

My Book

published by ReVIEW

hsm_ai と学ぶはじめての人工知能

hsm_hx 著

2018-11-10 版 発行

第 1 章

hsm_ai とは ~はじめての人工知能~

1.1 この本について

hsm_ai は、プログラミング言語 Ruby により制作された人工知能^{*1}です。
この本では、hsm_ai を制作するにあたって用いた以下の技術について取り上げます。

- 形態素解析
- マルコフ連鎖
- 係り受け解析
- 特徴語抽出
- 感情極性辞書による感情解析
- VRoidStudio による 3D モデル製作
- ゲームエンジン Unity

1.2 hsm_ai とは

hsm_ai は、開発者である私、hsm_hx の発言を学習し文章を自動生成するマルコフ連鎖による bot です。

私の Twitter^{*2}からツイートを取得し、形態素解析という手法を用いてそのツイートを単語ごとに分割します。単語ごとに分割した文章を数単語ずつの塊にし、そうしてできたたくさんの単語の塊を規則に従って組み替えることで日本語らしい文章を生成します (詳しくは第 1 章にてお話しします)。

百聞は一見にしかずとも言うので、まずは hsm_ai の生成した文章をいくつかご紹介します。以下に示す文章は、全て hsm_ai というシステムによって生成されたものです。

ひえ〜8 時間で自動的に目を覚ました

フォロワー 37 人もいるの人間になる

ミーン！（閃いた顔）

^{*1} 正確には人工知能と呼べるものではありませんが、それについては第 n 章にて詳しく記述します。

^{*2} @hsm_hx: https://twitter.com/hsm_hx

ミーン！ で爆笑してるでしょ

頭を使うことを学んだ

ああ ^ ~単位落ちる

声出して笑ってもらえるって嬉しいかもしれない

iTunes で素敵に心を購入

お前はやりたいことやるのが大事

いかがでしょうか？ 確かになんとなく不自然に見える文章もありますが、かなりの割合で日本語として解読が可能な文章が生成されていると思います。それどころか、人間には書けないような独創的（という表現が正しいのかはわかりませんが）な文も散見されます。

この本では、この hsm_ai が生み出され対話型 bot として高専祭で展示されるまでの成長の過程をひとつずつ追っていきます。

1.3 開発環境

hsm_ai を開発するにあたって利用した言語やライブラリ、ツールは以下の通りです。

Windows 10

言わずと知れた一般的 OS。Unity と VRoidStudio を利用するために使いました。

Antergos

Windows, MacOSX と並ぶ OS である Linux の一種。Windows よりプログラミングの環境構築が簡単（個人差あり）なのでメイン開発 OS として採用。Arch Linux というディストリビューションを使いやすくするため予め必要なパッケージを揃えたもの。Arch Linux は最高です。

Ruby 2.5.1

日本人によって開発されたスクリプト言語。web サービスを作るのによく使われています。

MeCab 0.996

日本語の文章を形態素解析（第 2 章にて解説）するためのエンジン。

CaboCha 0.69

日本語の文章を係り受け解析（第 4 章にて解説）するためのエンジン。

Unity

言わずとしれた超有名ゲームエンジン。3D モデルを動かすために使用します。

VRoidStudio

最近流行りの 3D モデル作成ソフト。絵を描くように直感的な操作で本格的な 3D モデルが作れます。

第2章

マルコフ連鎖による文章生成

2.1 マルコフ連鎖とは

マルコフ連鎖とは、物理や統計、強化学習など様々な分野において事象をモデル化するためにしばしば用いられる確率論の考え方の一種です。

この本ではマルコフ連鎖について詳しく踏み込むことはしませんが、ざっくりとその性質を説明すると、ある事象についてその未来を考えると、「その事象の未来は過去に関わりなく、現在の状態のみによって定まる」という特性を持つときの未来予測に用いられるアルゴリズムです。

この後で具体的に例を挙げて解説しますが、hsm_ai ではこのような理論を応用することで日本語らしい文章を機械生成しています。

2.2 マルコフ連鎖による文章生成アルゴリズム

では、実際にマルコフ連鎖を用いて文章を生成していきます。hsm_ai の文章生成には、以下のようなアルゴリズムを採用しています。実際に例を挙げながら、機械的に文章が生成される過程を追っていきましょう。

