

# 言語モデルを用いた Ansible のファイル作成時における ヒューマンエラー発生防止システムの検討

川口真護<sup>†</sup> 水谷后宏<sup>‡§</sup> 井口信和<sup>‡§</sup>

近畿大学大学院総合理工学研究科<sup>†</sup> 近畿大学情報学研究所<sup>‡</sup> 近畿大学工学部情報学科<sup>§</sup>

## 1. 序論

ネットワークエンジニアの業務の一つとして、ネットワークの運用保守(以下、運用保守)がある。この業務は夜間など一般的な業務時間外での作業が多いため、ヒューマンエラーが多く起こるおそれがある[1]。また、ネットワークの運用管理の複雑化などの課題がある[2]。これら課題の解決策の一つとして、運用保守の自動化があり、大きな効果が見込める。この自動化を実現するためには、人工知能の技術などを用いて人の手が介在するところを減らす必要がある。本稿では、運用保守作業の一つであるネットワーク機器の設定変更作業に焦点をあてる。

ネットワーク機器の設定変更作業では、作業手順書の作成、ネットワーク機器の設定変更、ネットワークの動作確認を実施する。これらの作業を自動化する方法として、構成管理ツールを用いた方法があげられる。このツールの一つとして、2014年頃から最も注目されている Ansible を例にとる。Ansible を利用すると、ネットワーク機器の設定変更の自動化が可能となり、ヒューマンエラーの低減が期待できる。しかし、Ansible を用いた場合でも、実行する処理の流れが記載されたプレイブックファイルの作成をする時に、人の手が介在するため、ヒューマンエラーが起こる場合がある。

そこで本研究では、運用保守の自動化を目的に Ansible のファイル作成時におけるヒューマンエラーの発生防止を目標とし、言語モデルを用いた Ansible のファイル作成時におけるヒューマンエラー発生防止システム(以下、本システム)を開発する。本システムでは、Web ブラウザ上のコード補完機能が搭載されたエディタを用いて Ansible のファイル作成を実施する。

Examination of human error prevention system when creating Ansible files using language model

<sup>†</sup>Mamoru KAWAGUCHI, Graduate School of Science and Engineering Research, Kindai University

<sup>‡</sup>Nobukazu IGUCHI, Kimihiro MIZUTANI, Department of Informatics, Faculty of Science and Engineering, Kindai University

<sup>§</sup> Nobukazu IGUCHI, Kimihiro MIZUTANI, Cyber Informatics Research Institute, Kindai University

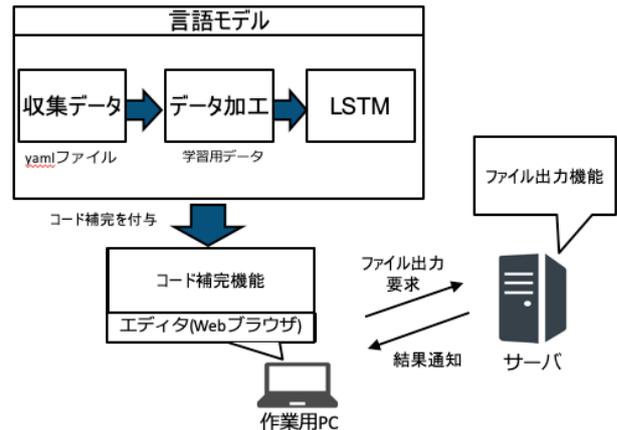


図1 システム構成

## 2. 関連研究

関連する研究として華山らの研究[3]がある。華山らの研究では、Docker のコード補完システムを開発している。

華山らの研究と本研究の類似点として、言語モデルを用いて、コード補完を実施している点が挙げられる。華山らの研究では、Docker に関する開発支援などの研究を推し進める事を目的としている。一方で、本研究では、運用保守の自動化を目的に Ansible のファイル作成時におけるヒューマンエラーの発生防止を目標としている。

## 3. 検討システム

本システムの構成を図1に示す。本システムはクライアントプログラムとサーバサイドプログラムによって構成される。クライアントプログラムは、言語モデルによって付与されるコード補完機能を有している。また、Web ブラウザ上のエディタを用いて Ansible のファイル作成を実施することができる。サーバサイドプログラムはファイル出力機能を有しており、クライアントプログラムからのファイル出力要求に応じて、ファイルを作成する。コード補完機能は、現在検討中の機能である。ファイル出力機能は、実装が完了した機能である。3.1節では、本システムのシステム要件について述べる。3.2節では、コード補完について述べる。3.3節以降では、本システムの機能について述べる。

