

---

## エンド ツー エンドの目標駆動型 Web ナビゲーション

---

Rodrigo Nogueira  
Tandon ニューヨーク大学工学部  
rodrigonogueira@nyu.edu

Kyunghyun Cho  
Courant ニューヨーク大学数理科学研究所  
kyunghyun.cho@nyu.edu

### 概要

自然言語を理解し、部分的に観察された環境を計画する能力を持つエージェントを評価するためのベンチマーク タスクとして、目標駆動型の Web ナビゲーションを提案します。この困難なタスクでは、エージェントは、ノードとしての Web ページと有向エッジとしてのハイパーリンクで構成されるグラフとして表される Web サイトをナビゲートして、クエリが表示される Web ページを見つけます。エージェントが成功するには、自然言語に基づく洗練された高度な推論と、効率的な順次意思決定能力が必要です。WebNav と呼ばれるソフトウェア ツールをリリースします。これは、Web サイトをこの目標駆動型の Web ナビゲーション タスクに自動的に変換するもので、例として、英語の Wikipedia から構築されたデータセットである WikiNav を作成します。WikiNav でニューラル ネット ベースの人工エージェントのさまざまなバリエーションを広範に評価し、提案された目標駆動型の Web ナビゲーションがモデルの進歩をよく反映していることを観察し、将来の進歩を評価するための適切なベンチマークにします。さらに、Jeopardy! からの質問と回答のペアで WikiNav を拡張します。強力な逆インデックス ベースの検索エンジンに対して、リカレント ニューラル ネットワークに基づいて提案されたエージェントをテストします。WikiNav でトレーニングされた人工エージェントは、エンジンベースのアプローチよりも優れており、提案された目標駆動型ナビゲーションの機能が、焦点を絞ったクローラや質問応答などの現実世界のタスクの進行状況を測定するための優れたプロキシとして実証されています。

### 1 はじめに

近年、多くの比較的大規模で複雑なタスクを解決するために、1 つの学習アルゴリズムでトレーニングできる人工エージェントの構築に多くの刺激的な進歩がありました ([8, 10, 6]などを参照)。これらの作品の多くは、ターゲット タスクが Atari ゲーム [8] やレーシング カー ゲーム [6] などのコンピューター ゲームでした。

これらの成功により、研究者は、マルチユーザー ダンジョン (MUD) ゲームなどの言語ベースのタスクに同様の学習メカニズムを適用するようになりました [9, 4]。視覚的な認識の代わりに、エージェントは記述された記述によって世界の状態を認識します。エージェントに許可される一連のアクションは、固定されているか、現在の状態に依存しています。このタイプのタスクは、エージェントの計画能力だけでなく、言語理解能力も効率的に評価できます。

ただし、これらの MUD ゲームは、自然言語の複雑な性質を十分に発揮していないことに気がきました。たとえば、Narasimhan らによってテストされた最大のゲーム世界 [9] は 1340 個のユニークな単語のみの語彙を使用し、He らによってテストされた最大のゲームです。[4] は 2258 語しか使用していません。さらに、各時間ステップでの状態の記述は、ほとんどの場合、現在のシーンの視覚的な記述に限定されており、自然言語に存在する高レベルの概念は使用されていません。

この論文では、自然言語の理解と計画能力を備えた人工エージェントを評価するためのテキストベースのゲームに代わる大規模な代替手段として、ゴール駆動型の Web ナビゲーションを提案します。提案された目標駆動型 Web ナビゲーションは、Web ページがノードでハイパーリンクが有向エッジであるグラフとしての Web サイト全体で構成されます。エージェントには、グラフでランダムに選択された Web ページから取得された 1 つまたは複数の文で構成されるクエリが与えられ、ネットワークをナビゲートします。

定義済みの開始ノードから開始して、クエリが表示されるターゲット ノードを見つけます。テキストベースのゲームとは異なり、このタスクは既存のテキストをそのまま利用するため、状態を真に自然な言語で説明する大量の語彙が得られます。さらに、エージェントの状態によって行動空間が大きく変化するため、タスクはより困難になります。

WebNav と呼ばれるソフトウェア ツールをリリースします。これは、特定の Web サイトを目標主導型の Web ナビゲーション タスクに変換します。その使用例として、英語のウィキペディアから構築された WikiNav を提供します。教師あり学習でトレーニングされたニューラル ネットワーク (NeuAgents と呼ばれる) に基づいて人工エージェントを設計し、ベンチマーク タスクでのそれぞれのパフォーマンスと人間のボランティアのパフォーマンスを報告します。WebNav によって生成されたタスクの難易度は、2 つの制御パラメーターによって適切に制御されていることがわかります。(1) 開始ノードからターゲット ノードまでの最大ホップ数  $N_h$ 、および (2) クエリの長さ  $N_q$ 。