まず、学習元になる複数の文章を用意します。ここでは、例として以下の2つの文を用意しました。

ここにりんごがあります。その箱にはぶどうが2つ入っています。

この文章をマルコフ連鎖するために、まずは下準備として、それぞれの文を単語ごとに分割します。これを分かち書きといいます。上の文を分かち書きすると、下のようになります。

ここ / に / りんご / が / あり / ます / 。その / 箱 / に / は / ぶどう / が / 2 / つ / 入っ / て / い / ます / 。

さて、次に、この分かち書きされた文章から、3つの連続する単語をひとまとめたブロックを作ります。ここでポイントになるのは、各文章のはじめとおわりにそれぞれ「ここが文のはじまり(おわり)です」という目印をつけることです。実際にブロックを作ってみます。ここでは、はじめとおわりを表す印として「*」という記号を使うことにします。

[*, ここ, に], [ここ, に, りんご], [に, りんご, が], [りんご, が, あり], [が, あり, ます], [あり, ます, 。], [ます, 。, *] [*, その, 箱], [その, 箱, に], [箱, に, は], [に, は, ぶどう], [は, ぶどう, が], [ぶどう, が, 2], [が, 2, つ], [2, つ, 入っ], [つ, 入っ, て], [入っ, て, い], [て, い, ます], [い, ます, 。], [ます, 。, *]

*)

[]の中に、3つの連続する単語をコンマ区切りで並べています。hsm_aiの文章生成には、プログラムにより生成された何千、何万もの単語ブロックが使われています。

さて、このたくさんのブロックをどのように使うのかというと、ここでマルコフ連鎖というものを使います。条件に合うブロックを探し、その中からランダムに1つを選び後ろにつなげていくことで文章を生成していきます。具体的に上のブロックを使ってマルコフ連鎖によって文章が生成される過程を追ってみましょう。

まず、文章のはじめは「*」としてありました。なので、「*」からはじまるブロックを探します。[* , ここ, に], [* , その, 箱]の2種類です。

この3つの中から、ランダムに1つを選びます。今回は[* , ここ, に]を選んだとします。

次に、[* , ここ, に]につながるブロックを探します。つまり、「に」からはじまるブロックを選べば良いというわけです。ここでは、[に, りんご, が], [に, は, ぶどう]の2つが考えられます。

この2つの中から、ランダムに1つを選びます。次は[に, は, ぶどう]が選ばれたとします。

こうして、[* , ここ, に], [に, は, ぶどう]という2つの接続可能なブロックが選ばれました。これを「*」で終わるブロックに到達するまで続けます。その様子を表したものが下の図です。

このように、「ここにりんごがあります。」「その箱にはぶどうが2つ入っています。」という2つの文から新しく、「ここにはぶどうが2つ入っています。」という意味の違う文章が生成されました。これが、マルコフ連鎖による文章生成です。

2.3 形態素解析エンジン MeCab

さて、マルコフ連鎖による文章生成アルゴリズムについてはなんとなくイメージを掴んでもらえたかと思います。しかし、先ほど登場した「分かち書き」という処理をプログラミングで実現するにはどうすればいいのでしょうか？3つの単語をブロックにして連鎖させる…といった箇所については、ある程度プログラミングの経験がある人であれば愚直にコードに起こすことができるでしょう。ですが、ある文章を単語ごとに分割してその品詞を特定する、という処理はどうやって書けばいいのでしょうか？おそらく、それを実現するには膨大な時間と研究が必要です。

そこで役に立つのが形態素解析エンジン MeCab^{*1}です。

MeCabは、京都大学とNTT株式会社の共同研究プロジェクトによって開発された形態素解析エンジンです。形態素解析というのは、日本語や英語など、私達が普段から使う言語（自然言語）の文を単語に分割し、その品詞などを判別する解析作業のことを指します。

MeCabを使用することで、文章を簡単に形態素解析し分かち書きされた状態にすることが出来ます。MeCabはGitHub上で公開されているオープンソースソフトウェアなので、Gitが導入されている環境であればリポジトリ^{*2}をcloneしビルドすることですぐ使えるようになります。

```
$ git clone https://github.com/taku910/mecab.git
$ cd mecab/mecab
```

^{*1} <http://taku910.github.io/mecab/>

^{*2} <https://github.com/taku910/mecab>

```
$ ./configure --with-charset=utf8
# make install
$ cd ../mecab-ipadic
$ ./configure --with-charset=utf8
# make install
```

