

ナレッジグラフ推論チャレンジ

VirtualHomeを対象とした生活行動を表現する 自然言語文からのアクションスクリプト自動生成

氏名：青山仁

所属：青山学院大学 工学部 情報テクノロジー学科

メールアドレス：jin2929@outlook.jp

取り組んだタスク：タスク2（生活行動ナレッジグラフの作成）

もとにしたデータ

- 生活行動を表現する自然言語による記述
 - VirtualHomeで提供されるデータセットに記載される行動に関する説明文
 - クラウドソーシングを用いて収集された*

Relax on sofa 行動に関するラベル

I turn on the tv and sit in the sofa. I watch the tv.

[WALK] <home_office> (1)
[WALK] <television> (1)
[FIND] <television> (1)
[SWITCHON] <television> (1)
[FIND] <couch> (1)
[SIT] <couch> (1)
[TURNTO] <television> (1)
[WATCH] <television> (1)

行動に関する説明文

アクションスクリプト

VirtualHome データセット例

*X. Puig, K. Ra, M. Boben, J. Li, T. Wang, S. Fidler, A. Torralba.
“VirtualHome: Simulating Household Activities via Programs” CVPR 2018.

生成したデータの種類

- 台本データ（アクションスクリプト）
 - VirtualHomeでは、アクションスクリプトを用いてシミュレーションを行っている。

“[Walk] <livingroom> (1)”

仮想ロボットの行動

行動の対象となる物や部屋

部屋や物の識別番号
(Object ID)

アクションスクリプトの例

- VirtualHomeでは、シミュレーション結果を動画として保存可能
- VirtualHome2KG*により、シミュレーション結果のナレッジグラフ化が可能

=> タスク2-1, 2-2を満たす。

*Egami, Shusaku and Nishimura, Satoshi and Fukuda, Ken.
“VirtualHome2KG: Constructing and Augmenting Knowledge
Graphs of Daily Activities Using Virtual Space” ISWC 2021.

生成したデータ数

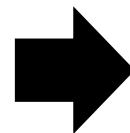
- 生成したデータ数 : 300個程度

変換例

- 変換例 :

入力データ

go to home office. turn the
computer on. begin working.

