

# 自然言語を用いたコンピュータとの会話

## —音声認識＋自然言語—

### プロジェクト構成員

三橋 謙太、藤田 真也、岩崎 博子

### 指導教員

松延 拓生（システム工学部）

西村 竜一（システム工学部）

## 【演習の背景・目的】

自然言語処理に興味があり個人的に人工無脳の研究をしてきた岩崎、音響・音声認識に興味を持っていた三橋に、先生から「自然言語処理と音声認識を融合させたアプリケーションを作ってみてはどうか」という声がかかったのが本演習の発端である。そして話を進めていくうちに、ロボットを使えばより面白くなりそうだ、ということで、プログラミングが得意な藤田が加わりこのプロジェクトは本格的に始動した。

プロジェクトでは“コンピュータと人間との自然言語を用いたコミュニケーションを円滑にする”という目標で、コンピュータと会話できるアプリケーションを作ることにした。それは、ユーザがマイクに向かって日本語でしゃべると、応答文が日本語で画面に表示され、同時にロボットが会話内容に応じた動きをする、といったものである。

自然言語処理や音声認識技術はかなり複雑で、3人という少人数、さらに1年という短期間ではとても研究しきれない分野ではないが、コンピュータとのインタラクションを向上させるための手段を研究する、という意味ではこの演習は役に立つと思う。

## 【演習の実施方法】

以下で、具体的にどのように演習を進めたかを示す。効率良く進められるように、いくつかの作業は並行して行われた。

### ①音声認識処理

- ・ Julius か Julian のどちらを使うかを決める

音声認識エンジンはフリーで配布されているものを使った。フリーの音声認識エンジンには Julius と Julian の2種類があり、それぞれの特徴は図1の通りである。

Julius	Julian
<ul style="list-style-type: none"> <li>・ 大量の文章を解析し、統計をとったデータをもとに音声解析（統計言語モデル）</li>   <li>・ 新聞読み上げの認識率は9割以上</li> </ul>	<ul style="list-style-type: none"> <li>・ 認識文法を記述</li>   <li>・ 入力文が限られているとき有効</li>   <li>&lt;長所&gt; 登録した単語は高確率で認識する</li>   <li>&lt;短所&gt; 認識文法記述が面倒</li> </ul>

図1 Julius と Julian の特徴

どちらを使うかを決める作業にはかなりの試行錯誤が必要だった。というのも、今回作るシステムでは入力文のバリエーションが多いので Julius を使うのが良い方法なのだが、実際 Julius を使ってみると認識率が悪かった。そこで Julian を使うことにして、認識文法を記述し始めたのだが、予想以上に骨の折れる作業であったため、やっぱり Julius を使うことに至った。

## ②自然言語処理

### ・プログラム作成(Perl)

自然言語処理プログラムでは文字によるキーワードのパターンマッチングを多用するので、文字列演算が得意な Perl を使うことにした。

### ・コンピュータの応答文のデータを記述

全て手入力で行った。ユーザが発した文に含まれるキーワードから意味を推測して、それに対する応答文を、応答文ファイルに記述する。また、気分によって会話内容を変化させるために、気分ごとに違う応答文を記述する必要があった。さらに自然言語のあいまい性をカバーするため、他の情報（例えば話題、気分の変化値、キーワードが記述されたファイルの場所など）を共に記述した。

### ・動作確認

自然言語処理プログラムの動作確認を行うため、キーボード入力でも動作するようにした。

## ③音声認識処理と自然言語処理の統合

Julius をサーバーモードで起動し、ソケットを使って自然言語処理プログラムと合わせる。最初 Julius がサーバーモードで上手く動かなかったため、西村先生のご協力を頂いた。

#### ④画面の設計

画面設計には Flash を使い、自然言語プログラムから吐き出されたテキスト文と対応させ、Web 表示させることにより、どのパソコンでも表示できるようにした。

#### ⑤ロボットを組み立てる

ロボットはマインドストーム NXT というものを用いたが、その構造上、モーターが3つしか制御できなかった。そこで自分たちで構造を考え、より感情表現ができるようにした。

