

# WebNavix : Domain-wise Mixture-of-Experts による 継続汎用 Web Navigation エージェント

塩畑 晴人<sup>†</sup> 柴沼 巖<sup>†</sup> 池田 耕<sup>†</sup>

<sup>†</sup>茨城工業高等専門学校

## 1 はじめに

Web Navigation エージェントは、ウェブページの状態とユーザーの指示を入力とし、意思決定エンジンを経て、ウェブページへの操作を出力するシステムである。[1] 最近の Web Navigation に関する研究は、意思決定エンジンとしての大規模言語モデル (LLM) の In-context Learning の工夫や、フレームワーク全体の改善にフォーカスされることが多い。一方、十分に探求されていない課題として、エージェントの未知の環境への汎化と、継続学習の効率化が挙げられる。既存の Web Navigation のためのデータセットは複数のドメインを含んでいるが、これに対して密な LLM がすべてのドメインで高い汎化性能を獲得するのは困難である。また、実世界では、機能の刷新や全く新しいドメインの登場などにより、理想的なデータセットは短期間で更新されるため、エージェントはその変化を継続的に学習する必要がある。しかし、巨大な LLM のフルパラメータ Fine-tuning には大規模な計算環境が必要であり、一度の継続学習でも莫大なコストが発生する。本論文では、これらの課題を解決するための手法として、データセットをドメインごとに分割し、それぞれのドメインに対して個別に Fine-tuning された LLM をドメイン専門家として継続的に統合する **WebNavix** を提案する。

## 2 WebNavix

WebNavix は、Mixture-of-Experts (MoE) [2] に基づくドメイン別モデルマージ手法である Branch-Train-MiX (BTX) [3] を Web Navigation のために拡張する。WebNavix は、Task Adapt、Branch、Domain Adapt、MiX の 4 つのステージから構成される。

### 2.1 Task Adapt

Task Adapt では、シード LLM  $\mathcal{M}$  を Web Navigation に適応させる。Web Navigation のためのデータセット  $\mathcal{D}$  が与えられたとき、Task Adapt では一度  $\mathcal{D}$  の全体で  $\mathcal{M}$  をフルパラメータ Fine-tuning する。これにより、 $\mathcal{M}$  内の Feed Forward Network (FFN)、Self-attention、その他すべてのサブレイヤーを Web Navigation に適応させる。ただし、後のステージでの学習の安定性を考慮して、 $\mathcal{M}$  は過学習を避けるために少量のエポック数で学習される。

### 2.2 Branch

Branch では、 $\mathcal{D} = \{D_i\}_{i=1}^N$  により  $\mathcal{D}$  が含むドメイン数  $N$  で  $\mathcal{D}$  を分割し、 $\mathcal{M}$  によって初期化された  $N$  個の Expert Language Model (ELM)  $\mathcal{M}_i$  を生成する。この時点で、 $\mathcal{M}$  と各  $\mathcal{M}_i$  は等価である。

### 2.3 Domain Adapt

Domain Adapt では、 $D_i$  で  $\mathcal{M}_i$  を Fine-tuning し、各  $\mathcal{M}_i$  にドメイン内での汎化性能を獲得させる。このとき、学習するパラメータは FFN のパラメータのみとし、その他のサブレイヤーは凍結して学習される。

### 2.4 MiX

MiX では、 $\mathcal{M}$  と各  $\mathcal{M}_i$  を MoE に基づいてマージし、WebNavix の最終モデルを構築する。FFN $_i^l$  が  $\mathcal{M}_i$  の  $l$  層目の FFN であるとき、入力表現  $x$  に対する  $l$  層目の MoE サブレイヤー FFN $_{\text{MoE}}^l(x)$  は次のように計算される。