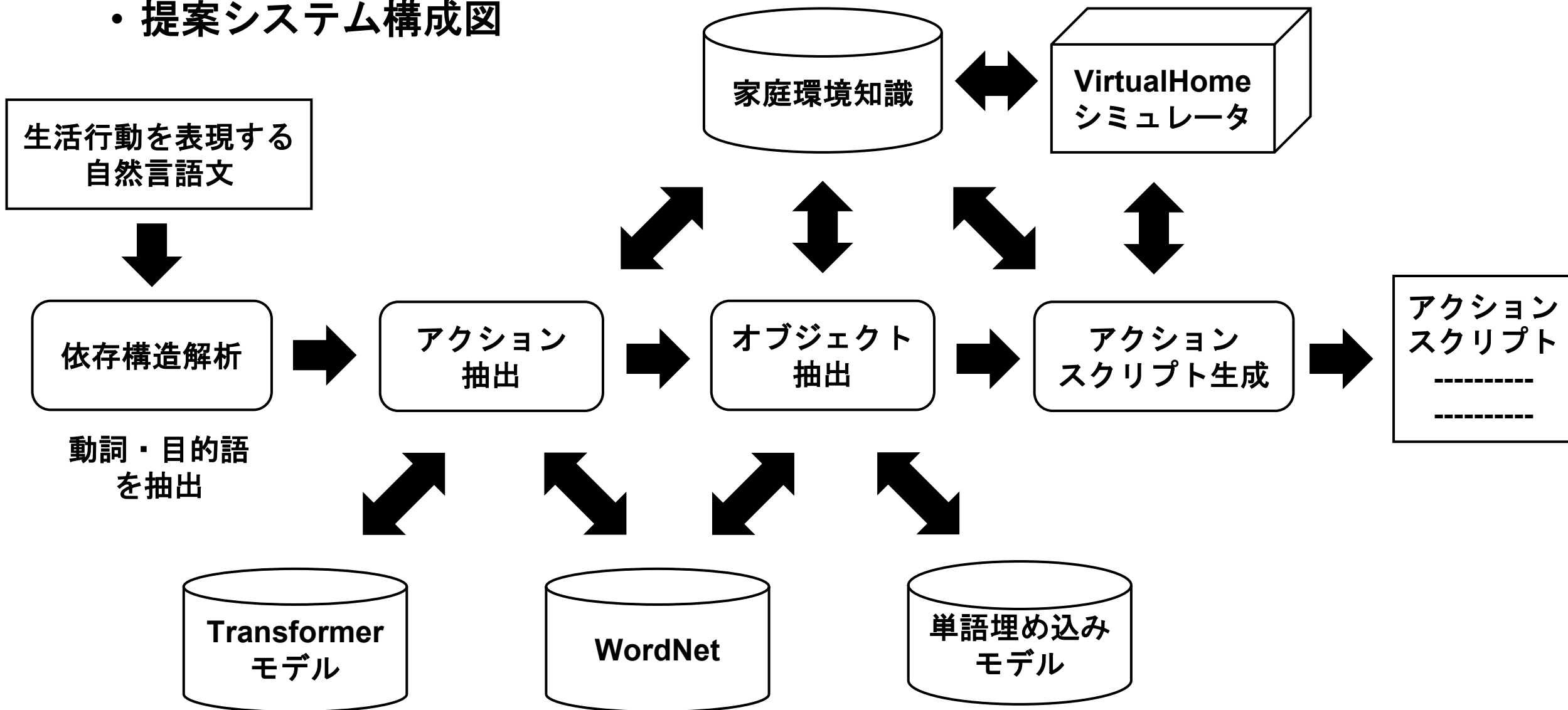


生成データ

[WALK] <livingroom>
[WALK] <computer>
[FIND] <computer>
[SWITCHON] <computer>

変換の技術的解説

・ 提案システム構成図



家庭環境知識 ①（変換の技術的解説）

• 環境グラフ

- VirtualHomeの環境は、環境グラフとして表されている。
- 環境内のオブジェクトに関する記述を”node”として表しており、識別番号、位置、状態、プロパティなどのデータが定義されている。
- 各オブジェクトの関係性を”edge”として表しており、”from_id”と”to_id”でオブジェクトの識別番号を指定して、”relation_type”で関係性を記述。

```
{
  "nodes":[
    {
      "id":427,
      "category":"Electronics",
      "class_name":"tv",
      "position":[5.26,0.00,-7.86],
      "properties":["HAS_SWITCH, HAS_PLUG"],
      "states":["OFF"]
    },
    {
      "id":336,
      "category":"Rooms",
      "class_name":"livingroom",
      "position":[3.64, 0.00, -5.49],
      "properties":[],
      "states":[]
    }
  ], "edges":[
    {
      "from_id":427,
      "to_id":336,
      "relation_type":"INSIDE"
    }
  ]
}
```

