Methods

Overview

Key Concepts

Attributes & Languages

Model Cards

Methods

Limits & Errors

FAQs

There are two methods available for Perspective API:

- **`AnalyzeComment`**, where a user sends a request for a comment to be analyzed, and a score is returned
- **`SuggestCommentScore`**, where a user can suggest a better score for a comment

Scoring comments: AnalyzeComment

To send a comment scoring request to the API, post a request object to this endpoint in your code:
`https://commentanalyzer.googleapis.com/v1alpha1/comments:analyze`

See example in Sample Requests.

AnalyzeComment request

Copy

```
{
  "comment": {
    "text": string,
   "type": string
 },
 "context": {
   "entries": [{
     "text": string,
     "type": string
    }]
  },
  "requestedAttributes": {
    string: {
      "scoreType": string,
      "scoreThreshold": float
    },
  },
  "languages": [string],
  "doNotStore": bool,
  "clientToken": string,
  "sessionId": string,
}
```

Field	Description
`comment.text`	(required) The text to score. This is assumed to be utf8 raw text of the text to be checked. Emoji and other non-ascii characters can be included (HTML will probably result in lower performance).
`comment.type`	*(optional)* The text type of comment.text. Either "PLAIN_TEXT" or "HTML". Currently only "PLAIN_TEXT" is supported.
`context.entries`	*(optional)* A list of objects providing the context for comment. The API currently does not make use of this field, but it may influence API responses in the future.
`context.entries[].text`	*(optional)* The text of a context object. The maximum size of context entry is 1MB.
`context.entries[].type`	*(optional)* The text type of the corresponding context text. Same type as comment.text. Currently only "PLAIN TEXT" is supported.
`requestedAttributes`	(required) A map from attribute name to a configuration object. See the 'Attributes and Languages' page for a list of available attribute names. If no configuration options are specified, defaults are used, so the empty object {} is a valid (and common) choice. You can specify multiple attribute names here to get scores from multiple attributes in a single request.
`requestedAttributes[name].scoreType`	*(optional)* The score type returned for this attribute. Currently, only "PROBABILITY" is supported. Probability scores are in the range [0,1].
`requestedAttributes[name].scoreThreshold`	*(optional)* The API won't return scores that are below this threshold for this attribute. By default, all scores are

	threshold for this attribute. By default, all scores are returned.
`spanAnnotations`	*(optional)* A boolean value that indicates if the request should return spans that describe the scores for each part of the text (currently done at per-sentence level). Defaults to false.
`languages`	*(optional)* A list of ISO 631-1 two-letter language codes specifying the language(s) that comment is in (for example, "en", "es", "fr", "de", etc). If unspecified, the API will auto-detect the comment language. If language detection fails, the API returns an error. Note: See currently supported languages on the 'Attributes and Languages' page. There is no simple way to use the API across languages with production support and languages with experimental support only.
`doNotStore`	*(optional)* Whether the API is permitted to store comment and context from this request. Stored comments will be used for future research and community attribute building purposes to improve the API over time. Defaults to false (request data may be stored). Warning: This should be set to true if data being submitted is private (i.e. not publicly accessible), or if the data submitted contains content written by someone under 13 years old (or the relevant age determined by applicable law in my jurisdiction).
`clientToken`	*(optional)* An opaque token that is echoed back in the response.
`sessionId`	*(optional)* An opaque session ID. This should be set for authorship experiences by the client side so that groups of requests can be grouped together into a session. This should not be used for any user-specific id. This is intended for abuse protection and individual sessions of interaction.
`communityId`	*(optional)* An opaque identifier associating this comment with a particular community within your platform. If set, this field allows us to differentiate comments from different communities, as each community may have different norms.

Note that only `comment.text` and `requestedAttributes` are required.

AnalyzeComment response

Copy

```
{
  "attributeScores": {
    string: {
      "summaryScore": {
        "value": float,
        "type": string
      }
      "spanScores": [{
        "begin": int,
        "end": int,
        "score": {
          "value": float,
          "type": string
        }
      }]
    }
  },
  "languages": [string],
  "clientToken": string
}
```

Field	Description
`attributeScores`	A map from attribute name to per-attribute score objects. The attribute names will mirror the request's `requestedAttributes`.
`attributeScores[name].summaryScore.value`	The attribute summary score for the entire comment. All attributes will return a `summaryScore` (unless the request specified a `scoreThreshold` for the attribute that the `summaryScore` did not exceed).
`attributeScores[name].summaryScore.type`	This mirrors the requested `scoreType` for this attribute.
`attributeScores[name].spanScores`	A list of per-span scores for this attribute. These scores apply to different parts of the request's `comment.text`. Note: Some attributes may not return `spanScores` at all.
`attributeScores[name].spanScores[].begin`	Beginning character index of the text span in the request comment.
`attributeScores[name].spanScores[].end`	End of the text span in the request comment.
`attributeScores[name].spanScores[].score.value`	The attribute score for the span delimited by begin and end.
`attributeScores[name].spanScores[].score.type`	Same as `summaryScore.type`.
`languages`	Mirrors the request's languages. If no languages were specified, the API returns the auto-detected language.
`clientToken`	Mirrors the request's `clientToken`.