さらに、Jeopardy! から構築されたクエリの追加セットで WikiNav を拡張します。WikiNav-Jeopardy で参照する質問。提案された NeuAgents を 3 つの検索ベースの戦略に対して評価します。(1) SimpleSearch、(2) Apache Lucene、および (3) Google 検索 API。ドキュメントの想起に関する結果は、NeuAgents がこれらの検索ベースの戦略よりも優れていることを示しており、提案されたタスクが、質問応答や焦点を絞ったクロールなどの実用的なアプリケーションの優れたプロキシとしての可能性を示唆しています。

## 2 目標主導の Web ナビゲーション

ゴール駆動型 Web ナビゲーションのタスク  $T$  の特徴は次のとおりです。

$$T = (A, sS, G, q, R, \Omega). \quad (1)$$

エージェント  $A$  がナビゲートする世界は、グラフ  $G = (N, E)$  として表されます。グラフは、ノード  $N = \{s_i\}$  のセットと、これらのノードを接続する有向エッジ  $E = \{e_{ij}\}$  のセットで構成されます。Web サイトの各ページ  $s_i$  を表し、その中の各欄合にのみ、ページ  $s_i$  から  $s_j$  に向かうエッジが存在します。ノードの 1 つは、ナビゲーションが始まる開始ノード  $sS$  として指定される。ターゲット ノードは、クエリ  $q$  を含む自然言語記述を持つノードであり、複数のターゲット ノードが存在する場合があります。

各タイム ステップで、エージェント  $A$  は、エージェントが到達した現在のノードの自然言語記述  $D(s_t)$  を読み取ります。ノードとエッジで構成される世界全体も、その構造やマップ (自然言語による記述のないグラフ構造) もエージェントには見えないため、このタスクは部分的に観察されます。

エージェント  $A$  が現在のノード  $s_i$  の記述  $D(s_i)$  を読み取ると、利用可能なアクションの 1 つを実行できます。可能なアクションのセットは、すべての出力エッジ  $e_{ij}$  と停止アクションの結合として定義され、エージェントに状態依存のアクション空間を持たせます。

各エッジ  $e_{ij}$ 、 $k$  は、エージェントが次のノード  $s_k$  にジャンプすることに対応し、停止アクションは、現在のノード  $s_i$  がターゲット ノードの 1 つであることをエージェントが宣言することに対応します。各エッジ  $e_{ij}$ 、 $k$  は、次のノード  $D(s_k)$  の記述によって表されます。つまり、実行するアクションを決定することは、隣接する各ノードをのぞいて、そのノードが最終的にターゲット ノードにつながる可能性があるかどうかを確認することと同じです。

エージェント  $A$  は停止アクションを選択したときに報酬  $R(s_i, q)$  を受け取ります。このタスクは単純なバイナリ報酬を使用します。

$$R(s_i, q) = \begin{cases} q \subseteq D(s_i) \text{ の場合は } 1、 \\ \text{それ以外の場合は } 0 \end{cases}$$

制約エージェントがすべての試行で成功するための最終的なポリシーが存在することは明らかです。これは、エージェントがクエリが表示されるノードを見つけるまで、幅優先でグラフをトラバースすることです。この種の退化したポリシーを回避するために、タスクには一連の 4 つのルール/制約  $\Omega$  が含まれています。

1. エージェントは各ノードで最大  $N_n$  個のエッジをたどることができます。
2. エージェントは、 $T$  より小さいサイズの有限のメモリを持っています。
3. エージェントが  $sS$  から最大で  $N_h$  ホップ移動します。

表 1: WikiNav-4-\*, WikiNav-8-\*, WikiNav-16-\*, および WikiNav-Jeopardy のデータセット統計。

	WikiNav-4-*	WikiNav-8-*	WikiNav-16-*	WikiNav-Jeopardy
訓練	6.0k	1M	12M	113k
有効	1k	20k	20k	10k
テスト	1k	20k	20k	10k

4. サイズ  $N_q$  のクエリは、開始ノードから少なくとも 2 ホップ離れたところから来ます。

最初の制約だけでは、エージェントが各ノードで可能な限り適切な決定を行うように強制する、幅優先探索などの縮退ポリシーが防止されます。2 番目の追加の制約は、エージェントが以前の試行を使用して (テスト時に) グラフ構造全体を再構築したり、(トレーニング時に) メモリに全世界を格納したりすることによって、不正行為をしないことを保証します。オプションの 3 番目の制約は次のとおりです。計算上の考慮事項があります。エージェントが次のノードのコンテンツを読み取ることが許可されているため、4 番目の制約が含まれています。

