

# 人間的な処理に基づくウェブ操作モデルの検討

## Consideration of Web Navigating Model Based on Human-like Processing

櫛翔佑 \*1

Shosuke Haji

山本翔太 \*2

Shota Yamamoto

中島光人 \*3

Raito Nakajima

萩原康弘 \*4

Yasuhiro Hagiwara

永原恒治 \*5

Koji Nagahara

\*1明治大学大学院

Meiji University Graduate School

\*2早稲田大学

Waseda University

\*3東京大学

The University of Tokyo

\*4長崎県立大学大学院

University of Nagasaki Graduate School

\*5放送大学

The Open University of Japan

We browse web pages and operate screens as a matter of course, but even within a single site, we perform complicated processes that require multiple steps, such as text input and clicking. For this reason, it is important to develop an agent that automatically operates the screen using natural language as a query to improve convenience. However, the design of web pages differs greatly from site to site, and there are discontinuous screen transitions caused by clicks and other actions, making learning difficult with existing methods. For this reason, no practical web manipulation agent has been developed to date. Therefore, we considered a web manipulation agent based on human-like processing. Specifically, we trained web operations using a world model. Conclusively, we failed in training the world model well and got no good results. In this paper, we summarized our attempts on the web manipulation agent and considered the results.

## 1. はじめに

我々は当たり前のようにウェブページを閲覧し画面操作しているが、1つのサイトの中でも、テキスト入力やクリックなど、複数のステップを必要とする煩雑な処理を行なっている。そのために、自然言語をクエリとして、自動で画面操作するエージェントの開発は、利便性の向上に重要である。しかし、ウェブページは、サイトごとにデザインが大きく異なり、クリック等による非連続的な画面遷移があり、既存手法では学習も難しい。そのために、現時点で実用的なウェブ操作エージェントは登場していない。そこで、我々は人間的な処理を元にしたウェブ操作エージェントを検討する。具体的には、ウェブ操作を世界モデルを学習させた。その結果としては、うまく学習させることができず、良い結果は得られなかった。

本論文では、今回行った取り組みをまとめ、考察を行う。

## 2. 関連研究

### 2.1 World of bits

World of bits(WoB)[Shi et al. 2017] は、Shi などによって導入された、強化学習の研究を促進するために、エージェントが低レベルのキーボードとマウスのアクションを実行することでインターネット上のタスクを完了するプラットフォームである。

### 2.2 世界モデル

World Models[Ha et al. 2018] は Ha などによる提案で、強化学習環境の生成ニューラルネットワークモデルの構築を探る。環境の圧縮された空間的及び時間的表現を学習するために、教師なしで迅速に訓練を可能とする。

Dreamer[Hafner et al. 2020] は Hafner らによる提案。学習済み世界モデルは、エージェントの経験を要約して、複雑な動作の学習を容易にする。次元の感覚から世界モデルを学習することは、深層学習によって実現可能になりつつあるが、そこから動を導き出す法は数多くある。Dreamer は、潜在的な想像

だけで画像から期的なタスクを解決する強化学習エージェントである。学習された世界モデルのコンパクトな状態空間で想像された軌跡を介して、学習された状態値の分析的勾配を伝播することにより、動作を効率的に学習する。20の挑戦的なビジュアルコントロールタスクで、Dreamer はデータ効率、計算時間、および最終的なパフォーマンスにおいて既存のアプローチを超える。

## 3. 取り組んだこと・経緯

### 3.1 タスク設計

我々の目的は、複数の煩雑なステップを必要とするウェブ操作において、世界モデルを使うことの有効性を示すことである。そこで、本研究では、ウェブ操作の強化学習の環境において、世界モデルと他のモデルの学習を行い、世界モデルを用いることによる精度向上を目指した。

### 3.2 環境の選定・構築

強化学習を行うにあたり、実環境でエージェントにおいて学習させるわけにはいかない。先行研究においては、WoBがウェブ操作の強化学習のための環境として用意されている。その中でもMiniWob++は、Wobの中でも比較的容易なベンチマークタスクとして用意されたものである。しかし、Python2やgymのバージョンが古く実装コストが大きすぎるため、実装上の課題からMiniWob++の環境構築はできなかった。その中で、Flight-Bookingのみであれば、上記MiniWob++よりは環境構築のコストが低い実装があった。そのために、今回はFlight-Bookingに取り組んでいる。実験で、Flight-Bookingの学習がうまく行かなかったときに、他タスクでの実験ができなかったのは、このような経緯によるものである。また、この環境構築はTAの方の協力をいただきながらも、かなりの時間を必要としたために、次節以降は限られた時間での作業となっている。

### 3.3 サーベイ

実験にあたり、学習のアイデアを集めるための論文調査を行った。具体的には、World of bits(WoB)[Shi et al. 2017]を引用した論文の概要や手法を調査した。調査の結果、既存のウェブ操作モデルにおいては、世界モデルを用いたアプローチがないことを確認した。