Here is a sample request for the `TOXICITY` and `UNSUBSTANTIAL` attributes for an English comment:

AnalyzeComment example

Copy

```
// Request
{
  "comment": {
     "text": "What kind of idiot name is foo? Sorry, I like your name."
  },
  "languages": ["en"],
  "requestedAttributes": {
    "TOXICITY": {},
    "UNSUBSTANTIAL": {}
  }
}
```

The sample response contains the TOXICITY and UNSUBSTANTIAL attribute scores. Each attribute has a single overall summaryScore as well as two spanScores.

Both attributes return the same spans in this case: the span [0,31) (corresponding to "What kind of idiot name is foo?") and the span [32,56) (corresponding to "Sorry, I like your name."); the span notation refers to string indices. However, attributes may not always return the same spans.

Example response

```
// Response
{
  "attributeScores": {
    "TOXICITY": {
      "summaryScore": {
        "value": 0.8627961,
        "type": "PROBABILITY"
      }
    },
     "UNSUBSTANTIAL": {
       "spanScores": [
         {
           "begin": 0,
           "end": 31,
           "score": {
             "value": 0.52690625,
             "type": "PROBABILITY"
           }
         },
         {
           "begin": 32,
           "end": 55,
           "score": {
             "value": 0.9106685,
             "type": "PROBABILITY"
           }
         }
       ],
       "summaryScore": {
        "value": 0.69036055,
        "type": "PROBABILITY"
      }
    }
  },
  "languages": [
    "en"
  ]
}
```

Sending feedback: SuggestCommentScore

The SuggestCommentScore endpoints lets you give the API feedback by allowing you to suggest a score that you think the API should have returned. You can use this method if you disagree with a score and would like to improve the attribute. All submissions to SuggestCommentScore are stored and used to improve the API and related services. This method should not be used for private data (i.e., for data that is not accessible publicly), or if the data submitted contains content written by someone under 13 years old (or the relevant age determined by applicable law in my jurisdiction).

POST https://commentanalyzer.googleapis.com/v1alpha1/comments:suggestscore

SuggestCommentScore request Copy

```
{
  "comment": {
    "text": string,
    "type": string
  },
  "context": {
    "entries": [{
      "text": string,
      "type": string
    }]
  },
  "attributeScores": {
    string: {
      "summaryScore": {
         "value": float,
         "type": string
      },
      "spanScores": [{
        "begin": int,
        "end": int,
        "score": {
           "value": float,
           "type": string
        }
      }]
    }
  },
  "languages": [string],
  "communityId": string,
  "clientToken": string
}
```

Field	Description
`comment`	(required) Same as AnalyzeComment request.
`context`	*(optional)* Same as AnalyzeComment request.
`attributeScores`	(required) Similar to AnalyzeComment response. This holds the attribute scores that the client believes the comment should have. It has the same format as the attributeScores field in the AnalyzeComment response, as it's what the client believes is the correct "answer" for what this comment should be scored as. The client can specify just summary scores, just span scores, or both.
`languages`	*(optional)* Same as AnalyzeComment request.
`communityId`	*(optional)* Same as AnalyzeComment request.
`clientToken`	*(optional)* An opaque token that is echoed back in the response. (Note: This is not the clientId, which is automatically set through Google Cloud. This is a field users can set to help them keep track of their requests.)

SuggestCommentScore response

```
{
  "clientToken": string
}
```

Field	Description
`clientToken`	Mirrors the request's clientToken.

SuggestCommentScore example

Description

Copy

```
// Request
{
  "comment": {
    "text": "I guess it comes down a simple choice: Get busy living, or get busy dyi
  },
  "attributeScores": {
    "TOXICITY": {
      "summaryScore": {
        "value": 0
      }
    },
  },
  "communityId": "/forum/movies",
  "clientToken": "comment-53922"
}
The response is not particularly interesting:
// Response
{
  "clientToken": "comment-53922"
}
```

← Previous: Model Cards

Next: Limits & Errors →

Perspective | Support

Providing support and resources to enable developers to build and experiment with Perspective API.

ABOUT

Perspective Developers

Perspective API

Jigsaw

RESOURCES

About the API

Documentation

Contact Us

CONNECT

API Updates

False Positive Blog

Jigsaw Blog