### 3 WebNav: ソフトウェア

この作業の一環として、Web サイトを目標主導型の Web ナビゲーション タスクに変えるソフトウェア ツールを構築し、リリースしています。<sup>1</sup> このツールを WebNav と呼んでいます。開始 URL を指定すると、WebNav は Web サイト全体を読み取り、Web サイト内の Web ページをノードとしてグラフを作成します。各ノードには  $a$  が割り当てられます。各ノードのテキスト コンテンツ  $D(s_i)$  は、WebNav は、サイト内ハイパーリンクの意図を識別し、バージョンです。対応する Web ページのコンテンツ。

ウェブサイトを式からグラフ  $G$  に変換することに加えて、(1)、WebNav はノードのテキストからクエリを自動的に選択し、それらをトレーニング、検証、およびテスト セットに分割します。クエリが選択される各ターゲット ノードがそれらの 1 つのみに属するようにすることで、3 つのセット間で重複がないようにします。

生成された各例はタプルとして定義されます

$$X = (q, s, p) \quad (2)$$

ここで、 $q$  は Web ページ  $s$  からのクエリです、これは、ランダムに選択されたパス  $= (s, s_1, \dots, s_n)$  に従って見つかりました。言い換えると、WebNav は開始ページ  $s$  から開始し、ランダム ウォークして、事前定義されたステップ数 (この場合は  $N_h/2$ ) のグラフは、 $D(s)$  からターゲット ノード  $s$  クエリ  $q$  に到達します。クエリは  $N_q$  文で構成され、平均 TF-IDF が最も高いターゲット ノードの上位 5 候補の中から選択されるため、WebNav が些細なクエリを選択するのを思いとどまらせます。

評価目的であれば、クエリ  $q$  自体を例として使用するだけで十分です。ただし、1 つのターゲット ノード (潜在的に他の多くのターゲット ノードの中) と、開始ノードからこのターゲット ノードへの 1 つのパス (多くの可能な接続パスの中) の両方を含めて、エージェントのトレーニング時にそれらを利用できるようにします。訓練されたエージェントを評価するとき、それらは使用されるべきではありません。

### 4 WikiNav: ベンチマーク タスク

WebNav を使用して、Wikipedia をターゲット Web サイトとして使用して、ベンチマークの目標駆動型ナビゲーション タスクを構築しました。2015 年 9 月の英語版ウィキペディアのダンプ ファイルを使用しました。これは、500 万を超える Web ページで構成されています。許可されるホップの最大数  $N_h \in \{4, 8, 16\}$  とクエリのサイズ  $N_q \in \{1, 2, 4\}$  を変えることで、難易度の異なる一連の個別のタスクを作成しました。各タスクは WikiNav- $N_h$ - $N_q$  で参照します。

タスクごとに、開始ページから許容最大ホップ数の半分のホップ数離れたページからトレーニング、検証、およびテストの例を生成します。

<sup>1</sup> ソース コードとデータセットは、<https://drive.google.com/folderview?id=0B5LbsF70cHjqUFhWQ242bzdITWc&usp=共有> で公開されています。

<sup>2</sup> この制限は、計算上の理由から選択した人為的な制限です。

表 3: WikiNav-Jeopardy からクエリと回答のペアの例。

クエリ	答え
の人生の最後の 8 年間、ガリレオはこの男の理論を支持したために自宅軟禁されていました。	コペルニクス
1971 年から 1972 年の冬、この州のレーニア パラダイス レンジャー ステーションでは記録的な 1,122 インチの雪が降りました。	ワシントン
1960 年に発表されたこの会社のアキュロン時計は、月に 1 分以内の精度が保証されていました。	ブローバ

最小限のクリーンアップ手順として、タイトルが「Wikipedia」で始まるメタ記事を除外しました。

ウィキペディア外の Web ページにつながるハイパーリンクは、「参考文献」、「外部リンク」、「参考文献」、「部分参考文献」のセクションとともに事前に削除されます。

表 2 に、英語版ウィキペディアの基本的な記事ごとの統計を示します。これらの統計から、クリーンアップ手順の後でも、WikiNav-Nh-Nqの世界が大きく複雑であることは明らかです。

	ハイパーリンク	ワード
12月	4.29 462.5	113.890.2
√ 最小	300 132881	0 1

最終的に、WikiNav-4-\* のデータセットはかなり小さくなりましたが、WikiNav-8-\* と WikiNav-16-\* のデータセットは大きくなりました。詳細については、表 1 を参照してください。

