Automation of Fast Configuration Error Diagnosis in Optical Transport Networks — Natural Language Processing is All You Need

Cen Wang,[1,*] Noboru Yoshikane,[1] Daniel Elson,[1] and Takehiro Tsuritani[1]

[1] *Photonic Transport Network Laboratory, KDDI Research, Inc., Saitama, Japan*

[*] *xce-wang@kddi.com*

Abstract: We train language models to automate the diagnosis of OTN configuration errors, and the diagnostic accuracy is up to 97.56%. We additionally demonstrate the effectiveness of the models on a real OTN system.

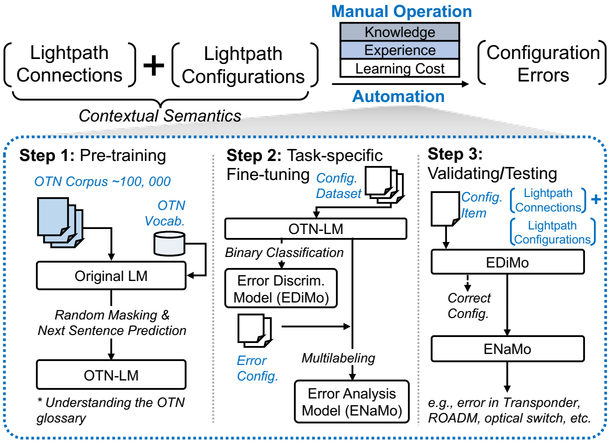
1. Introduction

Software-defined optical transport network (OTN) enables centralized control of multivendor equipment. Due to differences in control plane structures and automation levels, when configuring a lightpath, configuration errors may occur by intra-node contradiction, mismatches between ports in management and physical connections, control logic integrity defects, or unpredictable conditions during dynamic adjustment. Since commercial equipment widely uses human-comprehensible XML to represent configuration status, the process of (an administrator) retrieving status and discovering configuration logic errors (e.g., to read thousands of lines of configurations from equipment nodes with different functions) is tantamount to the process of human understanding of natural language. Manual retrieval is time-consuming, and the effectiveness of manual troubleshooting depends on the multivendor knowledge and experience of administrators, which requires learning costs.

The concept of automating simple error diagnosis has been demonstrated previously, and language model-based natural language processing (NLP) has recently achieved success in the task of understanding networking configurations [1–3]. In contrast, OTNs with emerging technologies have been widely demonstrated, diagnosis/locating of fault caused by physical layer anomalies has been actively studied [4–6]. However, in practice, the probability of configuration errors is much higher than that of physical layer failure. Considering these shortcomings, in this work, we automate configuration error diagnosis by OTN domain-specific (attention-style deep neural network based) language models (LMs). More specifically, we first use an (task-specific) LM to discriminate between correct and incorrect configurations, and then use another (task-specific) LM to find the device node where the error occurred for the incorrect configuration, and based on this, administrators can further locate error details using manual or automated logic. Moreover, we evaluate the performance of the LMs in error discrimination (97.89 in F1 score) and error analysis (85.26 in Macro-F1 score), and the diagnosis accuracy is up to 97.56%. We additionally verify the effectiveness of the proposed method through different cases on the experimental OTN system.

2. OTN Language Models

We first obtain an OTN domain-specific pre-trained LM, and fine tune this LM to task-specific LMs, to discriminate and detect configuration errors. The procedure is shown in Fig. 1. An original LM (i.e., we use *Bidirectional Encoder Representations from Transformers* (BERT), more specifically `bert-base-cased` [7] in this work) can understand natural languages for the reason that it is trained by random masking (i.e., cloze test) and next sentence prediction tasks using domain-agnostic corpus (e.g., news). **Step 1:** To make the LM understand the OTN glossary, we extend the vocabulary hold by the LM, especially, we insert important tokens such as *edfa*, *roadm* and *odu4* etc. Subsequently, we collect OTN corpus (from product manuals and Google search results with query of *OTN*, *transponder*, *wavelength selective switch (WSS)*, *reconfigurable optical add-drop multiplexer (ROADM)* and *optical switch* etc.) to pre-train the original LM and get the OTN-LM. **Step 2:** We collect and build a configuration error dataset, and the format is shown in Fig. 1 (b), where each item has configuration text (by conjugating connections and configurations of a lightpath) and its class and label. If the configuration in an item is correct, it has no further labeling. The error discrimination LM (EDiMo) is primarily trained as a binary classification task (i.e., to discriminate the configurations as correct or in error) using the dataset without considering the labels. Then, the error analysis LM (ENaMo) is additionally trained as a multilabeling task (i.e., to locate specific equipment node where the error(s) occur) considering the labels in the dataset. **Step 3:** We evaluate EDiMo and ENaMo by the items in the dataset for testing.



