**Various notes for Master Thesis**

**Section about IETF:**

In the final draft, it will be a good thing to write about the communication style in IETF mailing-lists, what special words are used, how references to drafts and RFC are prevalent, explain those terms etc.

throughout the process of labeling the data, there weren't many negative samples present

this might indicate, that in general, while pointing out towards issues with proposed drafts

or documents os ubiquitous (which seems fair and normal in IETF, cause that's what participants do

there), majority of the comments are either neutral or express positive feedback / reaction to

the proposed comments (at least textually).

Terms and abbreviations:

RFC: Requests for Comments, technical documentation. Each RFC has a status, generally one of 'Internet Standard', 'Proposed Standard', 'Best Current Practice' (or 'BCP' in short), 'Informational', 'Experimental' or 'Historic'.

Once an RFC is published, it is never revised. If the specification it describes changes, the standard will be re-published in another RFC that "obsoletes" the first. If a technical or editorial error is found in an RFC, an errata may be linked to the RFC and/or held for the next document update.

The work of the IETF is to produce technical documents that define how Internet technology works in detail, and can be operated and managed at scale.

New work in the IETF begins with one or more participants producing a discussion document called an Internet-Draft (I-D) and then working to get that I-D adopted for further work.

The IETF has policies about Intellectual Property Rights (IPR) that apply when submitting I-Ds and when making other contributions to the IETF, such as by participating on mailing lists

Working Groups are organized into several areas, Application and Real Time (art), General (gen), Internet (int), Operations and Management (ops), Routing (rtg), Security (sec), and Transport (tsv), with each area overseen by several Area Directors (AD).

The Internet Research Task Force (IRTF) focuses on longer term research issues related to the Internet while the parallel organization, the Internet Engineering Task Force (IETF), focuses on the shorter term issues of engineering and standards making.

Even well-established transport protocols, such as TCP (Transmission Control Protocol) and UDP (User Datagram Protocol), which have roots that pre-date the IETF itself, continue to be extended and refined to meet the needs of applications and users of the growing global Internet.

One example is Low Latency, Low Loss, Scalable Throughput (L4S), which extends the Explicit Congestion Networking (ECN) framework to potentially lower latency by orders of magnitude.

Birds of a Feather sessions (BOFs) are initial discussions about a particular topic of interest to the IETF community. BOFs are usually held at IETF meetings, and requests to hold a BOF must be made significantly in advance of the meeting.

Transport Layer Security protocol, TLS

Topic of interests in IETF: Automated Network Management, IoT, New transport technology, Security & Privacy

RFCs produced by the IETF cover many aspects of computer networking. They describe the Internet's technical foundations, such as addressing, routing, and transport technologies. RFCs also specify protocols like TLS 1.3, QUIC, and WebRTC that are used to deliver services used by billions of people every day, such as real-time collaboration, email, and the domain name system.

Statistics: https://www.ietf.org/about/open-records/

**Statistics about IETF:**

FOR MORE STATISTICS, WE SHOULD GROUP WORKING GROUPS BY AREAS AND, AGGREGATE STATISTCS ACROSS EACH AREA AND PRESENT IT IN THE FINAL DRAFT

**Future work:**

For now at least, it seems like there are not many very negative comments on IETF. Most of

them are discussions/pointing out towards certain pros and cons of proposed drafts.

While we will still perform SENTIMENT analysis on IETF data, it is probably not a bad idea

to keep in mind that a different classification task can be performed on IETF - with defining

classes for comments in the context of drafting internet protocols - proposals, disagreement, etc.

Given this simple scheme, there seems to be a prevailing amount of discussion, which

ultimately follows the adoptions of the drafts.

It is not often (at least, given our data) that the draft/proposal is straigtly denied. It

is expressed in terms of pointing out on weaknesses/strong sides of the proposal in different

use scenarios. That is why there might be a need to potentially perform a different classification

analysis, defining more proper classes or labels, which can be a big contribution in the

future work section, if it needs to be included.

Another area of investigation could also be exploring the effects of different granularity.

Sometimes, comments taking out of the message thread become slightly hard to interpret, since

they are lacking bigger context. Changing granularity to something bigger, such as classifying

the sentiment of the whole/parts of the thread could be interesting to explore, as this would

potentially capture the overall sentiment expressed towards a certain proposal.

**Prompts used to generate text with help of ChatGPT**

Please act as a message generator for the Internet standards development or Internet Engineering Task Force domains. Your task is to generate 10 VERY different messages mimicking the style of IETF and act as an Internet standards developer. The texts must heavily criticize the proposed drafts and adoption of changes. The texts must also contain very precise and rich technical details that argue for the criticism. You can include code samples and numbers. Don't include any greetings or formalism. Act as if you are in the middle of a message thread and talking directly to the author of the proposed document/draft. The texts should not be too long. Don't forget to say that you object to or do not support the proposed changes, draft, or document. Refer to relevant sections and draft names. All the text should be lowercase.

