Demonstration of Voice User Interface for Intelligent Network Orchestration

Xiaonan Xu(1), Haoshuo Chen(2), Jesse E. Simsarian(2), Roland Ryf(2), Mikael Mazur(2), Lauren Dallachiesa(2), Nicolas K. Fontaine(2), David T. Neilson(2)

(1) Aberdeen, NJ 07747, USA
(2) Nokia Bell Labs, 600 Mountain Ave., Murray Hill, NJ 07974, USA
xuxiaonan2008@gmail.com

Abstract: We demonstrate a voice user interface for ONOS, where natural language processing (NLP) is applied to translate human-spoken language into ONOS northbound API requests and to answer network-related questions with numerical reasoning.

1. Introduction

Data traffic continues to grow dramatically in telecommunications networks, along with the increased demand for more efficient and automated network management [1]. To interact with software-defined networking (SDN) controllers such as the Open Network Operating System (ONOS), which offers flow control to the underlying hardware infrastructure, network administrators commonly use a command line interface (CLI) and/or a graphical user interface (GUI) to perform various tasks. Traditionally, administrators would have to be familiar with the CLI commands and GUI of the SDN controller and are constrained to interact through computer terminals.

In this paper, we introduce a voice user interface employing natural language processing (NLP) technologies to simplify network management. The developed interface allows an administrator to communicate with an SDN controller in various natural languages such as English and Chinese to achieve human language to ONOS northbound API [2] translation (demo 1) and network-related question answering (QA) (demo 2), as illustrated in Fig. 1. Specifically, a voice command can be translated into a corresponding northbound API request using pre-trained artificial intelligent (AI) models for multi-class classification and named-entity recognition (NER). The voice control functionality augments other user interfaces, enabling administrators to interact with the network in a more natural way without learning specific CLI commands or GUI that are unique to a particular controller. Furthermore, voice interaction frees the user to interact with the network using mobile devices for off-site work and it increases user accessibility options. Additionally, we utilize an AI model for QA on tabular data to simplify the process of obtaining complex network information. This allows for the simplified analysis of large networks and can be seen as a first step towards platform-agnostic AI-powered network automation and troubleshooting.

2. Innovation

2.1. Design

Demo 1: Command translation presents functionality in two steps to translate human-spoken requests in one or multiple sentences into a northbound API of a SDN controller. In the first step, the spoken words in the request are identified and converted into readable text through NLP speech recognition technology [3]. In the second step, two NLP models accept the text as input and their outputs are obtained to form an API. Normally, two parts of information should be extracted from the text: 1) request header including both a HTTP method such as GET, POST or DELETE and main function of the API, and 2) request body which provides resource information such as a device identifier.The former is determined by a multi-class classification model and the latter is obtained from a NER model. Both models are based on bidirectional encoder representations from transformers (BERT) [4], which uses transformer encoder architecture [5] to generate bi-directional self-attention for input sequences.

In the multi-class classification model, different request headers are labeled as different classes. We create a number of written texts for each of the methods and split the data into training, validation, and test set with a proportion of 8:1:1. The text is reformatted into one sequence of tokens including initial and separation tokens. The hidden state of the first token at the output of the model is passed into a fully connected layer with a ReLU activation function to determine which label the input text corresponds to. During training, we set 10^{-5} as the learning rate to adjust network weights for gradient descent, and use cross entropy loss. Our model achieved a 99.4% accuracy on the classification task of test data, indicating that the model can accurately predict the request header corresponding to the voice request.

The NER model is used to locate and classify different named entities into pre-defined categories such as switch names, locations, quantities, etc. This model can be seen as a specific classification model, where the embedding

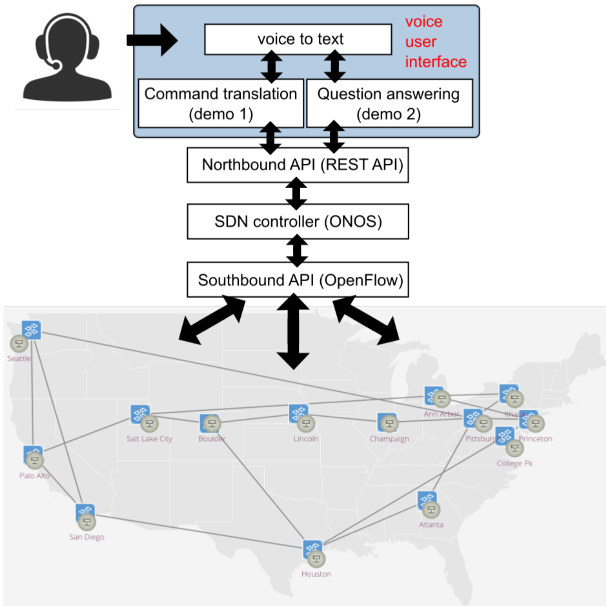

Fig. 1: Schematic of network orchestration with voice user interface.

vector output from all the tokens is classified into one of several pre-defined categories. During training, the words of text converted from voice request are labeled either as one of the pre-defined categories or as a word which does not belong to any category. Each entity can be a single word or a group of words. A fully connected layer on top of the model acts as a token-level classification layer to determine which category the entity in the text belongs to. Once the required entity, for example, a switch name is recognized from the input text, its mac address can be found from a look-up table. This information is used to form the request body, which combines with the request header obtained from the classification model to build a complete northbound REST API request.