環境グラフの一部（要素の一部省略）

家庭環境知識 ②（変換の技術的解説）

・アクション（行動）

- ・ VirtualHome上で実行可能なアクションを、**アクションリスト**として扱う
=> VirtualHome Document を参照

・オブジェクト

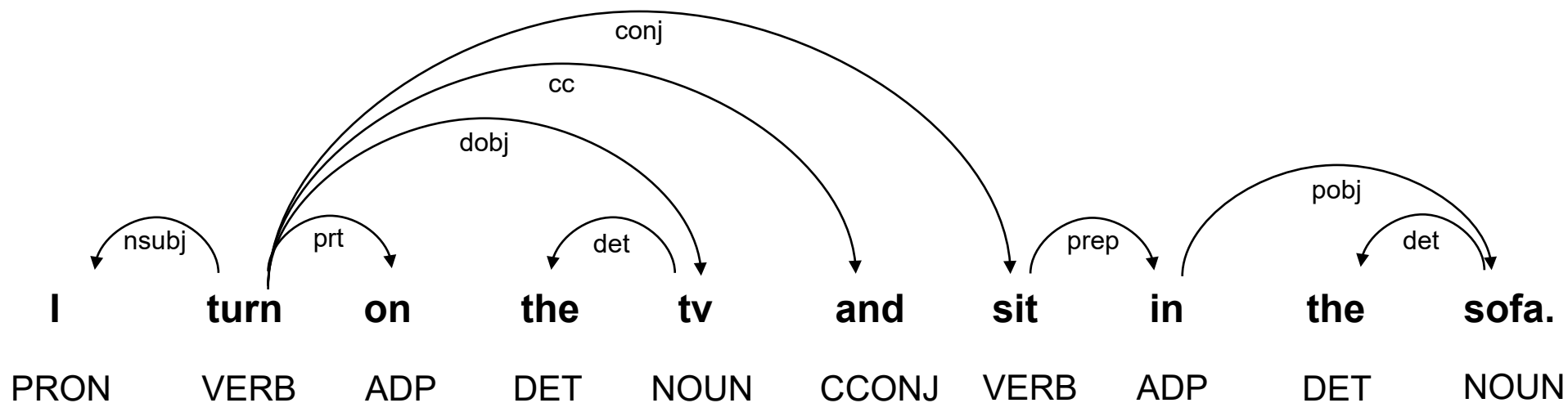
- ・ 環境内に存在するオブジェクトを、**オブジェクトリスト**として扱う
=> 環境グラフの”nodes”より取得

※ アクションスクリプトは、実行可能なアクション、環境内に存在するオブジェクトを正しく記載しないと実行できない。

=> アクションはアクションリスト、オブジェクトはオブジェクトリストから選択して、アクションスクリプトを記載する必要がある。

依存構造解析 ① (変換の技術的解説)

