UiO : Faculty of Mathematics and Natural Sciences
University of Oslo

Department of Informatics
Networks and Distributed Systems (ND) group

Instruction Tuning TextFlow Semi-Automatic RFCs Generation

Jie Bian

July. 04. 2025
NLDB 2025
Kanazawa, Tokyo

Abstract

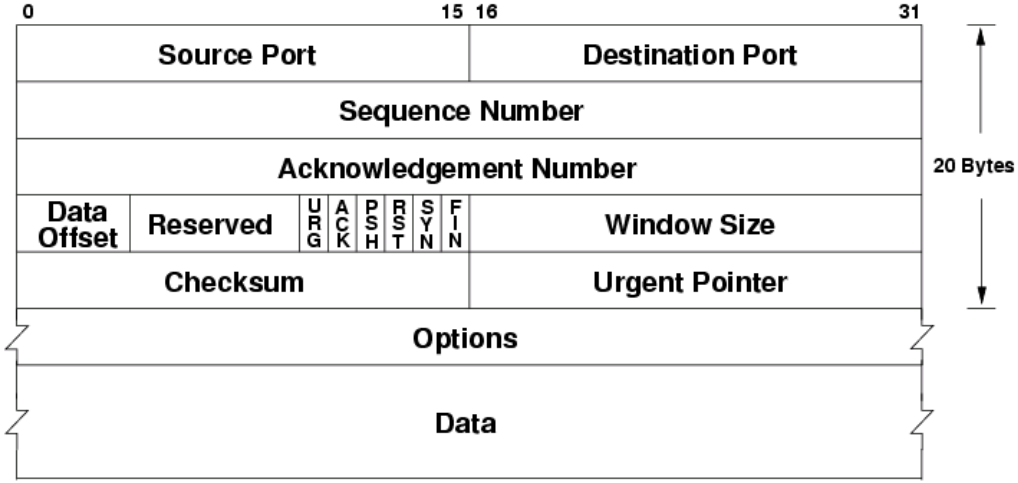
This paper explores the application of instruction tuning in generating and editing Internet Drafts (I-Ds), the preliminary versions of Request for Comments (RFCs).

Background

- IETF: Internet Engineering Task Force
  - Standardization body in charge of Internet specifications
- WG: working group
  - a group of experts formed to develop internet standards and protocols within a specific technical area.
- RFC: requeset for comments
  - formal document that describes standards, protocols, procedures, or informational content related to the internet and networking technologies.
  - RFCs are written in **natural language**, specifically **technical English**, but they often include:
    - Structured formats (like tables or diagrams)
    - Pseudocode or code snippets (for clarity)
    - **Precise definitions** (e.g., bit lengths, protocol behaviours)