表 2: 英語版ウィキペディアのページごとの統計。

#### 4.1 関連作品: ウィキスピードディア

知的エージェントが特定の目標を達成するために探索する世界として、ウェブサイトまたはウェブ全体の可能性に気付いたのは、この研究が初めてではありません。私たちにとって最も関連性の高い最近の研究の 1 つは、おそらく [14, 12, 13] の Wikispeedia です。

西等。 [14, 12, 13] は、Wikispeedia と呼ばれる次のゲームを提案しました。ゲームの世界は、この作品で提案されている目標駆動型のナビゲーション タスクとほぼ同じです。具体的には、2008 年現在で約 4,000 件の記事が掲載されている「学校向けウィキペディア」を、記事をノード、ハイパーリンクを有向辺とするグラフに変換しました。このグラフからノードのペアがランダムに選択され、エージェントに提供されます。

エージェントの目標は、最初のノードから開始し、グラフをナビゲートして 2 番目のノードに到達することです。WikiNav と同様に、エージェントは現在のノードとすべての隣接ノードのテキスト コンテンツにアクセスできます。大きな違いの 1 つは、ターゲットが記事全体として与えられることです。つまり、提案された WikiNav には複数のターゲットノードが存在する可能性があるのに対し、Wikispeedia には単一のターゲット ノードがあります。

この説明から、ゴール駆動型の Web ナビゲーションは、Wikispeedia の一般化と再構成であることがわかります。まず、クエリに含まれる情報を少なくして、エージェントがターゲット ノードに移動するのをより困難にします。さらに、West と Leskovec [13] による主要な研究課題は、「人間がどのようにナビゲートし、探している情報を見つけるかを理解する」ことでした。人工知能エージェントを設計および評価するためのタスク。

## 5 WikiNav-ジェパディ:ジェパディ!ウィキナビで

ゴールドライブナビゲーションを利用した潜在的な実用化の 1 つは、世界の知識に基づく質問応答です。この Q&A タスクでは、クエリは質問であり、エージェントは Web サイトなどの特定の情報ネットワークをナビゲートして回答を取得します。このセクションでは、クエリとターゲットのペアが実際の Jeopardy! から構築される WikiNav の拡張を提案し、説明します。質問と回答のペア。WikiNav-Jeopardy による WikiNav のこの拡張を参照します。

まず、J からすべての質問と回答のペアを抽出します。アーカイブ 3 組。回答がウィキペ 、 そのようなものが 300k 以上あります。ディアの記事のタイトルであるペアのみを保持し、13 万 3,000 のペアが残ります。これらのペアを 113,000 のトレーニング、10,000 の検証、および 10,000 のテスト例に慎重に分割します。

<sup>3</sup> <http://www.j-archive.com>

複数のパーティションに記事が表示されないようにします。さらに、トレーニングとテストの例が異なるエピソードからのものであることを保証するために、元のペアをシャッフルしません。

トレーニング ペアごとに、開始ノード「メイン トピック分類」からターゲット ノードまでの 1 つのパスを見つけ、それを教師あり学習に含めます。参考までに、対象ノードまでの平均ホップ数は 5.8、標準偏差は 1.2、最大と最小はそれぞれ 2 と 10 です。クエリと回答のペアの例については、表 3 を参照してください。

## 6 NeuAgent: ニューラル ネットワーク ベースのエージェント

### 6.1 モデルの説明

コア関数 NeuAgent のコアは、現在のノード  $\phi_c(s_i)$  の内容とクエリ  $\phi_q(q)$  を入力として取り、エージェントの隠れ状態を返すパラメトリック関数  $f_{core}$  です。

このパラメトリック関数  $f_{core}$  は、フィードフォワード ニューラル ネットワーク  $fff$  として実装できます。

$$h_t = fff(\phi_c(s_i), \phi_q(q))$$

これは、エージェントの以前の隠れ状態を考慮していないか、再帰型ニューラルネットワーク  $f_{rec}$  として考慮していません。

$$h_t = f_{rec}(h_{t-1}, \phi_c(s_i), \phi_q(q))$$

これら 2 種類のエージェントを、それぞれ NeuAgent-FF と NeuAgent-Rec と呼びます。 NeuAgent-FF では単一の  $\tanh$  レイヤーを使用しますが、NeuAgent-Rec では最近事実上の標準となった長短期記憶 (LSTM) ユニット[5]を使用します。