----------------------------------------------------

Generate 10 more different samples, make them shorter.

Generate 5 candidate prompts that can be fed to a language model, so that the model is tuned to perform sentiment analysis. The prompt should specify that the model should output a score from -1 to 1, where -1 indicates negative sentiment, 0 indicates neutral, and 1 indicates positive sentiment. Make sure that the prompt includes the role that the model should play. Don't generate any examples, just the prompt.

Sure, here are five candidate prompts for sentiment analysis:

1. Role: Language model, analyze sentiment.

Prompt: "Given the following text, assess the sentiment on a scale from -1 to 1, where -1 indicates a negative sentiment, 0 indicates neutral, and 1 indicates positive sentiment."

2. Role: Sentiment analysis model.

Prompt: "Evaluate the sentiment of the following sentence, providing a score between -1 and 1, where -1 represents negative sentiment, 0 is neutral, and 1 indicates positive sentiment."

3. Role: Sentiment evaluator.

Prompt: "Assess the sentiment of the provided text, outputting a numerical score from -1 (negative) to 1 (positive), with 0 representing a neutral sentiment."

4. Role: Sentiment analysis AI.

Prompt: "As an AI trained in sentiment analysis, analyze the sentiment of the following statement and provide a score ranging from -1 (negative) to 1 (positive), with 0 denoting a neutral sentiment."

5. Role: Sentiment assessment model.

Prompt: "Your task is to evaluate the sentiment of the given text and output a sentiment score between -1 and 1, where -1 signifies a negative sentiment, 0 indicates neutral, and 1 represents positive sentiment."

**Notes for preprocessing**

1. Whenever "on [date] [person] wrote" pattern is present in a foreign language,

the parser fails to identify it as a "link" to the original message, and keeps

the whole thread of messages (or until a valid "on [date] [person] wrote" pattern is

encountered.

2. For the messages, where a "on [date] [person] wrote" pattern is written with

in foreign language, but the thread is prepented with ">" chars., it is possible

to just remove everything after that character. This essentially will remove

the thread, though "on [date] [person] wrote" in foreign language will remain.

3. Some of the messages do not include the latest reply on top of the message body:

instead, "on [date] [person] wrote" pattern (or something similar, like

"[person] [email] writes") is the first thing in the message body, followed

by original messages prepended with a ">" char, alternating with the latest reply.

Thus it could look something like this:

"[person] [email] writes"

> [original message line nr. 1]

> [ ... ]

> [original message line nr. n]

[latest reply]

or

"[person] [email] writes"

> [original message line nr. 1]

[latest reply line nr.1]

> [ ... ]

> [original message line nr. n]

[latest reply line nr. n]

The problem with this, is that the mail-parser removes everything after

"[person] [email] writes", extracting only that single line.

(THOUGH THIS IS NOT ALWAYS THE CASE: PARSER DIDN'T

RECOGNIZE "On Fri, 2022-01-14 at 00:07 +0000, Greg White wrote:".

IS IT BECAUSE OF THE TIME FORMAT?)

Another thing is that some messages do actually include the latest reply on

top, but only part of it (e.g. greetings), and the proceed with the structure

described above. This means that everything after that one line gets removed,

though some parts of the latest reply are included in the text that gets removed.

4. It also seems like the mail-parser doesn't recognize the pattern like

this:

"On Dec 13, 2021, at 12:29 PM, Ira McDonald <blueroofmusic@gmail.com> wrote:",

Thought, the mail-parser managed to detect

"On Mon, Dec 13, 2021 at 2:55 PM Laurence Lundblade <lgl@island-resort.com <mailto:lgl@island-resort.com>> wrote:",

inside that message and removed everything after it.

The differece is that the latter starts with a what weekday it is (Mon in this case).

Parser also fail to detect this pattern as well:

"On Thu, Mar 10, 2022 at 03:20:43PM -0500, Rafael Misoczki wrote:"

Here, the time is displayed in a different format.

It didn't parse this either:

"On 2021-04-28 8:52, Stefan Santesson wrote:"

but parsed this

"> On 2021-04-28 04:29, Bret Jordan wrote:". Even though there is a ">" char before

the date, this chars were also present on the lines that came before, the

reason for which is unknown.

5. In general, the email parser does an alright job, cause it also helps

to remove not only the original message thread, but some parts of the latest reply as well:

for example, it removes any additional information after something like [Thanks, [person name]].

6. We have to deal with encrypted messages. One way is to somehow detect them and just remove them

(e.g. with the help of regular expressions). One other way is to to detect them, decrypt them back

to the original message, and keep them. The problem with the latter method is that we do not

know how the structure of the encrypted message is going to end up looking like. It could be

noisy, contain something that we would rather not include etc.

