

必要事項を記入し，卒業論文と一緒に提出すること

卒業論文受領証 学生保管

学籍番号	C	0	1	1	2	3	3	6
氏 名	寺田 佳輔							

受領印を受けた後、本票を受け取り 大切に保管してください。	受領印

卒業論文受領証 事務局保管

学籍番号	C	0	1	1	2	3	3	6	受領印
氏 名	寺田 佳輔								
指導教員	田胡 和哉								
論文題目	人工知能ハブの 開発・利用および評価								

(^ o ^) キレイに切り取ってね (^ o ^)

2015
年 度

人工知能ハブの開発と評価

[卒 業 論 文]

人工知能ハブの開発と評価

(指 導 教 員) 田胡 和哉

コンピュータサイエンス学部 田胡研究室

学籍番号 C0112336

寺田 佳輔

[2015 年度]

寺
田
佳
輔

田胡
研究室

東京工科大学

卒業論文

論文題目

人工知能ハブの開発と評価

指導教員

田胡 和哉

提出日

2016年01月18日

提出者

学 部	コンピュータサイエンス 学 部
学籍番号	C0112336
氏 名	寺田 佳輔

2015 年度 卒 業 論 文 概 要

論 文 題 目

人工知能ハブの開発と評価

コンピュータ サイエンス学部	氏 名	寺田 佳輔	指 導 教 員	田胡 和哉
学籍番号 C0112336				

【概 要】

ルいず

目次

第 1 章	現状	1
1.1	近年の機械学習	1
1.1.1	人工無能	1
1.1.2	人工知能	1
1.1.3	neural network	1
1.1.4	DeepLearning	1
1.2	一般的な人工知能開発フレームワーク	1
1.3	一般的な人工知能の返答の流れ	1
第 2 章	提案	2
2.1	開発した人工知能の活用	2
2.1.1	知能の開発をサポートする既存フレームワーク	2
2.2	開発した知能を試す環境	2
2.3	人工知能利用フレームワークの提案	2
2.3.1	全体構成	3
2.3.2	アルゴリズムのみを簡単に追加可能な知能ハブ	3
2.3.3	作成したアルゴリズムを Unity ですぐに試せる機構	3
2.3.4	Unity が利用可能なモーションを追加する機構	3
第 3 章	設計	4
3.1	解析アルゴリズムの追加を可能にする機構	5
3.1.1	解析する内容別にプログラムを保持する機能	5
3.1.2	会話の話題別に解析するアルゴリズムを選ぶ機能	5
3.1.3	解析アルゴリズムを簡単に追加する機能	5
3.2	解析した情報を共有する機能	5
3.2.1	解析情報を保存する機能	5
3.2.2	解析情報を取得する機能	5
3.3	返答アルゴリズムの追加を可能にする機構	5
3.3.1	返答を行うタイミング	5
3.3.2	返答する内容別にアルゴリズムのを保持する機能	5
3.3.3	会話の話題別に返答アルゴリズムを選ぶ機能	5
3.4	作成した知能を Unity で試す機構	5
3.4.1	Unity での出力について	5

3.4.2	Unity との連携に利用する WebSocket	5
3.4.3	Unity への送信フォーマットと作成	5
3.4.4	Unity からの受信フォーマット	5
3.5	アルゴリズムを選定する際に用いる GoogleAPI	5
3.5.1	GoogleAPI について	5
3.5.2	GoogleAPI の有効性	5
第 4 章	実装	6
4.1	開発環境	7
4.1.1	Java の利用	7
4.1.2	Maven フレームワーク	7
4.2	解析部分の実装	7
4.2.1	解析分野別にアルゴリズムを保持する機構	7
4.2.2	会話の話題別に解析するアルゴリズムを選択する機構	7
4.2.3	解析アルゴリズムを 3 行で簡単に追加する機構	7
4.2.4	現在実装している解析アルゴリズム	7
4.3	データベースの実装	7
4.3.1	全ての解析情報を保存する機構	7
4.3.2	解析した情報を取得する機構	7
4.4	返答情報を作成する機構	7
4.4.1	出力分野別にアルゴリズムを保持する機構	7
4.4.2	会話の話題別に出力を作成するアルゴリズムを選択する機構	7
4.4.3	返答アルゴリズムを 3 行で簡単に追加する機構	7
4.4.4	現在実装している出力アルゴリズム	7
4.5	Unity との通信の実装	7
4.5.1	Unity からの入力情報の受信	7
4.5.2	Unity への命令の送信	7
4.6	追加したモーションの利用	7
4.6.1	動作選択アルゴリズムの実装	7
4.7	GoogleAPI と形態素解析を用いた頻出単語表の作成する機構	7
4.7.1	形態素解析による検索ワードの作成	7
4.7.2	GoogleAPI を利用して検索結果を取得	7
4.7.3	検索結果のフィルタリング	7
4.7.4	頻出単語表の作成	7
第 5 章	実行結果	8
5.1	Unity の出力画面の図	8
5.2	実際の会話	8
5.3	アルゴリズムを追加した後の会話	8
第 6 章	結論	9
6.1	結論	9

6.1.1	アルゴリズムの追加による出力の変化	9
6.1.2	Google を用いた会話の話題推定の精度	9
6.1.3	簡単にアルゴリズムを追加できたか	9
	謝辞	10
	参考文献	11

