Poster: A Dataset on Peering Policies of Autonomous Systems

Martin Thodi
thdmar002@myuct.ac.za
University of Cape Town
South Africa

Josiah Chavula
josiah.chavula@uct.ac.za
University of Cape Town
South Africa

Amreesh Phokeer
phokeer@isoc.org
Internet Society
Mauritius

Abstract

This paper presents a methodology for extracting and structuring information from Internet Autonomous Systems peering policy documents using natural language processing techniques. We trained a named entity recognition model to identify and extract key entities related to peering practices. The resulting structured dataset, made publicly available, provides valuable insights into autonomous system peering requirements, preferences, and routing practices. This dataset serves as a foundation for understanding and modelling the peer selection processes of autonomous systems on the Internet. Our ongoing work focuses on developing a policy-aware approach to select peering partners based on compatibility scores derived from the extracted policy requirements.

CCS Concepts

• **Networks** → *Network management*; • **Computing methodologies** → Neural networks; • **Information systems** → **Information extraction**.

Keywords

information extraction; internet peering; autonomous systems; peering policies

ACM Reference Format:
Martin Thodi, Josiah Chavula, and Amreesh Phokeer. 2024. Poster: A Dataset on Peering Policies of Autonomous Systems. In *Proceedings of the 2024 ACM Internet Measurement Conference (IMC '24), November 4–6, 2024, Madrid, Spain.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3646547.3689667

1 Introduction

Autonomous Systems (ASes) on the Internet engage in settlement-free peering to enhance connectivity, reduce transit costs, and improve network performance by shortening traffic paths [5]. ASes interested in peering typically outline their requirements, restrictions, preferences, and routing practices in policy documents. These documents offer valuable insights into the complexities of peering practices.

However, the lack of a standardized structure for these policy documents complicates their analysis. To address this challenge, this paper presents a natural language processing (NLP) based methodology that extracts and organizes information from these documents into a structured dataset, making it readily available for analysis and modelling.

Our major contribution is the creation of this dataset, which we have made accessible to the research community [1]. The dataset is a foundational resource for understanding how ASes choose their peers. Furthermore, it serves as a basis for our ongoing work to optimize peer selection on the Internet, ultimately enhancing overall network performance.

2 Dataset Description

ASes specify peering requirements, preferences, and restrictions in their policy documents. These typically begin with general requirements, including maintaining a 24/7 Network Operations Center, keeping an up-to-date PeeringDB record, implementing Mutually Agreed Norms for Routing Security (MANRS), and using the Internet Routing Registry for routing rules.

Beyond these basic expectations, traffic volume requirements are set by some ASes. These specify minimum thresholds for Internet eXchange Point (IXP) and Private Network Interconnect (PNI) peering to justify the cost of establishing and maintaining links. Closely related to volume, some ASes also require a balance between ingress and egress traffic, expressed as an outbound-to-inbound traffic ratio.

In addition to traffic considerations, geographic coverage is important. Many ASes require peers to connect at multiple locations to distribute traffic load across regions. This requirement is often complemented by routing policies that detail accepted routes and strategies, such as best or shortest exit routing.

For ASes with more selective approaches, restrictive peering policies may be in place. These can include not peering with direct customers or customers of customers, and requiring a minimum number of transit customers. Finally, some ASes also specify whether contracts are required for peering or if paid peering options are available.

3 Information Extraction Methodology

3.1 Data Collection

ASNs seeking to peer list their details and policy document URLs in PeeringDB [6]. We used CAIDA's May 31, 2024 snapshot of the PeeringDB database [1] to fetch policy documents for 3060 ASes. We downloaded PDF and Word documents directly and used Playwright [4] to capture web pages as PDFs.

After converting PDFs to text, we preprocessed the documents by removing IXP route server policies, empty documents, error messages, non-English content, and irrelevant material. This process left us with 1276 policy documents for information extraction.

IMC '24, November 4–6, 2024, Madrid, Spain

ACM ISBN 979-8-4007-0592-2/24/11
https://doi.org/10.1145/3646547.3689667

[1] https://github.com/mthodi/asn-policies

Table 1: Description of data after preprocessing.

Used?	Why?	Count	%
No	Route Servers	183	5.98
No	Empty Files	240	7.84
No	Non English	692	22.61
No	Page Not Found	144	4.70
No	Index Page	525	17.15
Yes	No Errors	1276	41.70
-	Total	3060	100

Table 2: Named entity recognition model performance, including overall scores from pre-trained entities.

