# Annotation guidelines for the task of Sentiment Analysis of IETF mailing lists

This is annotation guidelines for the task of Sentiment analysis of IETF mailing list data. In our study, we chose to focus on labels including 3 polarities: **positive, neutral,** and **negative**.

IETF mailing lists comprise discussions about drafting internet protocols, their maintenance and further development/improvement, deliberations of proposed new adaptations and/or protocols. Thus, the nature of communication in IETF is to a great degree based on writing comments that address certain technicalities of proposed changes. This can range from discussing how this technicalities would fit and operate with current implementations to whether it would make sense to adopt such changes at all. This could be messages including wordings like: *"I believe Section X does not address issue, Y, but overall the premise of the proposed changes could be beneficial…"* or *"I like the idea of the draft, but I have a few comment concerning…"*. Essentially, this means that people address different pros and cons of proposed changes. This is important for defining the labeling scheme of such data, e.g. we don't want to label messages where authors point out potential disadvantages/cons/issues with proposed changes as **negative.** Because that is what people of IETF do - writing technical comments and discuss work.

Thus, we label our data following these guidelines:

1. **Positive**.
- If author of the message clearly state the he/she is in favor of adopting the draft/work;
- If author shows gratitude for the comments received;
- If author states that received information is/was useful and/or helpful;

In general, messages that should be labeled as **positive** are supportive of adopting proposed changes, even if including comments, pointing out what can be addressed. Moreover, given that discussions on IETF occasionally go beyond internet protocols drafting, messages containing generic information, clearly conveyed with a positive tone are also considered **positive.**

Examples of **positive** samples:

| |
|---|
| *i support the adoption of the draft. it provides a suitable solution for the very relevant topic of inter-domain constrained path setup, which will become even more relevant in the next future.* |
| *im pleased to see you respond to this call. i hope others will review this draft regarding inbound max prefix.* |
| *minimal drama :-) great!* |
| *as an individual, i strongly support adoption of a draft along these lines.* |
| *good news indeed! thank you and best regards, reto* |

## 2. **Neutral**

- If author of the message discusses technicalities;
- If author of the message points out different aspects of proposed changes, whether they would be in the favor of the proposed changes or not, **WITHOUT** explicitly stating whether he/she support the proposed changes
- If author objectively criticizes proposed changes, backing his/hers arguments with factual information, without attacking the author, and pointing out issues and how to potentially avoid them
-  If author asks a question(s) about technicalities
-  Announcements
-  General discussion

In general, neutral samples are the easiest to detect, as they are prevalent in such data. They usually comprise messages with discussions, whether this discussions would be technical or general, lacking sentiment.

Examples of **neutral** samples:

| |
|---|
| *the dpop spec currently defines how to obtain a dpop-bound token via token endpoint invocations (namely, authorization_code and refresh_token grants). but it is also possible to obtain access token prior to code-to-token exchange, via oauth implicit/hybrid flows. do we have any plans to support dpop in authorization endpoint (in addition to token endpoint) and implicit/hybrid flows? is yes, what it might look like? a "dpop" request parameter or a "dpop" header?* |
| *right, i was thinking of user=phone. phone-context and the other parameters are removed when applying the first bullet point in [link]* |
| *rtc and manual input is ok, but unsecured ntp wouldn't be acceptable as it is trivially spoofed in a mitm attack. starting with roughtime is certainly better than an unauthenticated ntp source. iirc it relies on long-term trusted keys, no pki involved. you could do the same with nts using self-signed certificates.* |
| *this version of the draft contains following updates: 1. security section: address comment: "man-in-the-middle" (mim) might break the assumption "normally" and must be considered. 2. iana section: add stamp tlv flag error flag d (wrong destination) 3. minor editorial changes welcome your review comments and suggestions.* |
| *i've uploaded the draft agenda: [link] please take a look and let us know if any changes are needed. presenters, please provide your slides as early as possible.* |

## 3. **Negative**

- If author heavily or harshly criticizes proposed changes
- If author clearly states that he/she is against proposed changes
- If author doesn't like someones work and explicitly expresses that
- If author personally attacks another participant
- If author shows his/hers distaste towards more generic subject

In general, when labeling a text with **negative** label, the text should clearly contain a harsh criticism, often including less factual arguments and more feelings. If the message states that author objects the adoption of the proposed changes, then we consider this a **negative** scenario, as it indicates that the proposed changes are bad and flawed. It is at times quite hard to decide whether a message should be labeled as **negative** or **neutral**, because they might be quite factual, and, while not explicitly revealing distaste, have a negative tone.

Examples of **negative** samples:

| *i strongly object to the adoption of the draft. there are 3 different flavors defined in the draft and all three flavors have significant difference in the forwarding plane behaviours. i would prefer the discussion on whether wg wants to work on all these flavors or only one of them to precede the adoption.* |
| --- |
| *okay. i didn't mention the fence as an endorsement of his personality or politics. not sure why that's relevant. generally, i think arguing that something is wrong and should be changed without understanding why it's the way it is isn't a great way to go.* |
| *you are not improving convergence. you are propagating liveness. the fact that this relates to convergence in the overlay is irrelevant to the igp. you are not retaining scalability. you are damaging it. you are proposing flooding a prefix per router that fails. if there is a mass failure, that would result in flooding a large number of prefixes. the last thing you want when there is a mass failure is additional load, exacerbating the situation. you are signaling the (lack of) liveness of a remote node. i propose that we instead use existing signaling mechanisms to do this. multi-hop bfd seems like an obvious choice. if you greatly dislike that for some reason, i would suggest that we create a proxy liveness service, advertised by the abr. this would allow correspondents to register for notifications. the abr could signal these unicast when it determines that the specific targets are unreachable.* |
| *it takes an awful lot here for me to bother to reply to a thread. here is a better algorithm to find the truth. purchase any of dozens of commercial routers today running cake or fq_codel, or the thousands available via reflash to openwrt/dd-wrt/tomato construct a repeatable experiment, and publish the code and results. the results from running, repeatable, code trumps theoretical objections every time. the evenroute v3 and edgerouter x series are pretty good bases for experimentation. wilful ignorance, and the lack of a willingness to construct repeatable experiments is not science. if you have a point to make, make it with a repeatable experiment against running code, please. tuning out again.* |
| *creating lots of special rules makes me feel uncomfortable. is there evidence that people reduce state value spaces a lot and in isolation, [link], they just rev a module to reduce some state value spaces? /js* |