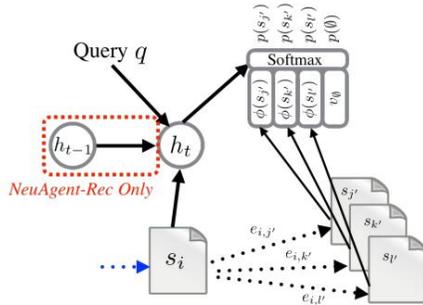


図 1: ベースラインモデル、NeuAgent によって実行される 1 つのステップの図。

新しい隠れ状態  $h_t$  に基づいて、NeuAgent はすべての出力エッジ  $e_i$  の確率分布を計算します。各出力エッジの確率は、次のような隠れ状態  $h_t$  間の類似度に比例します。

$$p(e_{i,j} | p_{\sim}) \propto \exp \phi_c(s_j) \cdot h_t \quad (3)$$

NeuAgent は、次のノード  $s_j$  の内容を、そのベクトル表現  $\phi_c(s_j)$  を考慮してピークすることに注意してください。

すべての発信エッジに加えて、エージェントが確率で停止することも許可します

$$p(\emptyset | p_{\sim}) \propto \exp v_{\emptyset} \cdot h_t \quad (4)$$

ここで、停止アクション ベクトル  $v_{\emptyset}$  はトレーニング可能なパラメーターです。

NeuAgent-Rec の場合、これらすべての (正規化されていない) 確率は、これまでにエージェントによって選択された一連のアクション (ノード) である履歴  $p_{\sim}$  に基づいて調整されます。正規化されていない確率にソフトマックス正規化を適用して、現在のノード  $s_i$  で可能なすべてのアクションの確率分布を取得します。

次に、NeuAgent は、このアクション確率分布に基づいて次のアクションを選択します(式(3)および(4))。停止アクションが選択された場合、NeuAgent は現在のノードを回答として返し、報酬  $R(s_i, q)$  を受け取ります。これは、正しい場合は 1、そうでない場合は 0 です。エージェントが発信エッジの 1 つを選択すると、エージェントは選択されたノードに移動し、この読み取りと動作のプロセスを繰り返します。

説明した NeuAgent の 1 つのステップについては、図 1 を参照してください。

コンテンツ表現 NeuAgent は、ノード  $s_i$  のコンテンツをベクトル  $\phi_c(s_i) \in \mathbb{R}^d$  として表現します。この作業では、各ドキュメントに連続する bag-of-words ベクトルを使用します。

$$\phi_c(s_i) = \frac{1}{|D(\text{はい})|} \sum_{k=1}^{|D(\text{はい})|} \text{私}_k$$

各単語ベクトル  $lek$  は、事前学習済みの連続バグオブワード モデル[7] からのものです。これらの単語ベクトルは、トレーニング全体で固定されています。

クエリ表現 クエリの場合、2種類の表現を考えます。最初のものは、ノードのコンテンツを表すために使用されるのと同じように、連続バグワード (BoW) ベクトルです。もう 1 つは注意メカニズムに基づく動的表現です [2]。

注意ベースのクエリ表現では、クエリは最初に一連のコンテキスト ベクトルに射影されます。k 番目のクエリ ワードのコンテキスト ベクトルは

$$c_k = \frac{1}{\sqrt{u}} \sum_{i=k-u/2}^{k+u/2} w_i$$

ここで、 $w_k \in \mathbb{R}^d$  はそれぞれトレーニング可能な重み行列と事前トレーニング済みの単語ベクトルです。 $u = \text{fatt}(ht-1, \text{size})$  です。各コンテキスト ベクトルは、各タイム ステップ  $t$  で、NeuAgent の以前の隠れ状態  $\beta$  によって  $c_k$  を射影されたすべてのスコアが正になるように正規化され、 $\exp(\beta)$  を 1 に合計します。つまり、 $\alpha |q| = 1$

$$\alpha_k = \frac{\exp(\beta_k)}{\sum_{k=1}^{|q|} \exp(\beta_k)}$$

時間  $t$  でのクエリ表現になるクエリ単語の加重和:

$$\phi(q) = \sum_{k=1}^{|q|} \alpha_k c_k$$

後で、これら 2 つのクエリ表現を経験的に比較します。

## 6.2 推論: ビーム探索

NeuAgent がトレーニングされると、提案されたタスクを解決するために NeuAgent を使用方法がいくつかあります。最も素朴なアプローチは、単純に各時間ステップでエージェントに貪欲な決定をさせることです。つまり、最も高い確率  $\arg \max_k \log p(e_i, k | \dots)$  で出力エッジをたどります。より良いアプローチは、エージェントがノードごとに最大  $N_n$  個の発信エッジを探索できるという事実を利用することです。ビーム幅を  $N_n$  に制限した単純な前方のみのビーム探索を使用します。ビーム探索は、各タイム ステップで、 $\log p(e_i, k | \dots)$  に関して、 $N_n$  個の最も可能性の高いトレースを単純に保持します。