$$\text{FFN}_{\text{MoE}}^l(x) = \sum_{i=1}^N g(W^l x) \text{FFN}_i^l(x)$$

ここで、 $W^l$  は線形変換である。また、 $g$  は通常スパースな出力を持つルーティング関数であり、実験では  $g(W^l x) := \text{SoftMax}(\text{Top-k}(W^l x))$  ( $k=2$ ) で定義する。Self-attention やその他のサブレイヤーについては、Domain Adapt 時の凍結により  $\mathcal{M}$  と各  $\mathcal{M}_i$  で等価であるため、 $\mathcal{M}$  のパラメータをそのまま使用する。このマージにより、新しい MoE モデル  $\mathcal{M}_{\text{MoE}}$  を得る。ただし、 $\mathcal{M}_{\text{MoE}}$  は未学習のパラメータ  $W^l$  を持つため、通常  $\mathcal{D}$  によって再度 Fine-tuning する必要がある。しかし、一般に MoE モデルの Fine-tuning には大規模な計算環境を要するため、効率的な継続学習の観点からこの方法は現実的ではない。そのため、WebNavix では非学習手法により  $W^l$  を決定する。

FFN $_i^l$  に対する適切な入力表現  $x_i^l$  が与えられたとき、 $\mathcal{M}$  の  $L$  層目までの各隠れ状態  $h_i^l$  は次のように計算される。

$$[h_i^l]_{l=1}^L = \mathcal{M}(x_i)$$

このとき、各層における各系列の隠れ状態は  $h_i^l$  として平均される。これを用いて、 $W_l$  は次のように決定される。

$$W^l = [h_i^l]_{i=1}^N$$

これにより、MiX 全体が非学習手法により実現され、ごく少ないコストでの継続学習が可能となる。以上の手順により、WebNavix は、データセット内のすべてのドメインでの汎化性能の獲得と継続学習の効率化を実現する。

WebNavix: Continual Generalist Web Navigation Agent with Domain-wise Mixture-of-Experts

Haruto Shiohata, Itsuki Shibenuma, Koh Ikeda, National Institute of Technology, Ibaraki College

<sup>1</sup>ポジティブプロンプトと呼称され、その ELM が対応するべきプロンプトの具体的な例が与えられる。

### 3 実験

従来手法に対する WebNavix の有効性を定量的に示すために、Web Navigation のための広範なベンチマークである WebLINX [4] によって評価実験を行う。

#### 3.1 実験設定

本実験における評価対象として、以下のパラメータをもとに WebNavix モデルを構築する。<sup>2</sup>

- $\mathcal{D}$ : WebLINX
- $\mathcal{M}$ : Sheared-LLaMA-2.7B [5] (従来 SoTA)
- $i$ : WebLINX の各カテゴリ<sup>3</sup> ( $N = 8$ )

これに対し、WebLINX が提供するメトリクスに従い、 $\text{TeST}_{\text{IID}}$  の\_intent・マッチ (IM)、\_element・グループ (IoU)、\_text・グループ (F1)、\_総合スコア (ターンレベルのスコアのマイクロ平均) の結果を報告する。比較対象として、いずれかの項目で SoTA を持つモデルのスコアを WebLINX 論文から引用する。なお、予算の制限により、WebNavix モデルは 4bit 量子化により推論した結果を報告する。

#### 3.2 結果

主な結果を表 1 に示す。本実験における WebNavix は量子化された推論であるにもかかわらず、総合スコアの SoTA を 7.48% 更新した。併せて、IoU は 31.26%、F1 は 6.59% 向上した一方、IM では 2.16% の低下が見られた。

#### 3.3 ルーティング解析

WebNavix の性能をより深く調査するために、実験で構築した WebNavix モデルのルーティング解析を行う。ルーティング解析は、WebLINX の各カテゴリに属する入力テキストに対する WebNavix モデルの各層のルーティング結果を可視化することで行う。理想的には、各層で入力テキストのドメインと割当てられた ELM のドメインが一致することが期待されるが、図 1 に示すように、WebNavix モデルのゲーティングネットワークは入力テキストのドメインに一切影響されず、層に影響されてルーティングを決定していることが確認された。この原因として、Task Adapt 時の epoch 数が多く、その時点でデータセットを十分に学習してしまい、Domain Adapt 時の学習で各 ELM の能力に十分な差が生じなかったことが考えられる。

