

# リクルート AI

## A3RT を使用した対話型 AI

横田 知紀  
藤原 大喜

### 1. まえがき

私たちは最初、Android のスマホで便利な対話型 AI アプリを制作しようとした。しかし、途中で断念しリクルートの A3RT を利用した AI を制作することにした。

### 2. 原理

- ①AI への問いを記入
- ②送信ボタンを押すと問いが AI のサーバーに送信される。
- ③問いに対して AI が返答し、その返答を受信する。
- ④AI の返答を棒読みちゃんて読み上げる。

といったプログラムである。(図 1)

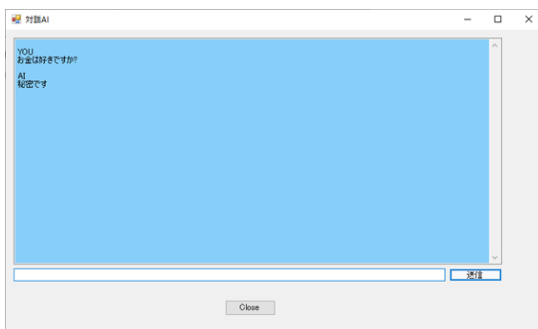


図 1 AI 画面

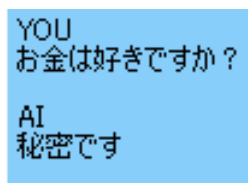


図 2 Q&A

### 3. 研究内容

#### (1) Post and Get

##### • Post の動作

AI への問いを投げかけるプログラムを図 3 に示す。

```
private void button1_Click(object sender, EventArgs e)
{
    textBox2.SelectionStart = textBox2.Text.Length;
    textBox2.ScrollToCaret();

    //文字コードを指定する
    System.Text.Encoding enc = System.Text.Encoding.GetEncoding("UTF-8");
    //POST送信する文字列を作成
    string postData =
        "apikey=DZ51TnUZPqrjEdYjmwj93F6VwJU6nNo&query=" +
        System.Web.HttpUtility.UrlEncode(textBox1.Text, enc);
    //バイト配列に変換
    byte[] postDataBytes = System.Text.Encoding.ASCII.GetBytes(postData);
    //WebRequestの作成
    System.Net.WebRequest req =
        System.Net.WebRequest.Create("https://api.a3rt.recruit.co.jp/talk/v1/smalltalk");
    //メソッドにPOSTを指定
    req.Method = "POST";
    //ContentTypeを application/x-www-form-urlencodedにする
    req.ContentType = "application/x-www-form-urlencoded";
    //POST送信するデータの長さを指定
    req.ContentLength = postDataBytes.Length;
    //データをPOST送信するためのStreamを取得
    System.IO.Stream reqStream = req.GetRequestStream();
    //送信するデータを書き込む
    reqStream.Write(postDataBytes, 0, postDataBytes.Length);
    reqStream.Close();
}
```

図 3 Post プログラム

##### • Get の動作

AI からの返答を受信し、表示するプログラムを図 4 に示す。

```
//サーバーからの応答を受信するためのWebResponseを取得
System.Net.WebResponse res = req.GetResponse();
//応答データを受信するためのStreamを取得
System.IO.Stream resStream = res.GetResponseStream();
//受信して表示
System.IO.StreamReader sr = new System.IO.StreamReader(resStream, enc);
String reply;

reply = Regex.Unescape(getdata(sr.ReadToEnd()));

textBox2.Text += Environment.NewLine + "YOU" + Environment.NewLine + textBox1.Text + Environment.NewLine;
textBox2.Text += Environment.NewLine + "AI" + Environment.NewLine + reply + Environment.NewLine;
textBox2.TextAlign = HorizontalAlignment.Left;

textBox2.SelectionStart = textBox2.Text.Length;
textBox2.Focus();
textBox2.ScrollToCaret();
//閉じる
sr.Close();

textBox1.Text = "";
textBox1.Focus();
```

図 4 Get プログラム

この Post and Get は AI との会話を行うためのプログラムであり、今回の研究の難関だった。

#### (2) 起動・終了プロセス

AI の返答を読み上げる「棒読みちゃん」の起動と終了を行う。

## ・起動

アプリ実行時に同時に棒読みちゃんを起動するプログラムを図5に示す。

```
1 個の参照
public Form1()
{
    InitializeComponent();

    System.Diagnostics.Process p =
        System.Diagnostics.Process.Start
            ("C:/Users/2035/Downloads/BouyomiChan/BouyomiChan.exe");
}
```

図5 起動プロセス

## ・終了

Close ボタンを押した時にプログラムの終了と棒読みちゃんを終了させるプログラムを図6に示す。

```
1 個の参照
private void button2_Click
(object sender, EventArgs e)
{
    Application.Exit();
    System.Diagnostics.Process[] ps =
        System.Diagnostics.Process.
            GetProcessesByName("BouyomiChan");
    foreach (System.Diagnostics.Process p in ps)
    {
        p.Kill();
    }
}
```

図6 終了プロセス

起動・終了のプログラムは2年時の課題研究で作っており、今回の研究でこのプログラム使うことで起動や終了がかなり楽になった。

(3) 棒読みちゃんにAIを喋らせるようにプログラムする。

①専用の場所を作ってvisualstudioでも使えるようにする。

棒読みちゃんに喋らせるためにまず棒読みちゃん専用のclassを作りvisualstudioのFontとプログラムで連動させた。

②AIの喋ったことを棒読みちゃんに声に出させる。

AIの返答をクリップボードに保存し、棒読みちゃんの処理対象文字列にペーストさせるプログラムを作った。(図7)

```
Clipboard.SetDataObject(reply);
```

図7

## 4. まとめ

今回の研究は、初めて試みた事ばかりでなく、自分が経験した事も多く活用して作ることができたと思う。しかし、AIの返答が正確ではなかったため、その点は改善の余地ありとなった。

## 5. あとがき

今回はリクルートのA3RTを使った対話型AIを制作した。プログラムのミスや、棒読みちゃんの読ませ方に少し詰まったが完成形にできてよかったと思う。(横田)

今回の研究は、初の二人がかりでの作業だったが、相方との連携が甘く作業バランスが悪くなってしまった。チームなどでの開発の機会は今後もあると思うので、次の機会ではしっかり計画を練り、バランスのよい分担ができるようにしたい。(藤原)

## 6. 参考にしたサイト

[DOBON.NET](https://www.dobon.net/)