### 3.4 モデル実装

調査の結果、先行研究ではウェブ操作で世界モデルが使われていないことから、本研究では、世界モデルを用いてベースラインより良い精度と分析が得られれば、手法の有効性を主張できると判断した。そこで、今回は世界モデルとして Dreamer のみを実装した。また、今回利用した Flight-Booking の実装では、予め Actor-Critic Agent が実装されていたため、これをベースラインとして利用した。

## 4. 実験

### 4.1 実験設定

ウェブ操作の環境として Flight-Booking[Flight-Booking-tf]を利用した。この環境のスクリーンショットを図1に示す。タ

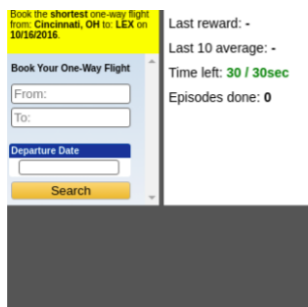


図 1: Flight-Booking 環境。

スクは、飛行機の予約サイトにおいて、出発地と到着地、日付を決めた上で、最短の飛行機を予約するものである。エージェントは、条件を入力して検索した上で、条件にあった便を探して予約する必要がある。今回は、タスクの難易度を下げするために、タスクを固定しており、場所や日時は固定された状態で学習を行った。

一方、エージェントは、各ステップごとに、画面の画像を入力として、座標 (x,y) とテキスト 1 文字を出力する。それに対して、モデルは、出力された座標にテキストを入力を行う。テキストは大文字アルファベット 'A'-'Z'、スペース ' ' および、空文字 '' であり、特にクリックだけを行いたい場合は、エージェントはテキストとして空文字を出力すればよい。

エージェントは、タスクを最後まで正しく完了できた場合には正の報酬、30 秒以内に完了できなかった場合は負の報酬-1 が与えられる。

ベースラインモデルとして Actor-Critic、世界モデルとして Dreamer を用いた。Actor-Critic は 384 エピソード、Dreamer は 737 エピソード行った。エピソード数が、統一されていないのは、課題の締切や omniscampus の制約等によるものであるが、次節で示す通り、今回はこれは問題とならない。

### 4.2 結果・考察

Actor-Critic、Dreamer とともに、学習中に一度もタスクを成功させることができなかった、そのため、両モデルの全エピ

ソードでの報酬は-1 のみで、学習できなかった。

学習できなかった原因として、報酬が極端にスパースであることが考えられる。今回の Flight-Booking を人手で最短で操作した場合にも、4 回の大きな画面遷移を伴う 30 ステップを必要とした。30 ステップ中には、文字を入力したり、日付指定のためにカレンダー上の正しい日付をクリックする必要がある、これらの操作を 1 つ誤ると、タスク完了に辿り着くのは人手でも困難であった。

## 5. 議論

今回の課題の中では、うまく学習させることができず、成果を出すことができなかった。ここでは、今回のタスクを実現するための、先行研究や今回行った実験について述べる。

### 5.1 実現のために

実験後、改めて先行研究の調査を行った。その結果、今回行った Flight-Booking は、通常の強化学習では先行研究でも、全く学習できておらず、成功率が 0%であることが明らかになった。

一方で、Flight-Booking においても、タスクを成功できている先行研究もあったため、それらを紹介する。

まず、QWeb[Gur et al. 2018]においては、DOM を入力とし、タスクを細かく分解し、カリキュラム学習を行っている。QWeb においては、報酬設計などを工夫することで、Flight-Booking の成功率 100%を達成している。

CC-Net[Peter et al. 2022]においても、DOM と画像を入力としており、BERT などの事前学習済みモデルを利用することで、チャレンジングな Flight-Booking においてもエージェントがタスクが成功できていることを主張している。

### 5.2 追加実験

今回の実験では、うまく学習できなかったが、今後の可能性を模索するために追加実験を行った。前節の内容を踏まえ、学習済みモデルを活用したアプローチとして検討した。

#### 5.2.1 実験設定

この実験での環境を図2に示す。実装や報酬設計などの容易さから、一般的なベンチマーク MiniWob++等とは異なる独自の環境を構築している。物件検索サイト REDFIN において、都市、予算、ベッド数を自然言語で指定し、該当する物件をクリックするタスクである。ただし、予算とベッド数の指定は任意で、指定されない場合もある。

#### 5.2.2 実験 1

Flight-Booking のタスクでは、座標 (X,Y) の数値を出力していたが、画像から正確な座標値を生成するのは容易ではない。そこで、表示された web ページ内から HTML の Input タグと Button タグの画像を取得した。これにより、「クリックする」というアクションをするときに、その座標を求めるのではなく、クリックすべきボタンの画像を予測するタスクとした。ここでは、自然文のコマンドと画像を対照学習することで、コマンドから操作すべき画像・HTML 要素を予測する学習を行った。OpenAI の CLIP を Fine-tuning することで、実現している。実験結果を図3に示す。入力のコマンドに対して、適切な画像が選択できていることが確認された。

#### 5.2.3 実験 2

実験 1 で、コマンドが生成できれば、操作すべき要素が選択できるようになった。そこで、タスクの自然文から、コマンドの生成を行うモデルの実験を行った。モデル図と結果を、図4に示す。モデルは、タスクを DistilBERT で潜在表現に変換