以上の手順で、MeCab 本体と MeCab を動かすための辞書データをインストールします。また、必要に応じて、新語やネット用語などに特化した辞書データである mecab-ipadic-neologd^{*3}も追加で導入します。

ここまでできたら、MeCab を実際に動かしてみます。コマンド上で ‘mecab’ コマンドを実行し、続けて好きな文章を入力することで動作を確認することができます。

また、MeCab は各種プログラミング言語からスムーズに使用するためのバインディングを標準で提供しています (Perl, Ruby, Python, Java, C#)。その他、MeCab をより快適に利用するためのライブラリも各種言語で充実しています。例えば、Ruby では natto という Gem が配布されており、hsm_ai は natto を採用しています。

2.4 Twitter の bot としてリリースする

さて、理論を抑えたところで、実際に Twitter からツイートの情報を取得し、そのデータを元に生成した文を Twitter に投稿してみます。

Twitter からデータを取得したり、ツイートやいいねなどをプログラムから行うには、Twitter が公式に提供している API を使います。API とは Application Program Interface の略で、プログラムから何らかのアプリケーションを利用するための決まった形式のことを指します。具体的には、今回使う TwitterAPI の他にも Google の提供する Google Maps API や Microsoft の提供する Face API など、様々なものがあります。このような API を利用することで、プログラミングを始めたての初心者でも既存のサービスの機能をプログラムから利用したり、顔認識や機械学習などの複雑な処理を自分で実装することなく自分のプログラムに組み込むことができます。

今回はツイートの取得、自動ツイートを実現したいので、TwitterAPI を使います。TwitterAPI の利用には Twitter 開発者登録が必要です。Twitter のデベロッパー向けページにアクセスし、開発者として認証してもらうために数点の質問に答えます。開発者として認証されると、プログラムから TwitterAPI を利用するために必要なトークンが発行されるので、それを使って TwitterAPI を利用するプログラムを書きます。

TwitterAPI の利用の仕方については、インターネットで検索するとたくさんヒットするのでここでは割愛しますが、多くのプログラミング言語では TwitterAPI を簡単に利用するためのライブラリが公開されています。

このようにして作られたのが、現在 Twitter 上で動作している hsm_ai です。ソースコードは GitHub^{*4}上で公開しているので、興味がある人は参考にとしてみると良いかもしれません。

^{*3} <https://github.com/neologd/mecab-ipadic-neologd>

^{*4} https://github.com/hsm-hx/hsm_ai

第 3 章

対話機能を実装する

この章では、受け取った文章から話題となるキーワードを見つけそのキーワードに沿った文章を返す機能を実装していきます。使用する技術は以下の通りです。

3.1 形態素解析による話題検出

今回は、形態素解析によって与えられた文章から話題であると推測される単語を検出する簡易的な手法を紹介します。

まず、文章の中で話題になり得る要素が何であるかを考えます。以下の文章を例にとってみましょう。

このオレンジジュースは美味しいです。

さて、この文に対する返答として期待されるものは一体何でしょうか？ 考えられるのは「そうですね」といった相槌、もしくは「私はオレンジジュースはあまり好きではありません」「駅前にもっと美味しいオレンジジュースが売ってあるよ」という風なオレンジジュースについての言及、「この店はぶどうジュースも美味しいですよ」というジュースについての言及などが考えられるでしょうか。

ここでは、会話に対する返答として考えられるものは大きく分けて 2 種類あるものと考えます。1 つは相槌、もう 1 つは会話中の名詞に関連する文です。

相槌については定型文を返せば良いとして、ここでは後者の「会話中の名詞に関連する文」を返す処理を作っていきます。

考え方は単純で、会話文として受け取った文データから名詞を抽出した上で不要なものを弾いていきます。文から名詞を抽出するという部分については、MeCab による形態素解析を用いることで簡単に実装することが可能です。

MeCab に文を入力することで、品詞 ID と呼ばれるものを得ることができます。基本的には、この品詞 ID が名詞のものである単語のみを抜き出すことで文中の話題を検出していきます。

さて、ここで実際に品詞 ID によって名詞の抽出を行ってみます。以下の文章を例に試してみましょう。

ぼくは君に借りた本を読んだ。

この文の場合、品詞 ID によって名詞を抽出すると次の 3 単語を得ることができます。

ぼく、君、本

さて、ここでこの結果が妥当であるか考えてみます。確かに名詞を抽出していることには間違いはないのですが、「ぼくは君に借りた本を読んだ」と言われたときに「ぼく」や「君」についての話題で返すのは本当に正しい会話でしょうか？