## 6.3 トレーニング: 教師あり学習

この論文では、教師あり学習を調査します。ここでは、各ステップでトレーニング セットに含まれるサンプル トレース  $(s_1, \dots, s_p)$  に従うようにエージェントをトレーニングします (式(2)を参照)。この場合、トレーニング例ごとのコストは

$$C_{\text{sup}} = -\log p(\mathcal{Q} | p, q) - \sum_{k=1}^{|p|} \log p(p_k | p_{<k}, q) \quad (5)$$

この例ごとのトレーニング コストは、ニューラル ネットワークのすべてのパラメーターに関して完全に微分可能であり、確率的勾配降下法 (SGD) アルゴリズムを使用して、トレーニング セット全体でこのコストを最小化します。勾配はバックプロパゲーションによって計算できます [11]。これにより、モデル全体をエンドツーエンドの方法でトレーニングでき、クエリからターゲットへのパフォーマンスが直接最適化されます。

## 7 人による評価

提案されたタスクのユニークな側面の 1 つは、情報ネットワークをナビゲートして情報を見つけるための特別な訓練を受けていない平均的な人にとっては非常に難しいことです。この困難の背後には多くの理由があります。まず、トレーニングを通じて、ネットワークのグラフ構造に精通する必要があります。これには、数年とは言わないまでも、数か月のトレーニングが必要になることがよくあります。第 2 に、ターゲット ノードへのクエリのテーマとトピックの間のさまざまな概念を介して接続するために、その人は幅広いトピックに関する深い知識を持っている必要があります。第 3 に、試行ごとにノードの内容全体を注意深く読む必要があり、これは時間と労力のかかる作業です。

5 人のボランティアに、WikiNav-{4, 8, 16}-4 データセットのテストセットから無作為に選択された最大 20 個の 4 文の長さのクエリ 4 を試すよう依頼しました。彼らは最大 2 時間与えられ、

<sup>4</sup> 他のボランティアとの予備調査では、クエリが 4 よりも短い場合、彼らは意味のある統計を得るのに十分な数の試行を解決できませんでした。

表 4: WikiNav-Nh-Nq のテスト セットでの NeuAgent と人間による平均報酬。

fcore # Layers × #	q Nh	Nq = 1															
		4	8	16	4	8	16	4	8	16							
17.7 (c) (d) fcore 8 × 2048 LSTM Att 22.9 (e)	12	4.7	1.2	40.0	9.2	1.9	45.1	12.9	2.9	5.1	1.7	41.1	9.2	2.1	44.8	13.3	3.6
人間	-	-	-	-	-	-	-	-	-	-	-	-	14.5	8.8	-	-	-

NeuAgents と同じノードNnあたりの探索エッジの最大数(つまり、Nn = 4) まで選択し、あきらめるオプションも与えられました。平均報酬は、提示されたすべてのクエリに対する正しい試行の割合として計算されました。

## 8 結果と分析

### 8.1 ウィキナビ

表 4 に、9 つのWikiNav-{4, 8, 16}-{1, 2, 4}データセットすべてのテスト セットに対する NeuAgent-FF および NeuAgent-Rec モデルのパフォーマンスを報告します。提案された NeuAgents に加えて、人間による評価の結果も報告します。

難易度は実際にはクエリの長さNqと負の相関がありますが、許容される最大ホップ数Nh とは正の相関があることが明確にわかります。後者は、探索空間のサイズがNhに関して指数関数的に増加するため、取るに足らないと考えられるかもしれませんが、前者はそうではありません。前者の負の相関関係は、クエリの情報が多いほどタスクを解決しやすいことを示しています。エージェントは、自然言語をより深く理解し、クエリの情報不足を克服してターゲットノードへのパスを見つける計画を立てる必要があると推測されます。

NeuAgent-FF と NeuAgent-Rec は、許可されるホップの最大数が小さい場合(Nh = 4) は同様のパフォーマンスを共有しますが、NeuAgent-Rec ((a) 対 (b)) はより高いNh で一貫してより優れたパフォーマンスを発揮します。履歴にアクセスできると、長期的な計画タスクに役立ちます。また、ターゲット ノードが開始ノード sS からさらに離れている場合、大きくて深い NeuAgent-Rec ((b) 対 (c)) が小さいものよりも大幅に優れていることもわかります。

(d) で最高のパフォーマンスを発揮するモデルは、特にタスクの難易度が上がるにつれて(Nq ↓とNh ↑)、アテンション ベースのクエリ表現を使用しました。将来の進歩を評価するため、図 2 では、モデルがターゲット ノードに向かって移動するにつれて、クエリワードに対する注意の重みがどのように動的に変化するかの例を示します。