图目次

表目次

第 1 章

現状

1.1 近年の機械学習

1.1.1 人工無能

1.1.2 人工知能

1.1.3 neural network

1.1.4 DeepLearning

1.2 一般的な人工知能開発フレームワーク

1.3 一般的な人工知能の返答の流れ

第 2 章

提案

2.1 開発した人工知能の活用

今回提案するのは先ほど説明した一般的な人工知能フレームワークを用いて開発を行った人工知能を活用するためのフレームワークである。

2.1.1 知能の開発をサポートする既存フレームワーク

既存の人工知能フレームワークは、人工知能自体を作成することをサポートしている。具体例で言うと Chainer は Preferred Networks が開発したニューラルネットワークを実装するためのライブラリであり、実際の開発をすることの手助けをしている。

2.2 開発した知能を試す環境

今回提案するのは、フレームワークを用いて開発したアルゴリズムや、独自のアルゴリズムを考え、作成したプログラムを実際に動かし試す環境である。

通常、人工知能のアルゴリズムを試したいと考えた場合、そのプログラムに対して入力を与える入力の部分とその処理結果を出力する出力の部分を作成する必要がある。

作成した知能の出力結果がただ単に文字で入力して、文字で出力されれば良い場合は、準備するのはほぼ手間が不要であるが、キャラクターとの会話などで試したい場合、非常に入出力の部分を作成するのに時間がかかってしまう。

そこで、今回は作成したアルゴリズムをすぐに試す環境を提供するフレームワークを提案する。

2.3 人工知能利用フレームワークの提案

人工知能利用フレームワークは会話や動作などの返答アルゴリズムを作成した際に、それらの作成したプログラムをフレームワーク上に適当に配置することで、状況や話題に応じて適切な作成した返答アルゴリズムが選択され Unity 上のキャラクターと会話を楽しむことができる、人工知能を利用すること

に焦点を当てたフレームワークである。

2.3.1 全体構成

この人工知能利用フレームワークの全体構成を

2.3.2 アルゴリズムのみを簡単に追加可能な知能ハブ

2.3.3 作成したアルゴリズムを Unity ですぐに試せる機構

2.3.4 Unity が利用可能なモーションを追加する機構

第 3 章

設計

3.1 解析アルゴリズムの追加を可能にする機構

- 3.1.1 解析する内容別にプログラムを保持する機能
- 3.1.2 会話の話題別に解析するアルゴリズムを選ぶ機能
- 3.1.3 解析アルゴリズムを簡単に追加する機能

3.2 解析した情報を共有する機能

- 3.2.1 解析情報を保存する機能
- 3.2.2 解析情報を取得する機能

3.3 返答アルゴリズムの追加を可能にする機構

- 3.3.1 返答を行うタイミング
- 3.3.2 返答する内容別にアルゴリズムのを保持する機能
- 3.3.3 会話の話題別に返答アルゴリズムを選ぶ機能

3.4 作成した知能を Unity で試す機構

- 3.4.1 Unity での出力について
- 3.4.2 Unity との連携に利用する WebSocket
- 3.4.3 Unity への送信フォーマットと作成
- 3.4.4 Unity からの受信フォーマット

3.5 アルゴリズムを選定する際に用いる GoogleAPI

- 3.5.1 GoogleAPI について
- 3.5.2 GoogleAPI の有効性

第 4 章

実装

4.1 開発環境

4.1.1 Java の利用

4.1.2 Maven フレームワーク

4.2 解析部分の実装

4.2.1 解析分野別にアルゴリズムを保持する機構

4.2.2 会話の話題別に解析するアルゴリズムを選択する機構

4.2.3 解析アルゴリズムを 3 行で簡単に追加する機構

4.2.4 現在実装している解析アルゴリズム

4.3 データベースの実装

4.3.1 全ての解析情報を保存する機構

4.3.2 解析した情報を取得する機構

4.4 返答情報を作成する機構

4.4.1 出力分野別にアルゴリズムを保持する機構

4.4.2 会話の話題別に出力を作成するアルゴリズムを選択する機構

4.4.3 返答アルゴリズムを 3 行で簡単に追加する機構

4.4.4 現在実装している出力アルゴリズム

4.5 Unity との通信の実装

4.5.1 Unity からの入力情報の受信

4.5.2 Unity への命令の送信

4.6 追加したモーションの利用 - 7 -

4.6.1 動作選択アルゴリズムの実装

4.7 GoogleAPI と形態素解析を用いた頻出単語表の作成する機構

第 5 章

実行結果

5.1 Unity の出力画面の図

5.2 実際の会話

5.3 アルゴリズムを追加した後の会話

第 6 章

結論

6.1 結論

6.1.1 アルゴリズムの追加による出力の変化

6.1.2 Google を用いた会話の話題推定の精度

6.1.3 簡単にアルゴリズムを追加できたか

謝辞

何か色々と感謝する。

参考文献

- [1] ゼロの使い魔制作委員会：“ゼロの使い魔公式ウェブサイト” <http://www.zero-tsukaima.com/zero/index.html> (2012/12/28) .
- [2] 奥村晴彦 著：“ \LaTeX 2 ϵ 美文書作成入門 改訂第 3 版” (技術評論社 2004, 403pp) .