### 3.1 システム要件

本研究の目標が Ansible のファイル作成時におけるヒューマンエラーの発生防止であるため、システム要件として次の2つを考えた。

要件 1: Ansible のファイル作成時の手作業を削減する

要件 2: 作成したコードを Ansible で運用できる形にする

この二つの要件に対応するために、コード補完機能、ファイル出力機能の実装を考えた。

### 3.2 コード補完

コード補完とは、コードの入力中に次のコードを予測し、補完候補を表示する機能である。コード補完は2種類存在している[3]。1つ目は、伝統的なコード補完である。伝統的なコード補完では、可能性のある補完候補を全て表示する。そのため、開発者がその非常に長い候補から入力値を選ぶ必要がある。このような課題を解決するために提案されたのが、2つ目のインテリジェントコード補完である。インテリジェントコード補完は、文脈を考慮し、言語モデルを用いて補完候補を絞ることで、伝統的コード補完よりも表示する候補が少なくなり、開発者の生産性を向上できる。また、N-gram, Best Matching Neighbor(BMN)や Recurrent Neural Network(RNN)などは、インテリジェントコード補完システムを構築する際に、高い性能を発揮してきた。そこで、本システムでのコード補完機能の実装には、RNNを用いた手法を採用している。

### 3.3 コード補完機能

本機能は、Ansible のファイル作成時のコード補完を行う機能である。本システムの利用者は Web ブラウザ上のエディタに任意のコードを記述する。言語モデルは、入力されたコードの文脈を考慮し、次のコードの候補をポップアップで表示する。本機能により、Ansible のファイル作成時の手作業を削減する事ができる。また、言語モデルを作成するために、データを収集し、そのデータを RNN で学習させる。以下に、本機能を実装する手順を説明する。

#### 3.2.1 データ収集

Ansible のドキュメントから Cisco モジュールを含む Ansible のファイルを収集する。

#### 3.2.2 データ加工

収集した Ansible のファイルから RNN の学習に必要なコードのコメント部分を除去する。name や ip-address などのパラメータ部分は、Ansible を使用する人の環境によって変わるため、String 型は\*, Integer 型は\$に置換を実施する。また、入力内容と期待する内容がセットになるようにデ

```

1 - name: *
2   cisco_ios_ios_acls:
3     config:
4       - afi: ipv4
5         acls:
6           - name: *
7             aces:
8               - sequence: $
9                 protocol_options:
10                  icmp:
11                    tracert: true
12            state: merged
13
14
15
16 - name: *
17   cisco_ios_ios_acls:
18     config:
19       - afi: ipv4
20         acls:
21           - name: *
22             acl_type: standard
23             aces:
24               - grant: deny
25                 source:
26                   address: $
27               - grant: deny
28                 source:
29                   address: $

```

図2 エディタ画面

ータを加工する。

#### 3.2.3 言語モデルの作成

本システムでは、コードを予測するために言語モデルを使用する。言語モデルの作成には、Long Short-Term Memory (LSTM)を用いる。LSTMとは、RNNの一種であり、長期的な依存関係を学習する事ができるアーキテクチャである。

#### 3.2.4 言語モデルの展開

作成した言語モデルをWebブラウザ上のエディタに展開し、コード補完機能を付与する。

### 3.4 ファイル出力機能

本機能は、Ansibleのファイルを出力する機能である。本機能の画面の一部を図2に示す。本システムの利用者は、Webブラウザ上のエディタを用いて、Ansibleのファイルを作成し、保存ボタンを押下する。サーバは、作成されたAnsibleのファイルを利用者のローカル環境上に保存する。本機能により、利用者が作成したコードをファイルへ変換できる。

## 4. 結論

本研究では、運用保守の自動化を目的に Ansible のファイル作成時におけるヒューマンエラーの発生防止を目標とし、言語モデルを用いた Ansible のファイル作成時におけるヒューマンエラー発生防止システムを検討した。

## 参考文献

- 1) 有藤平八郎：ヒューマンエラーと睡眠，産業医学，Vol. 35, No. 4, p. 255(1995)。
- 2) 情報通信技術委員会：ネットワーク分野の課題とその解決に向けた TTC の取り組み，入手先< [https://www.soumu.go.jp/main\\_content/000578678.pdf](https://www.soumu.go.jp/main_content/000578678.pdf)>(参照 2021-12-2)。
- 3) 華山魁生，杉本真佑，楠本真二：言語モデルに基づく Dockerfile コード補完システムの案，電子情報通信学会技術研究報告信学技報，Vol. 120, No. 193, pp19-24(2020)。