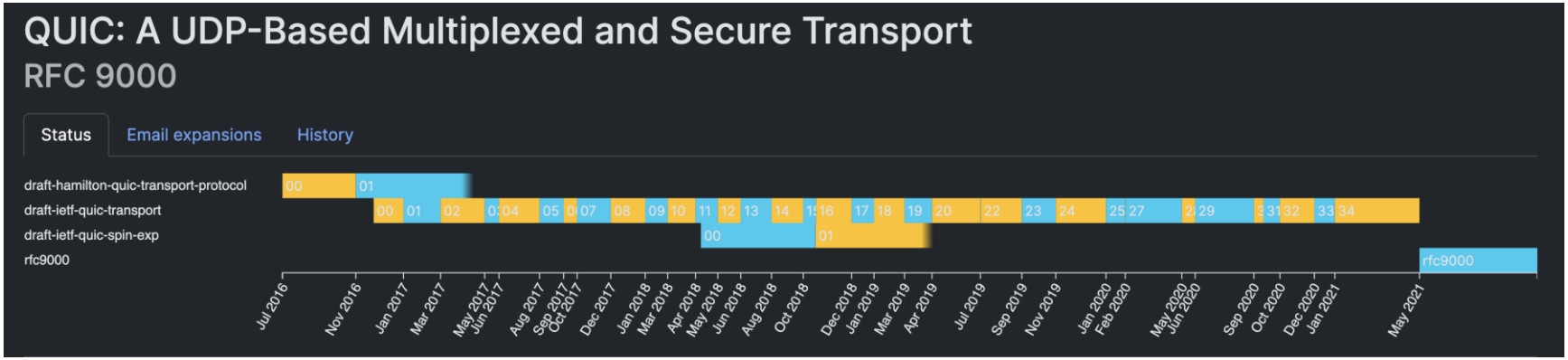
Example: TCP packet header

TCP packet header is defined in **RFC 793**, which specifies the **Transmission Control Protocol (TCP)**. For instance, when you load a webpage, your browser sends TCP packets with headers containing fields like **source port**, **destination port**, **sequence number**, and **flags** (e.g., SYN, ACK) — all structured exactly as described in RFC 793.



RFC development lifecycle

Idea and Drafting: An individual or Working Group writes an Internet-Draft (I-D) proposing a new standard or update.
Review and Discussion: The draft is reviewed and discussed within relevant IETF groups and the wider community.
Revisions: Based on feedback, the draft is revised multiple times.
Approval: If consensus is reached, the draft is reviewed by the IESG (Internet Engineering Steering Group).
Publication: Once approved, the draft is published as an RFC and becomes part of the official IETF document series.

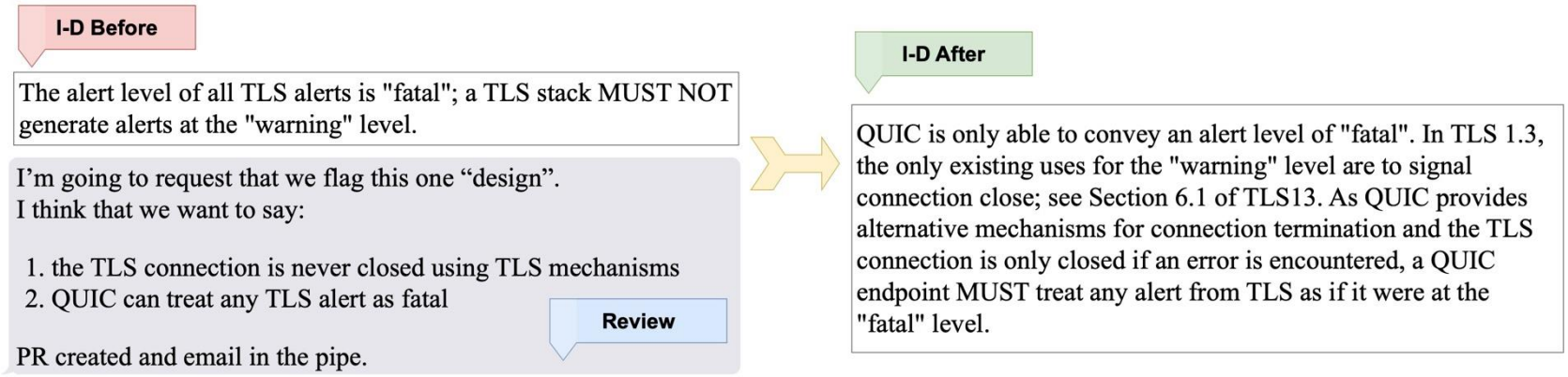


Motivation

- Success of LLM in many languages
  - **Code**, Math, Chemistry, Law, etc. ...
  - IETF?
- Long process and manual labor of publishing an RFC (Duration, e.g., from 8 months to several years)
  - Length, e.g. From 10 page to 200 pages
  - Version: e.g., from 00 to 10 and beyond
  - Authors: geologically distributed, located in different time zones
  - Communication: not real-time, mostly via text, e.g., emails and GitHub
  - Rough Consensus: majority vote, most participants agree upon that and approved by the chief editor. The consensus is reached by discussion, debate, etc.

The Internet Drafts Editing History Data set

- 160 I-Ds as training, development and test
- 41 I-Ds as OOD



The iteration of I-Ds. After a draft version is published, reviewers within the same WG will provide feedback, and the editors will modify the protocol accordingly.

Tasks

We have developed multiple tasks based on our envisioned inputs and outputs.