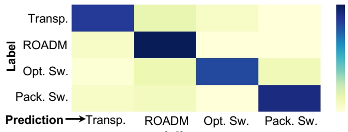
(a)

No.	Text	Class	Label
1	… (connections) [SEP] … aend 1/2/c3/et100/odu4 zend 1/2/n2/ot100/odu4 1/2/c3 plug-name QSFP28-112G-LR4-SM-LC in-service 1/2/n2 tuned frequency 195500 in-service 1/2/n2/ot100 in-service 1/2/n2/ot100/odu4 in-service … [SEP] … module 1 channel 85 start frequency 195525 end frequency 195575 in-port 4101 out-port 4201 in-service … [SEP] … ingress 1 egress 12 1 12 enabled… [SEP] …	Correct	
2	… [SEP] … transponder 1 1/2/n1 tuned frequency 195000 in-service 1/2/n2/ot100 in-service 1/2/n2/ot100/odu4 in-service … transponder 2 1/2/n1 tuned frequency 195500 in-service 1/2/n2/ot100 in-service 1/2/n2/ot100/odu4 in-service [SEP] …	Error	Transp.
3	… [SEP] … module 2 channel 84 start frequency 195475 end frequency 195525 in-port 5201 out-port 5202 in-service attenuation 0.0 … edfa 1 out-of-service edfa 2 out-of-service [SEP] …	Error	ROADM
… … … …			
n	… [SEP] … optical switch 1 ingress 1 egress 9 ingress 2 egress 10 1 2 9 10 enabled optical switch 2 no connection pair … [SEP] …	Error	Opt. Sw.
… … … …			

(b)

LM	Accuracy (%)	F1
Original	92.53	92.77
OTN-LM w/o glossary	95.14	97.59
OTN-LM	97.87	97.89

(c)



(d)

LM	Accuracy (%)	Macro-F1	MCC	CKS
Original	75.17	73.29	0.65	0.64
OTN-LM w/o glossary	81.52	80.31	0.73	0.72
OTN-LM	84.39	85.26	0.79	0.84

(e)

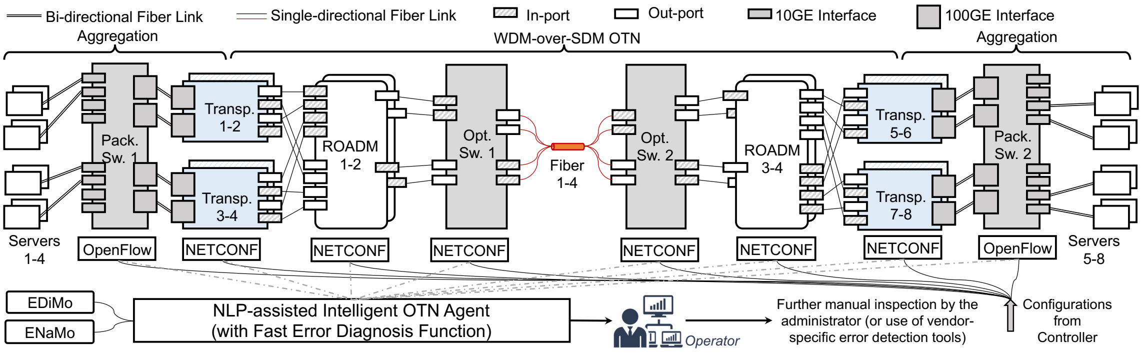
Fig. 1. (a) The procedure of configuration error diagnosis automation; (b) the format of dataset; (c) performance evaluation of EDiMo; (d) visualized confusion matrix of ENaMo; (e) performance evaluation of ENaMo.