#### ⑥ロボット制御プログラムを書く

マインドストーム NXT の SDK が REGO から提供されているので、それを利用している。プログラムは複数に切り分け、再利用、拡張しやすくしている。

#### ⑦開発効率を上げるための工夫

- ・ web 上にキーワードと応答文のペアを入力できるページを用意

応答文のデータベース作りを自主演習プロジェクトのメンバーで効率よくできるように PHP で web 上に応答文登録ページを作った。

#### ⑧ロボット制御プログラムとの統合

ソケット通信を使って自然言語プログラムとロボット制御プログラムを統合した。

### 【演習の成果】

- ・実際に動くアプリケーションが出来たこと

場所によっては雑音により音声認識がうまくいかないところもあったが、それ以外では期待通りの動作が得られた。

- ・スケジュール通りに作業を進められたこと

- ・より複雑な自然言語処理アルゴリズムを考えたこと

少しでも自然な会話ができるように、自然言語処理はただのパターンマッチングだけでなく、いくつかの工夫を施した。

- ・音声認識処理についての理解を深められたこと

Julius か Julian のどちらを使うかで試行錯誤したことで、現状の音声認識の問題点を実際に確認できた。

### 【今後の検討課題】

- ・音声認識の認識率を上げる

雑音や音の反響がひどい場所ではほとんど認識できなかったのが、今後どうすれば認識率を上げられるか考えなくてはならない。

- ・応答文の自動生成

今回はコンピュータからの応答文は全て手入力で行っていたが、とても面倒で時間のかかる割にはあまり報われない作業だった。そこで、応答文をコンピュータで自動生成できればこのアプリケーションの質は飛躍的に向上すると思う。

- ・プログラムの書き方について

プログラムが長くなっても後で見えて理解しやすいように、わかりやすく簡潔に書く必要性を痛感した。

## 【感想】

(岩崎)

この演習に参加する以前にも人工無脳（コンピュータと会話できるプログラム）の研究をしていたこともあり、どのようにプログラムを書けばよいかといったアルゴリズムは最初から頭の中で出来ていた。しかしそのアルゴリズムを実際に形にするのは予想以上に大変だった。

機械が人間のように考え、言葉をしゃべれるようになるのは我々の夢であるが、そのためには膨大なデータベースと処理時間が必要なので、今のところ実用化にはほど遠い。しかし効率的なアルゴリズムを発見でき、話題の範囲を絞ればコンピュータにも人間と区別がつかないくらいの流暢な会話ができるようになるかもしれない。それを信じて、これからも自然言語処理の研究を続けていきたい。

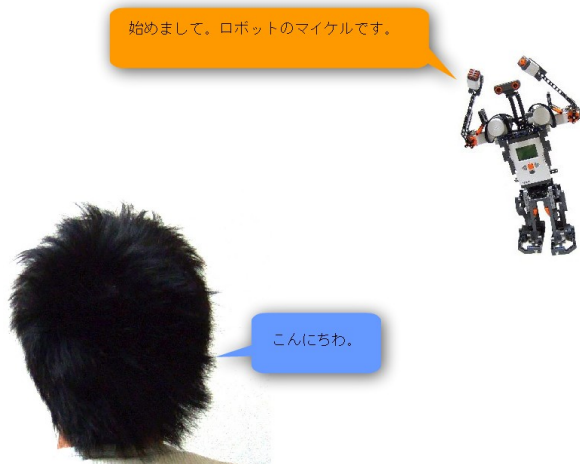
(三橋)

私はまだ2回生ということもあり、授業ではこのような費用のかかる研究もできません。それに加え、1つのことを1年に渡り研究できたことで、卒業研究のイメージがつかめ、プレゼンへの抵抗がなくなり、最先端の技術がどういうものか、またどのような研究がされているのか、といったようなことも理解できました。なにより、チームとして1つの目標に向かって努力し、スキルアップできたことが1年間の成果物です。来年もまた何かチャレンジしたいと考えています。

(藤田)

私は主にロボットの制御を行ったのだが、プログラミングに関しては授業ではまだ基礎的なことしか習っておらず、またNXTはC++という新たな言語を覚えなくてはならなくて、その勉強が大変だった。しかし、この研究を通じて実際にロボットを動かし、また仲間の作ったプログラムと連結して一つの物に仕上げたという経験は、私の中で非常に大きなものであると思う。この経験を生かし、また新たなステップに踏み出したい。

## 【参考写真】



◆音声対話メイン画面



◆中間報告 [展示]



◆作業風景



◆西村先生より指導を受けています