- **Auto Complete: I-D (before)→ I-D (after)**
- **Edit: I-D (before) + Comments → I-D (after)**

- Startup: Comments → I-D (after)
- Explain: I-D diff (before and after) → Comments

Tasks: TextFlow Startup

- Startup: Comments → I-D (after)
- Based on comments that highlight issues within the old version of an I-D, the LM is tasked with generating the revised text.
- Prompt: What is the suggested text from the feedback below?
- Difficulty: ★★★★★

Task: Explain

- TextFlow Explain: I-D diff (before and after) → Comments

- Revisions often reflect substantive design decisions rather than mere editorial changes or updates to factual information. These modifications typically involve balancing competing technical, operational, or deployment priorities—such as performance, security, backward compatibility, or implementation complexity—where trade-offs between alternatives must be carefully evaluated.

- Prompt: please analyze the following details and explain the likely reasons behind the changes made. (design rationale)
- Difficulty: ★★★★

Task: Auto Complete

- TextFlow Auto Complete: I-D (before)→ I-D (after)
- LMs have been trained on vast amounts of internet data, potentially including RFCs and I-Ds. We are interested in exploring whether these models can independently handle tasks without human feedback. This capability is similar to auto-complete or auto-repair, where the LM autonomously corrects and enhances older versions of text based solely on its own capability.
- Please revise the following text using your knowledge and understanding.
- Difficulty: ★★★

Task: Edit

- TextFlow Edit: I-D (before) + **Comments** → I-D (after)
- The LM is tasked with improving an initial version of I-Ds. Given the old text and comments that highlight issues or suggest enhancements, the model must generate a refined version of the text. This process involves understanding the feedback provided in the comments and applying it to produce a more polished and accurate description.
- Prompt: You are a professional IETF RFC writer. Identify the parts of the original text that need revision based on the feedback. Revise the text accordingly.
- Difficulty: ★★★

Preliminary Studies: Task Startup

- This task simulates the code synthesis task, where the LM generate the correct code based on an NL docstring or comments. However, unlike code, the LMs are not extensively pre-trained on the domain-specific languages.

- The LM struggled to produce outputs aligned with I-D content, as the provided feedback consisted mainly of opinions or critiques, or some keywords appeared in the original I-D text without the whole context.

Preliminary Studies: Task Explain

- LLMs like Chat-GPT will try to give explanations that seem reasonable at first glance. But after reading carefully, we found out it is hallucinating.
- LMs require domain-specific training (e.g., on the IETF's extensive corpus) to contextualize feedback and generate task-appropriate outputs.
- We simplify the task from text generation to information retrieval, where we ask the language model to find "relevant" discussions instead of generating explanations.

Main Focus

We focus on TextFlow Auto-Complete Task and TextFlow Edit Task.

- Baseline Task

  Auto Complete: I-D (before)→ I-D (after)

- Compare Task

  Edit: I-D (before) + **Comments** → I-D (after)

Models

- General
  - Mistral-7B/ Mistral-24B
  - Llama-8B/ Llama-70B
- Reasoning
  - Qwen-32B/QwQ-32B
  - Deepseek R1 distilled (Llama-8B/Qwen-32B/Llama-70B)
- Comparison
  - Llama-8B/R1(Llama-8B)
  - Qwen-32B/R1(Qwen-32B)
  - Llama-70B/R1(Llama-70B)

Instruction Tuning

- Zero-shot Configuration: The two genres of LMs
- Instruction Tuning: Mistral-7B/ Llama-8B/Mistral-24B
- maximum input length of 2048 tokens and limited the generated output to 512 tokens
- Temperature 0.1

Evaluation Challenges

- What to include (I-Ds)
  - The whole RFC or I-D: lengthy
  - The changes lines: short, lacking context information
  - The changed lines with surrounding text (text snippets)

- What to compare the LM generated I-D with
  - new text (ground truth, revised by IETF experts)
  - old text (previous version)
  - comments (suggestions, discussions by WGs)

Evaluation Challenges: Metrics

- BLEU Score: BLEU/GLEU/SacreBLEU:
  - Measure the rigid closeness
- Word Error Rate (WER):
  - Measure the differences using "distance", word-level Levenshtein
- METEOR:
  - Similar as BLEU, but also take into account additional information such as synonym, word forms, and sentence structure.
- BERT Score:
  - Semantic closeness
- Mauve:
  - Macro-level closeness
  - Open-ended text generation
  - A way to check how human-like a language model's text is by comparing it to real writing in both style and content.