- 生活行動を表現する自然言語文を依存構造解析する。(spaCyを利用)
=> 動詞と動詞と依存関係にある目的語を抽出する。



依存構造解析の例

依存構造解析 ②（変換の技術的解説）

- 抽出結果

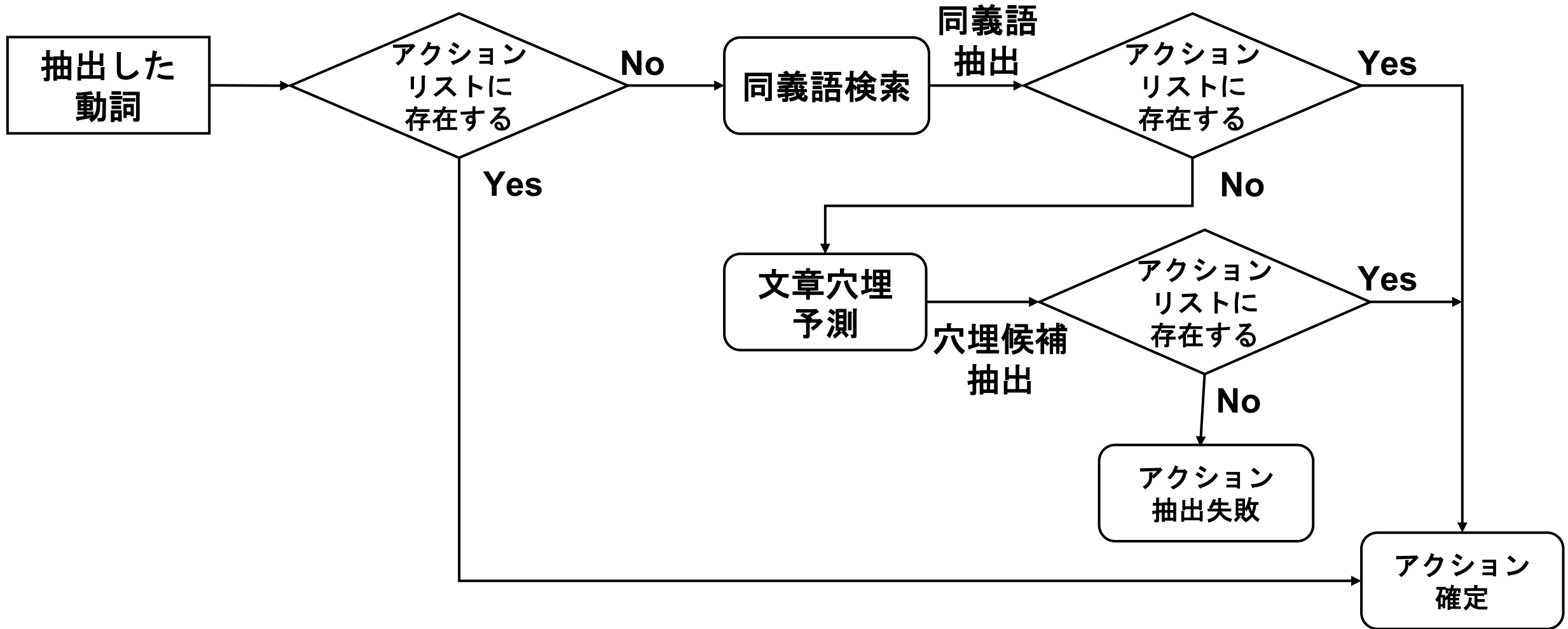
- {verb: **turn**, prep: **on**, objects: **tv**}
- {verb: **sit**, prep: **in**, objects: **sofa**}

※ 実行可能なアクションには, "switchon", "putin"など, 動詞+前置詞の形式で記載するものが存在するため, 動詞と依存関係にある前置詞も抽出しておく

- 抽出した動詞と目的語がそれぞれアクションリスト, オブジェクトリストに含まれるものではない。

- 抽出した動詞がアクションリストに含まれるかの判定, 実行不可の場合, 実行可能なアクションへの変換をアクション抽出モジュールにて行う。
- 抽出した目的語がオブジェクトリストに含まれるかの判定, 環境内に存在しない場合, 環境内に存在するオブジェクトへの変換をオブジェクト抽出モジュールにて行う。

アクション抽出（変換の技術的解説）



同義語検索（変換の技術的解説）

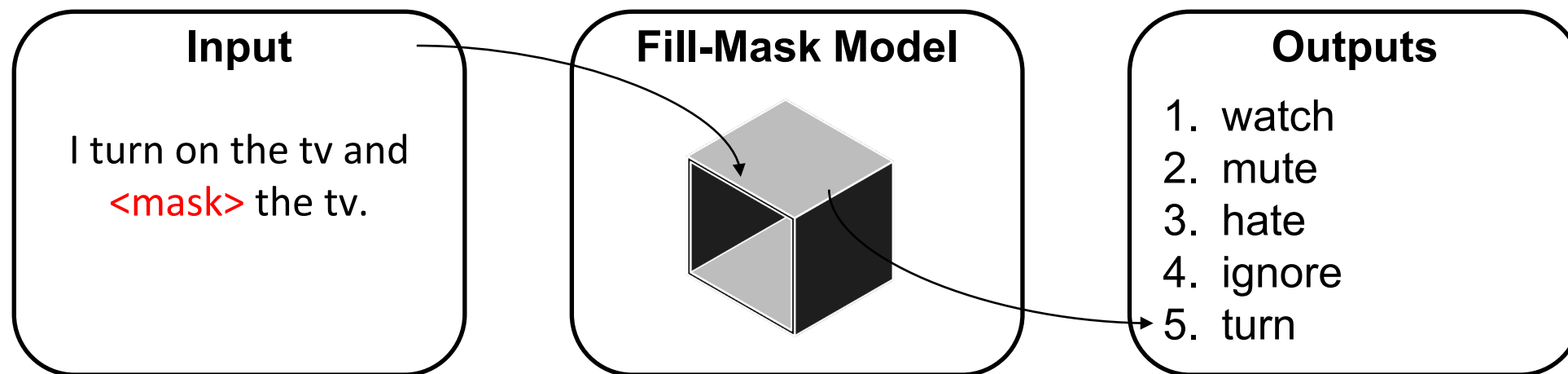
- NLTKライブラリの**WordNet** を利用して同義語を検索する.
- アクション抽出モジュールにて、依存構造解析で抽出した動詞がアクションリストにない場合に、動詞の同義語を検索して候補を挙げる.
=> 候補に挙げた同義語がアクションリストに含まれる場合は、その同義語をアクションとして確定する.