We show the evaluation results through Fig. 1 (c) to (e), and conduct real-time validation in next section. The dataset has 4096 entries. Correct and error configurations are equally divided. Thus, 4096 entries are used for training EDiMo, and 2048 for ENaMo. For each model, the training data and testing data are 90% and 10% of the dataset, respectively. We compare the performance of OTN-LM with that of OTN-LM without OTN vocabulary (i.e., only using OTN corpus to pre-train original LM) and original LM. For the binary classification task, the accuracy and F1 scores are calculated in Fig. 1 (c). Although the performance of original LM is good, the downstream task requires more accurate classification of the upstream task. OTN vocabulary and OTN corpus successfully help the LM to understand OTN configurations. Visualized the confusion matrix is shown in Fig. 1 (d), with darker diagonal lines indicating that most of the predictions are correct. From Fig. 1 (e), in the mutilabeling task, OTN-LM outperforms the baselines in terms of accuracy, macro-F1, Matthew Correlation Coefficient (MCC) and Cohen Kappa Score (CKS). It is common for multilabeling tasks to perform worse than binary classification tasks. However, since the ENaMo outputs predicted probabilities for each label, we can report the predicted results to the administrator in order of probabilities. We find that even if the first predicted label is incorrect, the first and second predicted labels combined show an accuracy of 97.56%.

3. Experimental Validation

We verify the effectiveness of EDiMo and ENaMo in assisting locating of misconfigurations on an actual OTN system, as shown in Fig. 2 (a). In the multivendor data plane, we use 8 servers, 2 (virtualized) packet switches (Dell S4128F-ON), 8 (virtualized) transponders (ADVA FSP3000), 4 (virtualized) ROADMs (Lumentum ROADM-20) and 2 (virtualized) optical switches (Polatis). We use 4 SMFs to emulate spatial division multiplexing (SDM). In the control plane, we embed the fine-tuned EDiMo and ENaMo in our intelligent OTN agent (it is no problem to embed the models in the controller). We show 6 cases, in each case, we try to build a lightpath, the agent captures the real-time configurations and reports class and labels after model inference. The administrator finds and fixes the configurations according to the diagnosis of the models.

We report the records of the 6 experimental cases in Fig. 2 (b). It can be found for the binary classification, EDiMo successfully divides *Case 1* and other cases. For the error configurations, we capture top 2 predicted labels. Only *Case 5* required the second ranked label to correctly identify the error. In all other cases the first predicted label is sufficient. The second predicted label is correct in *Case 5*. More specifically, in *Case 1*, the configuration is correct, we retrieve the pre-FEC bit error rate (BER) 1.39×10^{-4} of *Transponder 5*, and we successfully `ping` from *Server 1* to *Server 5*. In *Case 2*, the misconfiguration is from a too high attenuation in the out-port of *ROADM 2*. We fix the attenuation according to the hint from the agent, and then we successfully `ping` from *Server 3* to *Server 7*. In *Case 3*, the error is mismatch of frequency in *Transponder 6*, which causes RX power blocking. We fix the error and successfully `ping` the corresponding servers. In *Case 4*, we fix the error client-to-network connection in *Transponder 3* according to model inference labels. In *Case 5*, we find and fix the



(a)

Case	Ground Truth	Class, Top 2 Labeling	Before Error Correction	After Error Correction
1	**Desc.** Server 1 – Pack. Sw. 1 – Transp. 1 – (195100) – ROADM 1 – Opt. Sw. 1 – (Fiber 1) -- Opt. Sw. 2 – ROADM 3 – (195100) – Transp. 5 -- Pack. Sw. 2 – Server 5, 100Gbps, VNI 108			
	Correct	Correct, <--, -->	--	BER=1.39×10^{-4}, success in *ping* from S1 to S5
2	**Desc.** Server 3 – Pack. Sw. 1 – Transp. 3 – (195000) – ROADM 2 – Opt. Sw. 1 – (Fiber 2) -- Opt. Sw. 2 – ROADM 4 – (195000) – Transp. 7 -- Pack. Sw. 2 – Server 7, 100Gbps, VNI 110			
	Too high attenuation in the out-port of ROADM 2.	Error, <ROADM, Opt. Sw.>	No light monitored in the monitoring port of ROADM 2.	Success in ping from S3 to S7
3	**Desc.** Server 2 – Pack. Sw. 1 – Transp. 2 – (195600) – ROADM 1 – Opt. Sw. 1 – (Fiber 1) -- Opt. Sw. 2 – ROADM 3 – (195600) – Transp. 6 -- Pack. Sw. 2 – Server 6, 100Gbps, VNI 109			
	Frequency mismatch in Transp. 6	Error, <Transp., ROADM>	RX power blocked in transponder	Success in ping from S2 to S6
4	**Desc.** Server 4 – Pack. Sw. 1 – Transp. 3 – (195300) – ROADM 2 – Opt. Sw. 1 – (Fiber 4) -- Opt. Sw. 2 – ROADM 4 – (195300) – Transp. 7 -- Pack. Sw. 2 – Server 8, 100Gbps, VNI 111			
	Error client-to-network connection in Transp. 3	Error, <Transp., Pack. Sw.>	No Eth. link-up between packet switch and transponder	Success in ping from S4 to S8
5	**Desc.** Server 2 – Pack. Sw. 1 – Transp. 2 – (195500) – ROADM 1 – Opt. Sw. 1 – (Fiber 3) -- Opt. Sw. 2 – ROADM 3 – (195500) – Transp. 6 -- Pack. Sw. 2 – Server 6, 100Gbps, VNI 109			
	Wrong deletion of a necessary connection of Opt. Sw. 2	Error, <ROADM, Opt. Sw.>	Low RX power in transponder	Success in ping from S2 to S6
6	**Desc.** Server 1 – Pack. Sw. 1 – Transp. 1 – (195100) – ROADM 1 – Opt. Sw. 1 – (Fiber 1) -- Opt. Sw. 2 – ROADM 3 – (195100) – Transp. 5 -- Pack. Sw. 2 – Server 6, 100Gbps, VNI 108			
	VXLAN wrong remote VTEP (of S5)	Error, <Pack. Sw., Opt. Sw.>	BER=2.06×10^{-4}, fail to *ping* from S1 to S6	Success in ping from S1 to S6