Demo 2: QA presents the ability to provide answers to human-spoken questions with numerical reasoning based on the network information. This demo converts a human-spoken question to text through speech recognition and searches for an answer using a QA model about tabular data [6], which allows the system to find the most appropriate answer for a natural language question by using a knowledge base of information. In this case, the knowledge base is retrieved from the SDN controller using the command translation function and converted into tabular data. The QA model uses 1) text embeddings, 2) structure embeddings such as the value of each table cell and its column index, row index, and 3) one special rank index which indicates the order between values in numerical columns. The answer to a question can be predicted by a) first selecting table cells that are related to the question and b) then calculating the values in the table cells using aggregation operators like AVERAGE, COUNT, and SUM. The two steps are achieved by using two different classification layers at the model output. During fine-tuning, the model learns how to answer questions based on a table by computing the expectation over all the possible aggregation decisions and comparing it with the given correct answer.

2.2. *Example of Operation*

In both demos, ONOS controls a network of Mininet virtual packet switches with the topology of the National Science Foundation Network (NSFNET), consisting of 14 nodes and 21 bidirectional fiber links, as shown in Fig. 1. One operation example of command translation demo in English and Chinese, respectively, are illustrated in Fig. 2(a) and 2(b) using Jupyter notebook. In the English case, the interface monitors network operator's voice and converts verbal input *"Show flows of the device San Diego"* into text. The text is further processed by the classification and NER models. The classification model chooses the most appropriate one out of 29 command classes in total and triggers the NER model to extract the switch information to replace the $\{deviceId\}$ in the command. The NER model successfully identifies *"San Diego"* as geographical entity and one virtual switch is found at that region through a look-up table. A complete REST API request is constructed and sent to ONOS through the northbound interface. The detailed flow information returned from ONOS is partially displayed in Fig. 2(a). Fig. 2(b) shows a Chinese example with the verbal input meaning *"Show information of all devices"* in English. No information of named entity is required in building the REST API for this case and part of the returned details of devices are shown in the format of tabular table. Based on this table, one operation example of QA in demo 2 is illustrated in Fig. 2(c). The QA model takes both verbal input *"How many devices in the network"* and the tabular data about devices obtained from the demo 1 as input. The model selects the first column of the table

Fig. 2: Illustration of demo operation for (a) command translation with English and (b) Chinese verbal input, (c) network-related question answering.

as cells which are crucial to answer the question and applies the aggregation operator COUNT to produce the answer. This demo successfully calculates the total number of devices as 14.

3. OFC Relevance

The voice user interface composed of the two demos will be presented on-site to the OFC audience. We will begin with the introduction of NLP technologies applied in the demos and network details, and then illustrate the architectures of the AI models.

In demo 1, attendees will be able to use only voice to interact with the network controlled by ONOS. We will show that attendees can easily obtain network state, switch and server details and flow information with no prerequisites about Python coding and commands used for communication with ONOS. We will show that a server connection can be established by a voice command through a wireless headset which enables remote operations and freedom of movement. Moreover, attendees can use different languages as the verbal input. This multilingual functionality can help local service providers who might not be familiar with English to provide quicker services to their customers, making their business model more agile and competitive.

In demo 2, attendees will be able to use only voice to ask network-related questions and answers will be returned by the interface through numerical reasoning. We will show that QA functionality can be time-saving for the network administrators and beneficial for autonomous network management and troubleshooting.

References

1. F. Musumeci, R. Cristina, N. Avishek, M. Irene, Z. Darko, R. Marco, and T. Massimo, “An overview on application of machine learning techniques in optical networks,” IEEE Commun. Surv. Tutor., 21(2), pp.1383-1408, 2018.
2. ONOS REST API, [online] available: https://wiki.onosproject.org/display/ONOS/Appendix+B%3A+REST+API
3. A. Zhang, Speech Recognition (version 3.8), 2007.
4. J. Devlin, M. Chang, K. Lee, K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” Proc. NAACL 2019, pp.4171–4186.
5. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. Gomez, Ł. Kaiser, I. Polosukhin, “Attention is all you need,” Advances in neural information processing systems, 30, 2017.
6. J. Herzig, P. Nowak, T. Müller, F. Piccinno, J.M. Eisenschlos, “TaPas: Weakly supervised table parsing via pre-training,” Proc. ACL 2020, pp.4320–4333.