Results and Analysis I (models)

- General models tend to give an output, while reasoning models tend to "think" before giving an answer.
- The scales of model sizes determine their capability to generate new versions of I-Ds, as demonstrated by Mistral-7B and Mistral-24B, Llama-8B and Llama-70B.
- Reasoning model R1(Llama-70B) is more capable than its peer Llama-70B; R1(Qwen-32B) exhibits stronger reasoning abilities compared to Qwen-32B or QwQ-32B

Results and Analysis II (two tasks)

- The LMs have very limited capability to reproduce the golden new text when only the old version is provided; Comments are crucial for LM to generate the new version of I-D.
- The LM generated I-D is generally closest to the old version (compared to new version and comments) regardless of whether Comments are included or not (Both tasks), according to our selected metrics (except Mauve).

w/o Comm (1271/1299)	ID_gen	ID_new	Llama-70B	28.42
	ID_gen	ID_old	Llama-70B	45.98
Comm (1229/1299)	ID_gen	ID_new	Llama-70B	31.99
	ID_gen	ID_old	Llama-70B	**46.49**
	ID_gen	Comm	Llama-70B	0.09

- Mauve is the most sensitive score when comments are “taken into account”, we use it to pick the best-performing model while doing instruction-tuning.

Results and Analysis III (zero-shot vs ft.)

- Our instruction-tuned models have shown remarkable performance improvements.
- To further validate the robustness of our tuned models, we applied them to OOD data. Remarkably, they exhibited the same level of effectiveness as seen with the test sets.
- The instruction-tuned Llama-8B emerged as the top performer on both the test set and OOD data.

Human Evaluation and limitation

- Those metrics only partially reflect the performance of LMs in our tasks.
- The revised text in the ground truth addressed an issue identified in earlier drafts—specifically, a scenario where a technical mechanism was prone to suboptimal behavior under certain conditions.
- While most LM-generated variants retained the observation of this issue (e.g., noting that the mechanism was more likely to occur in problematic contexts), the finalized version **went further by introducing a prescriptive guideline to prevent the issue altogether.**
- LMs often frame problems descriptively rather than proposing actionable solutions, failing to align with the ground truth's goal of resolving the issue through explicit protocol adjustments.

Future Work

- It is not essential for models to address every instance. What matters more to us is that the models typically generate revised text only when they are confident in their output.
- Our next step is to focus on generating detailed reasoning steps, or a thought process to further enhance the instruction tuning of these reasoning-specialized models.

Real Demo

Original: The public key of the account key pair, encoded as a JSON Web Key object RFC7517. <del> “valid” or “deactivated” </del> An array of URIs that the server can use to contact the client for issues related to this authorization.

Comments: what’s the difference between “deactivated” and “revoked”? Deactivated is done by the user; revoked is done by an administrator. OK, so we need to either say that or use clearer terms :) I talked with NAME offline, and I think he’s going to propose something.

Ground Truth: The public key of the account key pair, encoded as a JSON Web Key object RFC7517. <ins> The status of this registration. Possible values are: “valid”, “deactivated”, and “revoked”. “deactivated” should be used to indicate user initiated deactivation whereas “revoked” should be used to indicate administratively initiated deactivation. </ins> An array of URIs that the server can use to contact the client for issues related to this authorization.

Llama3-70B: The public key of the account key pair, encoded as a JSON Web Key object RFC7517. The status of the key pair, which can be either “active”, “user-deactivated”, or “admin-revoked” An array of URIs that the server can use to contact the client for issues related to this authorization.

Ours(Llama3-8B): The public key of the account key pair, encoded as a JSON Web Key object RFC7517. <ins> “active” or “revoked” </ins> An array of URIs that the server can use to contact the client for issues related to this authorization.

Conclusion

- This work examines the training and evaluation of LMs that follow instructions for RFC (I-D) generation tasks.
- We introduce **TextFlow**, an I-D editing history dataset, and formulate several tasks to thoroughly explore the capability of generating the next version of I-Ds with human feedback, which we refer to as semi-automatic generation.
- Additionally, we investigate the most effective **metrics** for evaluating our approach. Our work opens new avenues for exploring the incorporation of human feedback into automatic RFC generation.

Thank you!

Q & A