人間の参加者は一般的に、NeuAgents よりも悪い成績を収めました。これにはいくつかの理由が考えられます。まず、NeuAgent はターゲット ドメイン (ウィキペディア) で特別にトレーニングされますが、人間の参加者はトレーニングを受けていません。第二に、ボランティアが複数の記事を順番に読むことで急速に疲れ果てていることが観察されました。言い換えれば、提案されたベンチマークである WebNav は、機械知能の優れたベンチマークであることがわかりますが、人間の知能と比較することはできません。

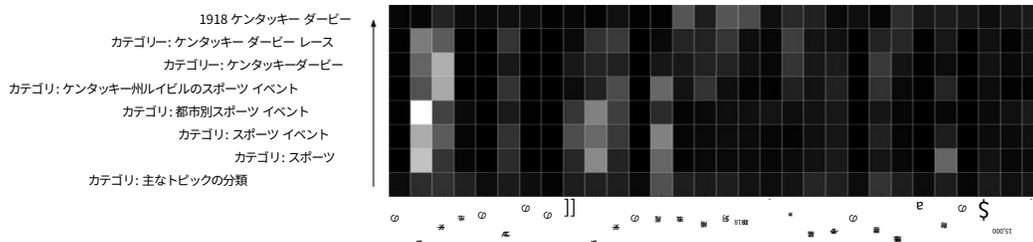


図 2: テスト クエリに対するアテンション ウェイトの視覚化。横軸は検索キーワード、縦軸はアクセスした記事のタイトルに対応しています。

## 8.2 WikiNav-Jeopardy

設定WikiNav-Jeopardyで、前の実験 (2048 LSTM ユニットの 8 層とアテンション ベースのクエリ表現を持つ NeuAgent-Rec) からの最良のモデルをテストします。2つのトレーニング戦略を評価します。最初の戦略は、WikiNav-Jeopardy で NeuAgent-Rec をゼロからトレーニングする、直接的な監督学習です。もう1つの戦略では、最初に WikiNav-16-4 で NeuAgent-Rec を事前トレーニングし、WikiNav-Jeopardy で微調整します。

提案された NeuAgent を3つの検索戦略と比較します。最初の SimpleSearch は単純な逆インデックス ベースの戦略です。SimpleSearchは、記事とクエリで共起する単語の TF-IDF 加重和によって各ウィキペディアの記事をスコア付けし、上位 K の記事を返します。2つ目は、人気のあるオープンソースの情報検索ライブラリである Lucene を、ウィキペディアのダンプ全体の既定の構成で使用します。最後に、ドメインをwikipedia.orgに制限しながら、Google Search API5 を使用します。

各システムは、K でのドキュメント リコールによって評価されます (Recall@K)。K を 1、4、または 40 に変化させます。NeuAgent の場合、幅を K に設定してビーム検索を実行し、K 個の最終ノードをすべて返してドキュメントの再現率を計算します。

表 5: WikiNav-Jeopardy でのリコール。() WikiNav-16-4 で事前トレーニング済み。

モデル	プレ	リコール@1	リコール@4	リコール@40
ニューエージェント		13.9	20.2	33.2
ニューエージェント		18.9	23.6	38.3
シンプルサーチ		5.4	12.6	28.4
ルセン		6.3	14.7	36.3
グーグル		14.0	22.1	25.9

結果と分析表 5 に、WikiNav-Jeopardy での結果を報告します。提案された NeuAgent は、WikiNav-16-4 で事前トレーニングされた場合、3つの検索ベースの戦略すべてより明らかに優れています。事前トレーニング済みの NeuAgent の優位性は、候補ドキュメントの数が少なくなるように制約されている場合により明白になります。これは、NeuAgent が正しいターゲット記事を正確にランク付けできることを意味します。NeuAgent は、事前トレーニングを行わなくても他の検索ベースの戦略と同等のパフォーマンスを発揮しますが、はるかに大規模な WikiNav での事前トレーニングの利点は明らかです。

これらの検索ベースの戦略は、各入力クエリのすべてのノードにアクセスできることを強調します。一方、NeuAgent は、ナビゲーション中にアクセスしたノードのみを監視します。この成功は、提案された WebNav によってコンパイルされたデータセットで事前トレーニングされた提案された NeuAgent を、集中的なクローリング[3, 1]のタスクに使用できる可能性を明確に示しています。非表示または動的に生成されます[1]。