この会話の場合、より話題として適切なのは「本」でしょう。なぜ「ぼく」や「君」は不適なのかを考えます。今回は、その理由を「話し手を指す言葉であるため」と仮定します。この「ぼく」や「君」のように会話に登場する人間を指す言葉は、直接的な話題になりにくい傾向があると考えたためです（私は専門家ではないため、本来この考え方は正しくないかもしれませんが）。

このように話し手を指す言葉を抽出対象から外すため、今回は簡易的な措置になりますが、品詞が名詞であると推定された単語の中でも、代名詞の品詞 ID を持つ単語を除外するように指定します。こうして、与えられた文から話題となりそうなものを抽出することができるようになりました。話題と推定される単語が2つ以上抽出された場合はランダムにどれか一つを選択することで単純なものではありますが話題検出のコードとしています。

3.2 話題に対する応答を作る

会話文として与えられたデータに対して話題を検出できるようになったところで、それに対してどのように応答をするかという部分を作っていきます。ここで話題がマルコフ連鎖による文章生成に戻ります。会話の話題と推定されるものを単語で得ることができているため、その単語についての文を生成して返却すれば良いでしょう。

これは単純で、話題として抽出した単語を含むブロックを探してきて、その中からランダムにひとつを選び、先頭と末尾が「*」になるまで前後にマルコフ連鎖をさせれば良いわけです。

このような手法で話題に対するレスポンスを作ることで、将来的に話題として「きれいな景色」や「おいしい水」、「オレンジジュース」などの修飾された名詞や MeCab 標準だと2つの異なる名詞の連続と認識されてしまうようなものを抽出するようにした際にもそのように単語が連続しているブロックを探すと良いので、比較的容易に会話システムをアップデートできるでしょう。

第 4 章

3D モデルと音声の実装

4.1 VRoidStudio による 3D モデル製作

VRoidStudio^{*1}はピクシブ株式会社によって開発された 3D モデリングソフトウェアで、人の形をした 3D アバターを簡単に作成することができます。Windows と Mac に対応しており、無償で誰でも使うことができます。

このソフトウェアを使うことで、絵を描くように 3D モデルを作ったりそのモデルに表情を付け実際に動かしたりすることができます。作った 3D モデルは VRM という形式で書き出すことができるため、vrm に対応した各種ソフトウェア・サービスにて使用することができます。

今回は、この VRoidStudio を使って hsm_ai のイメージキャラクターを作成し、Unity（ゲームエンジン）上で動かせるようにします。

VRoidStudio は公式サイトからダウンロードして利用することができます。頻繁に機能の追加や改良がされた最新版が公開されるので、メールニュースや公式 Twitter で最新の情報を追うと良いでしょう。

ここでは、Windows 版を例として hsm_ai のモデルが作られるまでを追っていきます。ダウンロードしてファイルを解凍したら、フォルダの中の VRoidStudio.exe を開きます。これが VRoidStudio の本体です。VRoidStudio を起動すると、解像度や描画性能の設定画面が開くので好みに設定して Play! ボタンをクリックします。

表示されたメニュー画面で新規作成を選び、作成する 3D モデルの名前を入力しモデル作成を始めます。VRoidStudio では、元あるモデルから顔、体型、髪型を編集することができます。顔と体型は目の形、口など各種パーツの位置、身長など様々な項目をパラメータを 0~100 の間で操作することで調整します。髪型は、元ある素体に対してマウスの D&D で房を生やすことで実現します。ここでは、実際に hsm_ai の髪型を再現する過程の一部を紹介します。

髪型編集タブを開きます。視点が頭に近づき、パネルの配置が髪型編集モードに変わります。ここで、左パネルから『手書きガイドを追加』をクリックします。すると、頭の周りに網状の表示が出現します。マウスを用いて髪の手書きガイドを描画すると、この手書きガイドに沿った形になります。手書きガイド上の頂点を D&D で動かすと、髪を描画するときの流れ方を変えることができます。今回は毛先を少し内向きにしたいので、以下のような形に手書きガイドを修正しました。

