

知能プログラミング演習 II 課題 2

グループ 8

29114060 後藤 拓也

2019 年 10 月 28 日

■提出物 rep3

■グループ グループ 8

■グループメンバー

学生番号	氏名	貢献度比率
29114003	青山周平	no
29114060	後藤拓也	no
29114116	増田大輝	no
29114142	湯浅範子	no
29119016	小中祐希	no

■自分の役割 必須課題 3.3

「知識システムの質問応答システム」

1 課題の説明

課題 3-1 または 3-2 で作った知識表現を用いた質問応答システムを作成せよ。なお、ユーザの質問は英語や日本語のような自然言語が望ましいが、難しければ課題 2 で扱ったような変数を含むパターン (クエリー) でも構わない。

2 手法

1. 課題 2 で扱ったような変数を含むクエリーによる質問
2. 英語による質問

1. に関して,
2. に関しては,

3 実装

手法 1 に関する「課題 2 で扱ったような変数を含むクエリーによる質問」の部分を実装したソースコード 1 に示す。

ソースコード 1 1 文すべて終わったら格納する

```
1  /**
2   * 課題 2 で扱ったような変数を含むパターン (クエリー) による質問応答システム
3   * "?x is-a sports" と "?y hobby ?x" をとらえる
4   * → 質問は 3 つのトークンに分けられる
5   */
6 Scanner stdIn1 = new Scanner(System.in); // 文字列読み込み
7 Scanner stdIn2 = new Scanner(System.in); // 数値読み込み
8 ArrayList<ArrayList<String>> queryList = new ArrayList<ArrayList<String
   >>(); // 質問 (query) を入れる
9 StringTokenizer st; // トークンごとに分解
10 int retry;
11 do {
12     ArrayList<String> tokenList = new ArrayList<>();
13     System.out.println("質問を入力してください");
14     String s = stdIn1.nextLine(); // 質問文がここに入り,
15     st = new StringTokenizer(s); // トークンごとに分解し,
16     for(int i=0; i<st.countTokens(); i++) {
17         tokenList.add(st.nextToken());
18     }
19     tokenList.add(st.nextToken());
20     queryList.add(tokenList);
21     System.out.println("もう 1 つ? 1...Yes/ 0...No");
22     retry = stdIn2.nextInt();
23 }while(retry == 1);
24
25 ArrayList<Link> query = new ArrayList<Link>();
26 for(int i=0; i<queryList.size(); i++) {
27     query.add(new Link(queryList.get(i).get(1), queryList.get(i).get(0),
```

```
        queryList.get(i).get(2));
28 }
29 sn.query(query);
```

flag はフィールド変数として, Unifier クラスのどのメソッドでも用いられる. 上記には書かれていないが, tokenMatching メソッドの varMatching メソッド内で 1 単語 (トークン) をマッチングさせるたびに, マッチング成功の成否に合わせ flag 管理をしている.

また, ハッシュマップは実際には 2 次元リストにより構築されているため, 1 文解析がすべて終わり, フラグが立っていない場合の処理だが, 手法 5 の説明時には簡略化のために省略したが, 実際には”ハッシュマップに格納する”のではなく, ”ハッシュマップに格納するためのリスト varslis に登録する”である.

4 実行例

日本語で言うと, [スポーツを趣味にしている人はだれか?] と質問したときの実行結果が以下になる.

```
1 Successfully started
2 検索結果を取得
3 質問を入力してください
4 ?x is-a sports
5 もう 1 つ? 1...Yes/ 0...No
6 1
7 質問を入力してください
8 ?y hobby ?x
9 もう 1 つ? 1...Yes/ 0...No
10 0
11 *** Query ***
12 ?x =is-a=> sports
13 ?y =hobby=> ?x
14 [{?x=baseball, ?y=Taro}]
```

まずはスポーツが何かを求め, その後, それを趣味としている人を探す.

正しい関係性が出力されていることが確認される.

5 考察

手法 1 のやり方では, 1 つの質問しかできない. というのも, 2 つ入力しているが, これは”かつ”の関係で結ばれている 1 つの質問文なのである.

違和感を覚えたのは, 「Taro hobby baseball」という例文. 何となく「太郎の趣味は野球です」になるが, Google 翻訳にかけたら, 「ヒロキホビーサッカー」である. そもそも hobby は動詞になり得ない. 正しくは, 「Taro ' s hobby is baseball」である. この 1 文を扱うのは相当難しい. ただ, 「Taro hobby baseball」が単純に, 「Tail, Label, Head」を表していると考えれば, 何とかなるかもしれない.

6 感想

Java の使い方は, ググってもいいが, 昔しっかり使い込んだ教科書に立ち戻るのも, また一挙である.

参考文献

- [1] Java による知能プログラミング入門 ー著: 新谷 虎松
- [2] 新・明解 Java 入門 ー著: 柴田望洋
- [3] Java 指定型の読み取り ー著: Let's プログラミング
<https://www.javadrive.jp/start/scanner/index2.html>