※ オブジェクト抽出モジュールでも同様

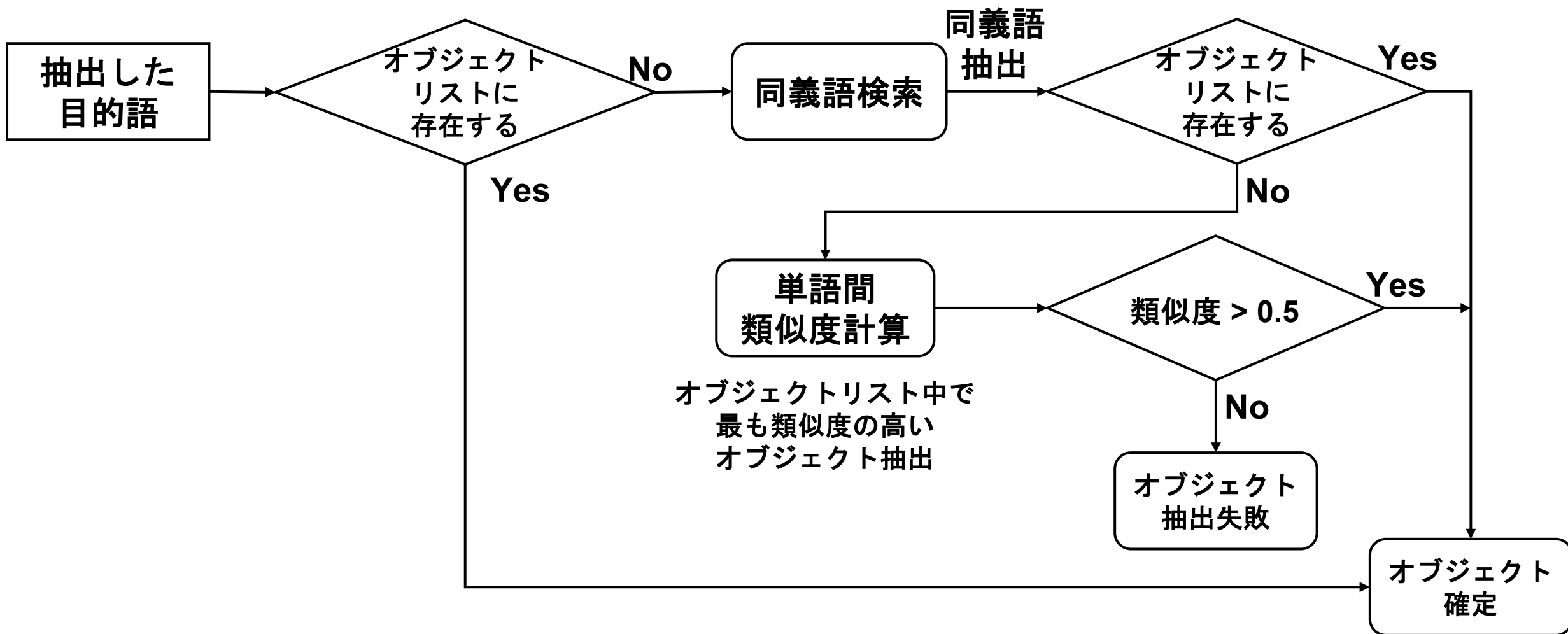
- 例：
 - "turn_on" の同義語："switch_on"
 - "television" の同義語："tv"