(b)

Fig. 2. (a) Experiment setup; (b) demonstration by cases.

incorrect deletion of a necessary connection in *Optical Switch 2* according to the second predicted label. In *Case 6*, we prove the OTN domain has no problem (with pre-FEC BER $= 2.06\times10^{-4}$), and we find and fix the problem in the VXLAN configurations in *Packet Switch 1* according to the hint from the agent.

4. Conclusion

In view of the high incidence of configuration errors in SDN-based OTNs, and considering that the process of error locating conforms to the natural language understanding, we implement (partial) automation of configuration error diagnosis taking the advantage of NLP. More specifically, we collect OTN vocabulary and OTN corpus to pre-train the original LM (`bert-base-cased`) and obtain OTN-LM. We then use the error configuration dataset to train OTN-LM and get EDiMo and ENaMo, and the two models are for error/correct configuration classification and error configuration locating (labeling), respectively. In evaluation, EDiMo achieves 97.89 in F1 score, and ENaMo achieves 85.26 in macro-F1 score. The top 2 predicted labels can improve the diagnosis accuracy up to 97.56%. Furthermore, we demonstrate the effectiveness of the fine tuned model on an OTN experimental system. Although we have not yet achieved complete automation (locating specific errors), locating errors to device nodes has greatly accelerated the process of manual error shooting. The proposed approach is not limited to point-to-point setups, and it would support meshed OTNs etc., after the expansion of the dataset.

References

1. H. Chen, Y. Miao, L. Chen, H. Sun, H. Xu, L. Liu, G. Zhang, and W. Wang, in *Proceedings of the ACM SIGCOMM 2022 Conference*, 2022, p. 281–297.
2. M.-T.-A. Nguyen, S. B. Souihi, H.-A. Tran, and S. Souihi, in *ICC 2022 - IEEE International Conference on Communications*, 2022, pp. 2972–2977.
3. A. S. Jacobs, R. J. Pfitscher, R. H. Ribeiro, R. A. Ferreira, L. Z. Granville, W. Willinger, and S. G. Rao, in *2021 USENIX Annual Technical Conference (USENIX ATC 21)*, 2021, pp. 625–639.
4. H. Date, T. Kubo, T. Kawasaki, and H. Maeda, *IEEE Photonics Technology Letters*, vol. 33, no. 13, pp. 649–651, 2021.
5. F. Inuzuka, T. Oda, T. Tanaka, K. Kitamura, S. Kuwabara, A. Hirano, and M. Tomizawa, *Journal of Lightwave Technology*, vol. 38, no. 9, pp. 2695–2702, 2020.
6. D. Rafique, T. Szyrkowiec, H. Grießer, A. Autenrieth, and J.-P. Elbers, *Journal of Lightwave Technology*, vol. 36, no. 7, pp. 1443–1450, 2018.
7. H. Face, "Bert base model (cased)," https://huggingface.co/bert-base-cased.