えば，「Python」は「PON」と変換され，また，「Bison」は「PSN」と変換される．この変換された２つの文字列に対して，レーベンシュタイン距離を計算することで，発音の近さを近似する．この例の場合，発音の近さは「1」となる．

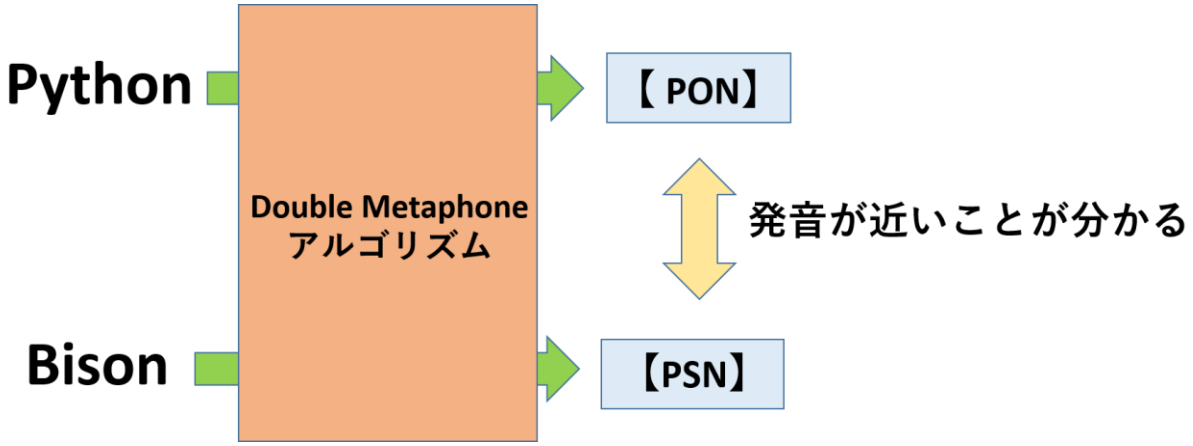

図５：Double Metaphone アルゴリズムの概要.

５．ソースコード

本タスク向けのソースコードを以下にアップロードする．

https://github.com/hibikino-musashi-athome/hma_exia_ggi

参考文献

[1] S. Hori, et al., “Hibikino-Musashi@Home 2017 Team Description Paper,” arXiv:1711.05457 [cs.RO], 2017.

[2] python-Levenshtein, https://github.com/ztane/python-Levenshtein, 2018 年 5 月 2 日アクセス.

[3] Metaphone 0.6, https://pypi.org/project/Metaphone/, 2018 年 5 月 2 日アクセス.