文章の穴埋め（変換の技術的解説）

- 入力文中の動詞部分をマスクして，マスク部分に入る単語の候補を挙げる.
=> 候補に挙げた単語がアクションリストに含まれる場合，その単語をアクションとして確定する.
- HuggingFace Transformersより提供される，DistilRoBERTa base の Fill-Mask model を利用している.



オブジェクト抽出（変換の技術的解説）



単語間類似度計算（変換の技術的解説）

- Gensimの**Word2Vec**モデルを用いて、単語間の意味的類似度を測る。
 - 依存構造解析より抽出した目的語と、オブジェクトリスト中の各オブジェクトとの意味的類似度を測る。
 - 類似度の最高点が0.5以上を満たす場合、最も類似度の高かったオブジェクトを抽出するオブジェクトとして確定する。
- ※ 意味的類似度（コサイン類似度）は、-1~1の間で表され、1に近いほど良い。
- 例：
 - “sofa” と “couch” の類似度：0.831

アクションスクリプト生成①（変換の技術的解説）

- アクションとオブジェクトが確定したら、アクションスクリプトを生成.
=> “[アクション]<オブジェクト> (オブジェクトの識別番号)”

※ 識別番号は同じオブジェクトが環境内に複数ある場合、オブジェクトを識別するためであるが、VirtualHomeのシミュレーション時の設定により、同じオブジェクトが環境内に複数ある場合、自動で探す設定が可能である。今回は、こちらの設定を想定する。

=> 識別番号の部分には何かしら記載する必要があるため、“1”を記載しておく。

アクションスクリプト生成②（変換の技術的解説）

- 各アクションには前提条件があり，それを満たす必要がある。
- 例：“[sit] <sofa> (1)” を生成する場合
 - “sit”を実行する前に，対象物に近づく必要がある => 事前に“walk”が必要
 - “sit”を実行する前に，対象物を見つける必要がある => 事前に“find”が必要
 - 対象物が “SITTABLE” のプロパティを持っている必要がある。

```
[WALK] <sofa> (1)
[FIND] <sofa> (1)
[SIT] <sofa> (1)
```

- 各アクションの前提条件とルール
 - VirtualHome Document を参考にルールを作成
 - => 条件を満たすようにアクションを追加
 - ※ 状態，プロパティ関連で条件を満たせない場合は生成をしない。

評価実験①

- 評価環境
 - 今回は, Scene1, 2, 4のVirtualHomeの環境を評価対象とする.
- 評価データ
 - VirtualHomeのデータセットを用いて, 正解データを, もとデータとして用いたデータセット中のアクションスクリプトとする. (環境ごとにデータが異なる)
 - データセットには実行ができないアクションスクリプトが記載されるデータセットも存在するため, 実行が可能なアクションスクリプトを含むデータセットのみ厳選して用いる.

VirtualHomeの環境	元のデータ数	評価用データ数
Scene1	518	174
Scene2	510	104
Scene4	519	132

評価用データセット数

評価実験②

- 評価方法
 - 生成データと正解データを比較し、正解数から適合率、再現率、F1値を求める。
- 評価結果
 - 3つの環境の全体の評価結果であり、適合率、再現率、F1値の平均値を示す。

	適合率	再現率	F1値
アクション & オブジェクト	0.64	0.56	0.57
アクション	0.74	0.62	0.64
オブジェクト	0.81	0.64	0.69

付録

- GitHubのURL : https://github.com/JinAoyama/generate_action_scripts
- 実行方法 : READMEファイルに記載