^{*1} <https://vroid.pixiv.net>

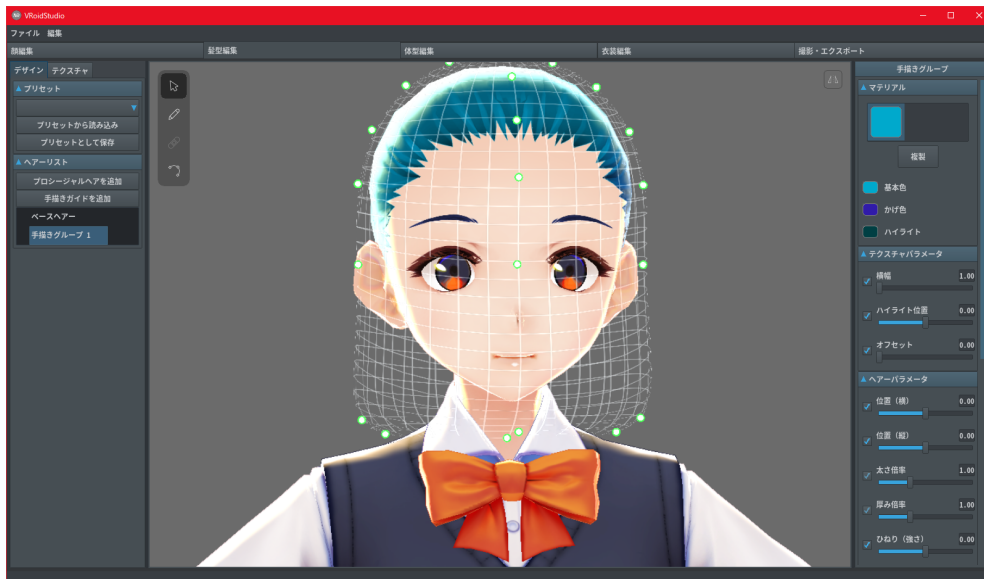


図: 手書きガイドの修正

さて、それでは髪を描画していきましょう。マウスでも良いですが、ペンタブレットのような直感的に入力することができるデバイスがあるとより快適です。中央パネルの左上にあるアイコンのうち、上から二番目のペンアイコンをクリックすることで描画モードにすることができます。まず、髪の房の形を決めます。デフォルトでは毛先に行くにつれて房が細くなっていきますが、hsm_aiの髪は毛先を四角に近い形（ぱつん）にしたいので、そのように設定を変更します。右パネルを下側にスクロールすると、『形状』という項目があるので、『ぱつん』を選択します。すると、その下のグラフの形が変わったことがわかります。このグラフを直接編集することでより細かく形状の指定ができますが、今回はこのプリセットのまま進めます。また、パネル上部の『ヘアーパラメータ』からも髪の形状を細かく指定することができます。

まずは前髪から作っていきます。つむじ付近から前髪の長さの位置まで線を書く感覚で D&D します。このとき、カメラを前髪の斜め上付近にするとうまく描くことができます。前髪を生やし終わったら、長さを調整します。中央パネル左上のアイコンのうち、一番下のワイヤアイコンをクリックします。このモードでは、描画した房の形を直接編集することができます。編集したい房をクリックし、頂点を好きな位置に移動させて前髪の長さを揃えます。