7. Some of the original messages are unfortunately not prepended with a ">" char. This

seems to be a rare occasion though.

8. It also appears, that some of the messages have very interesting line breaks,

as for example if the message doesn't fit on one line, the forst line gets prepended

with a "=" char at the end, while the message continues on the next line.

Example:

"Example mess=

age."

10. It is also important to remember to remove greetings at the beginning and farewells at the end.

One way would be to (1) preserve the structure of the emails, that is, do not remove line separators.

This way we could seperate out lines, (2) identify wheteher they contain greetings/farewells, and

(3) replace them with an empty string in case they do.

Greetings:

=====================

"Hi, [name]"

"Dear [name]"

"Hey, [name]"

"Hello, [name]"

"Thanks / Thank you [name]"

NB: Names are not always included (ex.: Hi.)

Farewells:

=====================

"Best Regards, CH

[Name]"

"Thanks, CH

[Name]"

"Regards, CH

[Name]"

"Kind Regards, CH

[Name]"

"Thank you, CH

[Name]"

"-[Name]" or "--[Name]" (whitespaces/NEWLINE can be included in between) CH

"Yours Irrespectively, CH

[Name]"

"All the best, CH

[Name]"

"[WG Name] Chair(s), [Name/names]" WAIT

"Yours, CH

[Name]"

"Best, CH

[Name]"

"Thanks again, CH

[Name]"

"Cheers, CH

[Name]"

"Ciao

[Name]"

"R's,

[Name]"

NB: Between the farwell and the name there usually is a white space, but not

always. Names can also be prepented with one or more -.

Some farewells/greetings include just a name, such as "Marco,", which could very hard to identify.

We could potentially check whether the first line contains only one word, and that word is in a set

of some names, but that is a meh of a solution.

One way to solve this could potentially be to check, whether the line before

the name is an empty one, possibly in addition to checking whether the following

line is also an empty one (because it usually is present there). Remeber that

farewell can end with Name + Last Name

It could also be benefitial to remove everything (including) farewells, as original

messages not prepended with ">" will also be removed. This won't always work though.

It is also important to remember that after farewells, original messages might

not be removed by parser. This essentially means that farewells will not always

be last, and will make sense to remove everything (including the farewell itself)

after it.

11. Some messages contain the original message that starts with

------------------原始邮件------------------ (or any other language).

If the text following this line is not prepended with a ">" char.,

it is being kept fully. Thus, some regex detecting this pattern

and removing it in addition to everything after it might be useful.

12. Some messages, probably specific for IETF have the following structure:

"Some text, that explains the guidelines on how to respond. This, if I

am not mistaken, is there called 'introductory paragraph'"

This is essentially not informative and can this be removed. The piece

of text is then followed by:

----------------------------------------------------------------------

DISCUSS:

----------------------------------------------------------------------

Some discussion here. I am not yet sure whether keeping it will make sense or not.

It probaly makes sense to first look and see wheter this part might be repetetive, i.e.

contained in other message as well.

The discussion part is then followed by:

----------------------------------------------------------------------

COMMENT:

----------------------------------------------------------------------

(UPPER LINE GETS SOMETIMES REMOVED BY THE PARSER FOR SOME ODD REASON)

Which contains a comment. Comment actually seems to be pretty relevant here.

It might be a good idea to develop a regex/code that will extract this last

part.

13. Some messages are announcements, such as

"A new Request for Comments is now available in online RFC libraries.",

followed by a bunch of information on status/author/title/date.

"A New Internet-Draft is available from the on-line Internet-Drafts directories.

This draft is a work item of the Babel routing protocol WG of the IETF."

"Michael Scharf has requested publication of draft-ietf-tcpm-rfc793bis-23

as Internet Standard on behalf of the TCPM working group."

"A new meeting session request has just been submitted by Theresa Enghardt,

a Chair of the taps working group."

"Christian Hopps has requested publication of draft-ietf-lsr-isis-rfc5316bis-02

as Proposed Standard on behalf of the LSR working group."

This mails bodies really just contain technical details that aren't suitable for

sentiment analysis. Thus, they should probably be removed.

Reviewer: Russ Housley

Review result: Ready

14. One problem is also that lines in original messages are prepended with ">" while

other aren't.

15. Points of notice regarding encrypted messages:

They only contain alphanumeric chars, no white spaces, not punctuation

Thus, we could potentially look for an n number of lines,

and if all these n number of lines do abide to the rules above,

they it is highly likely that the message is encrypted

16. It doesn't seem that the mailing list contain a lot of negative sentiment,

at least not the expressed very explicitly. This begs the question of

what sentiments should we look for then? Postive, neutral? Slightly more

fine-grained?