Entity	Precision	Recall	F1-Score
ASN	0.99	0.99	0.99
IRR	0.96	0.97	0.97
LOCATION	0.99	0.97	0.98
MANRS	0.85	0.89	0.87
NOC	0.99	0.97	0.98
PEERINGDB	1.0	1.0	1.0
PNI	1.0	0.98	0.99
ROUT	1.0	0.5	0.67
VOLUME	0.95	0.94	0.95
Overall	0.8	0.79	0.8

Our next iteration will include multilingual support and improved collection methods for sites with additional access requirements. Table 1 summarizes the collected data after preprocessing.

3.2 Training and Data Extraction Workflow

Our workflow began by creating a taxonomy of 10 entities to capture important words and phrases related to peering, such as traffic volume (VOLUME), PNI, routing strategy (ROUT), and geographical location (LOCATION). We sampled 100 text documents and fed each one into a pre-trained SpaCy [3] NER model, which can extract common entities like quantity, organization names, and cardinals. SpaCy is an industrial-strength NLP library in Python. Using SpaCy's Matcher rules, we identified the custom entities and added them to the document's entities to create training and evaluation examples. We manually verified these bootstrapped entities using a data annotation tool to create training and validation data used to fine-tune the model.

We fine-tuned a transition-based SpaCy model to recognize and extract our custom entities. Using a pre-trained model enabled accurate recognition with minimal training examples. We used SpaCy's large English model trained on web text with default hyperparameters [7]. Table 2 shows the model's performance in recognizing our custom entities.

After training, we used the fine-tuned model and data extraction rules to identify relevant text in the documents. The NER model enabled us to define more complex data extraction rules than simple text matching. For instance, if a document had PNI as an entity, we would check the context of the entity for a VOLUME entity to extract the volume requirements for PNI peering. Some entities, such as traffic ratio, were straightforward to extract by simply verifying their existence in the document. However, some information, such as which routes will be exchanged, was easier to extract using a combination of regular expressions and Matcher rules. Figure 1 shows recognised entities from a sample policy document.

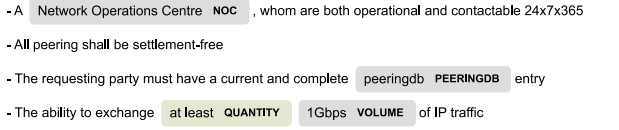

Figure 1: Sample of recognised entities from the peering policy document of FyfeWeb (AS212396) [2]

4 Conclusion

We have detailed a methodology for extracting information from peering policy documents and provided an overview of the resulting dataset. This dataset serves as a foundation for understanding and modelling the peer selection processes of autonomous systems on the Internet.

It is important to note that most ASes indicate that meeting the outlined peering requirements does not guarantee a peering agreement, nor does failing to meet some criteria necessarily result in a denial. Peering decisions are often made case-by-case, considering additional criteria such as the 'business value' expected from the peering relationship. In our ongoing work, we are developing a policy-aware approach to select AS peering partners by computing compatibility scores based on mutual satisfaction of extracted policy requirements. The proposed policy-aware framework is designed to provide explainable automated recommendations while leaving the final peering decision to human actors.

Acknowledgments

This work was financially supported by the Internet Society (ISOC) and AFRINIC, through the Computer Science Department at the University of Cape Town.

References

[1] CAIDA. 2024. PeeringDB Archive. https://www.caida.org/catalog/datasets/peeringdb/. (Accessed on 05/13/2024).
[2] FyfeWeb. 2024. FyfeWeb Peering Policy. https://fyfeweb.com/downloads/legal/peering-policy.pdf. (Accessed on 07/10/2024).
[3] Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python. (2020). https://doi.org/10.5281/zenodo.1212303
[4] Microsoft. 2024. Fast and reliable end-to-end testing for modern web apps | Playwright Python. https://playwright.dev/python/. (Accessed on 07/10/2024).
[5] William Norton. 2014. *The Internet Peering Playbook: Connecting to the Core of the Internet* (2014 edition ed.). DrPeering Press.
[6] PeeringDB. 2024. PeeringDB. https://peeringdb.com/. (Accessed on 05/13/2024).
[7] SpaCy. 2024. Training Pipelines & Models. https://spacy.io/usage/training. (Accessed on 07/10/2024).