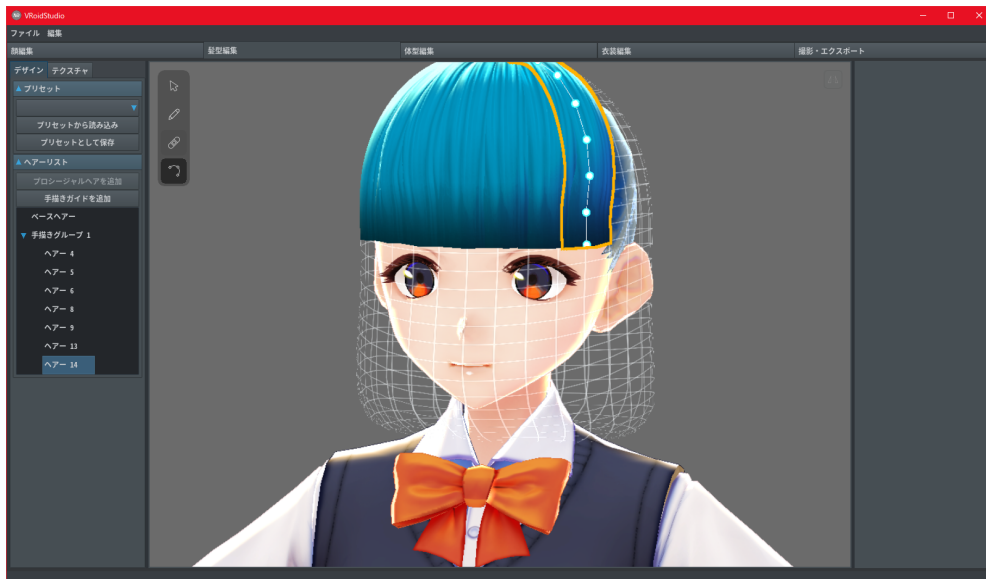


図: 前髪モデリング

また、描画した房を削除することもできます。中央パネル左上のアイコンのうち、マウスカーソルをクリックします。この状態で房をクリックすると、左パネルのヘアーリストから該当するオブジェクトが選択された状態になるので、右クリックして削除します。今回は、より自然な前髪に見えるよう数点の房を削除し新しく『形状：ふんわり』で房を置き換えました。

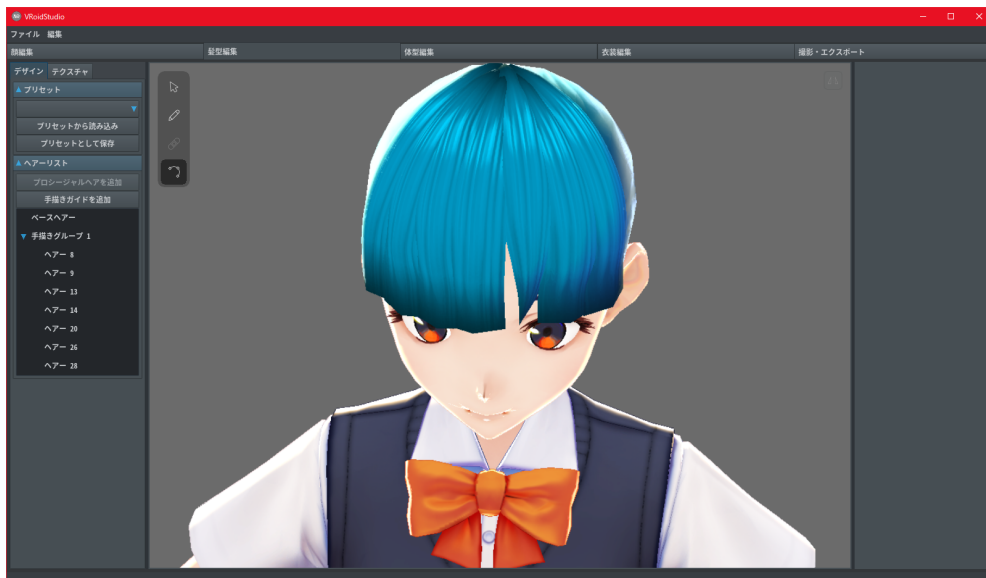


図: 前髪の修正

横髪や後ろ髪も同じように描画・修正を繰り返してイメージに合う形を作っていきます。髪型が完成したら、色を変更します。選択モードにした上で、左パネルのヘアーリストから『手描きグルー

ブ 1』と表示されている箇所をクリックすると右パネルにマテリアルタブが表示されます。この中の『基本色』『かけ色』『ハイライト』を変更することで、髪を任意の色に変更することができます。

また、髪や顔のテクスチャは手動で描き換えることも可能です。顔編集・髪型編集タブを開くと左パネルにテクスチャタブがあるので、その中から任意の部位のテクスチャを描き替えることができます。最新版ではレイヤ分割に対応しているので、まさに絵を描くように直感的にテクスチャを編集することができます。

以下の図は以上の機能を使用して作られた hsm_ai の 3D モデルです。

このモデルを Unity で扱えるよう VRm 形式でエクスポートします。『撮影・エクスポート』タブから、左パネル下部のエクスポートをクリックして必要な箇所を入力しエクスポートします。こうして、VRoidStudio による 3D モデルの準備が完了しました。

4.2 Unity で VRM モデルを動かす

hsm_ai と学ぶはじめての人工知能

2018 年 11 月 10 日 初版第 1 刷 発行

著 者 hsm_hx