## 9 結論

この作業では、大規模な目標駆動型の Web ナビゲーション タスクについて説明し、自然言語の理解と計画に関する人工エージェントの機能を評価するための有用なテスト ベッドとして機能することを主張します。WebNav と呼ばれるソフトウェア ツールをリリースします。これは、特定の Web サイトを目標駆動型の Web ナビゲーション タスクにコンパイルします。例として、WebNav を使用して Wikipedia から WikiNav を構築します。Jeopardy! で WikiNav を拡張します。質問、したがって WikiNav-Jeopardy を作成します。WikiNav と WikiNav-Jeopardy でさまざまなニューラル ネット ベースのエージェントを評価します。私たちの結果は、より洗練されたエージェントの方がパフォーマンスが優れていることを示しており、このタスクは自然言語の理解と計画の将来の進歩を評価するのに適しているという私たちの主張を裏付けています。さらに、WikiNav で事前トレーニングされたエージェントが、WikiNav-Jeopardy の2つの強力な逆インデックス ベースの検索エンジンよりも優れていることを示します。これらの経験的結果は、焦点を絞ったクローリングや質問応答などの挑戦的なアプリケーションにおける提案されたタスクとエージェントの有用性に関する私たちの主張を裏付けています。

<sup>5</sup> <https://cse.google.com/cse/>

## 謝辞

KC は、Facebook、Google (Google ファカルティ アワード 2016)、および NVidia (GPU センターオブ エクセレンス 2015-2016) によるサポートに感謝します。RN は、高等教育職員の改善のための調整 (CAPES)によって資金提供されています。

## 参考文献

- [1] マヌエル・アルバレス、ファン・ラボソ、アルベルト・パン、フィデル・カケダ、フェルナンド・ベラス、ピクトル・カルネイロ。Deepbot: 非表示の Web コンテンツにアクセスすることに特化したクローラー。電子商取引およびサービスにおけるデータ エンジニアリングの問題に関する第 3 回国際ワークショップの議事録: 電子商取引に関する ACM 会議 (EC'07) と併せて、18 ~ 25 ページ。ACM、2007 年。
- [2] Dzmitry Bahdanau, Kyunghyun Cho, および Yoshua Bengio。関節によるニューラル機械翻訳 整列と翻訳を学びます。ICLR 2015、2014。
- [3] Soumen Chakrabarti, Martin Van den Berg, Byron Dom。集中クローラー: トピック固有の Web リソース発見への新しいアプローチ。コンピュータ ネットワーク、31(11):1623-1640、1999 年。
- [4] Ji He, Jianshu Chen, Xiaodong He, Jianfeng Gao, Lihong Li, Li Deng, Mari Ostendorf。無制限のアクション スペースを使用した深層強化学習。arXiv プレプリント arXiv:1511.04636, 2015。
- [5] ゼップ・ホホライターとコルゲン・シュミットフォーバー。長期短期記憶。ニューラル計算、9(8): 1735年から1780年、1997年。
- [6] Jan Koutník, Jürgen Schmidhuber, および Faustino Gomez。ビジョンベースの強化学習のための教師なし深層畳み込みネットワークの進化。遺伝的および進化的計算に関する2014年の会議の議事録、541 ~ 548 ページ。ACM、2014 年。
- [7] トーマス・ミコロフ、カイ・チェン、グレッグ・コラード、ジェフリー・ディーン。ベクトル空間での単語表現の効率的な推定。arXiv プレプリント arXiv:1301.3781, 2013。
- [8] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fiedjeland, Georg Ostrovski, 他。深層強化学習による人間レベルの制御。ネイチャー、518(7540):529-533、2015。
- [9] Karthik Narasimhan, Tejas Kulkarni, Regina Barzilay。深層強化学習を使用したテキストベースのゲームの言語理解。arXiv プレプリント arXiv:1506.08941, 2015。
- [10] セバスチャン・リシとジュリアン・トゲリウス。ゲームにおける神経進化: 最先端かつオープン課題。arXiv プレプリント arXiv:1410.7326, 2014。
- [11] デビッド・ルメルハート、ジェフリー・ヒントン、ロナルド・ウィリアムズ。バックによる表現の学習 エラーの伝播。ネイチャー、323 ~ 533 ページ、1986 年。
- [12] ロバート・ウェストとジュレ・レスコベック。情報ネットワークにおける自動ナビゲーションと人間によるナビゲーション。ICWSM、2012年。
- [13] ロバート・ウェストとジュレ・レスコベック。情報ネットワークにおける人間の経路探索。第21回インテルナショナルの World Wide Web Conference。619 ~ 628 ページ。ACM、2012 年。
- [14] ロバート・ウェスト、ジョエル・ピノー、ドイナ・プレカップ。Wikispeedia: 概念間の意味的距離を推測するためのオンライン ゲーム。IJCAI、1598 ~ 1603 ページ、2009 年。