### 4 課題と展望

Web Navigation のためのモデルは、システムの機密性や可用性の観点から、ユーザーのマシン上で動作可能であるように十分に軽量であることが望まれる。しかし、本実験で構築した WebNavix モデルのサイズは 14.6B であり、これは量子化などによりモデルサイズを圧縮しなければ、一般的な PC やスマートフォン上で動作させることはできない。

また、WebNavix は HTML からウェブページの情報抽出するが、この方法では画像コンテンツのような視覚的要

表 1: WebLINX に基づく評価の主な結果。予算の制限により、WebNavix は 4bit 量子化により推論した結果を報告する。

Models	Size	IM	IoU	F1	OVERALL
S-LLaMA	2.7B	<b>87.70</b>	35.54	37.66	37.43
Llama-2	13B	<b>87.70</b>	35.92	37.43	37.09
WebNavix <sup>q</sup>	14.6B	85.81	<b>47.15</b>	<b>40.14</b>	<b>40.23</b>

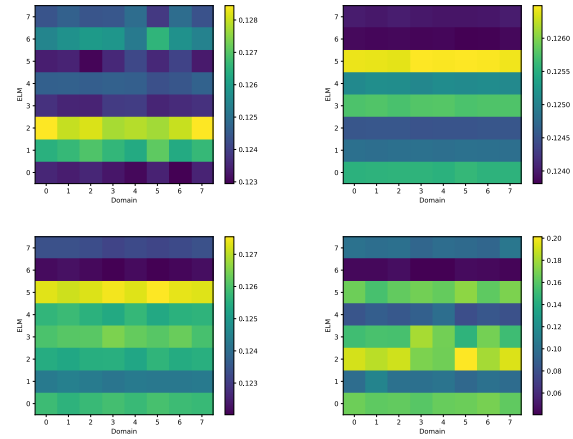


図 1: WebNavix モデルのルーティング解析結果。左上が 1 層目、右上が 8 層目、左下が 16 層目、右下が 32 層目のルーティング結果を示す。

素の情報をモデルが得ることはできない。従って、今後の研究では、WebNavix の視覚言語モデルへの応用が望まれる。

### 5 おわりに

本論文では、個別に学習されたドメイン専門家としての LLM を継続的に統合する MoE に基づくドメイン別モデルマージの方法である WebNavix を、初めて Web Navigation に導入した。実験では、WebLINX による評価において、WebNavix が最近の手法と比較して優れた性能を示すことが確認された。この成功の一方で、WebNavix は、学習時に肥大化したモデルサイズの圧縮や、視覚言語モデルへの応用などの、実用化に際して解決しなければならない課題を多数残している。

### 参考文献

- [1] X. Deng et al., “Mind2Web: Towards a Generalist Agent for the Web.” arXiv: 2306.06070, 2023.
- [2] A. Q. Jiang et al., “Mixtral of Experts.” arXiv: 2401.04088, 2024.
- [3] S. Sukhbaatar et al., “Branch-Train-MiX: Mixing Expert LLMs into a Mixture-of-Experts LLM.” arXiv: 2403.07816, 2024.
- [4] X. H. Lù, Z. Kasner, and S. Reddy, “WebLINX: Real-World Website Navigation with Multi-Turn Dialogue.” arXiv: 2402.05930, 2024.
- [5] M. Xia, T. Gao, Z. Zeng, and D. Chen, “Sheared LLaMA: Accelerating Language Model Pre-training via Structured Pruning.” arXiv: 2310.06694, 2024.

<sup>2</sup><https://huggingface.co/collections/nitic-nlp-team/webnavix-6733e035c7b41d86090b2bf3>

<sup>3</sup>AI Tools, Booking, Composing, Information Lookup, Shopping, Social Interaction, Summarizing, Task Management