TALK GUIDELINE:

1. Definition
2. Score Interpretation
3. Characteristics
4. Prerequisites
5. Enable API
6. Applications and Examples

Definition

Perspective API is a free API that uses machine learning models to score the attributes (emotional concepts) of a comment.

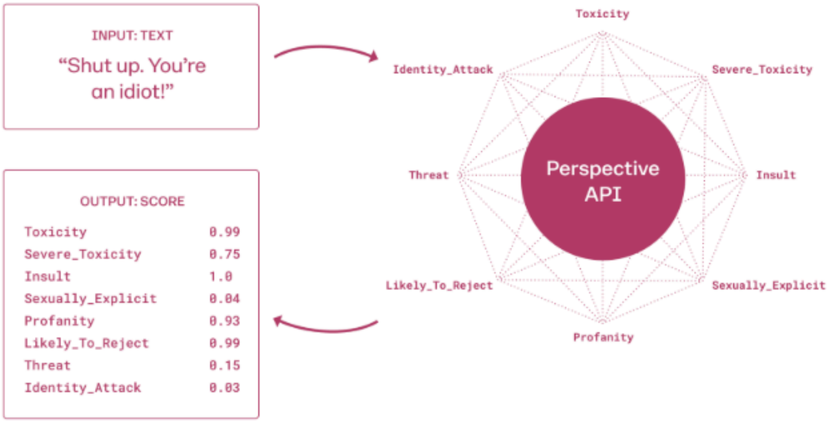
When you send a request to the API, you'll send the text of a single comment and the response will contain predictions about the perceived impact that comment may have on the conversation.

A comment could be a single post to a web page's comments section, a forum post, a message to a mailing list, a chat message, etc.

From the lens of NLP, Perspective API is a machine learning model that leverages word-embedding techniques to build representations of words as vectors in a high-dimensional space, in which a metric distance should reflect the conceptual distance among words, therefore providing linguistic context.

Input: Text

Output: A probability score between 0 and 1. A higher score indicates a greater likelihood a reader would perceive the comment as containing the given attribute.



Toxicity	0.99
Severe_Toxicity	0.75
Insult	1.0
Sexually_Explicit	0.04
Profanity	0.93
Likely_To_Reject	0.99
Threat	0.15
Identity_Attack	0.03

(Well Defined) Emotion Attributes: Toxicity is the most popular attribute.

Model Reliability(Training Data)

Probability Score Interpretation

The only score type currently offered is a probability score.

Probability scores represent a probability, with a value between 0 and 1. A higher score indicates a greater likelihood that a reader would perceive the comment as containing the given attribute. As such, a comment with a TOXICITY score of 0.9 is not necessarily more toxic than a comment with a TOXICITY score of 0.7. Rather, it's more likely to be perceived as toxic by more readers. The score reflects the percentage of readers who would perceive the comment as toxic; a score of 0.9 indicates that 9 out of 10 readers would perceive the comment as toxic, and a score of 0.7 indicates that 7 out of 10 readers would perceive toxicity.

Other Characteristics

1. Support multiple language: English, Spanish, French, German, Portuguese, and Italian.

2.Quota limit: Check your quota limits by going to your Google Cloud project's Perspective API page, and check your project's quota usage at the cloud console quota usage page.

3.The maximum text size per request is 20 KB. One character does not necessarily equal one byte, as different characters have different encodings. Note that models are trained on online comments, so performance will be best on text around that length. Read the W3C guide on character encoding.

4. Perspective API is hosted on the Google Cloud Platform and is open to any programming language.

Prerequisites

1. Have a Google account for accessing to the Google Cloud
2. Use the Google Cloud console to set a Google Cloud Project (or use any your existing ones; please follow the steps from #2 to #3)
3. Fill the form to request API access (usually replied within 1h)

*Google cloud is a set of cloud computing services that runs on Google's infrastructure.

Enable API and Run the Code in Python

We can enable the API either from the command line or the Google Cloud console.

Cloud Console

1.Navigate to the Perspective API overview page and click "Enable".

2.Generate API key: Navigate to Google API credentials page and click "Create credentials".

3.Copy and securely save your key.

Below is a python scrpit using a sample from the Google API python Client Libraries.

[terminal] source pspapi/bin/activate
[kernel] pspapi

[1]:

```python
from googleapiclient import discovery
import json
```

[7]:

```python
API_KEY = 'fill_in_your_key'

client = discovery.build(
  "commentanalyzer",
  "v1alpha1",
  developerKey=API_KEY,
  discoveryServiceUrl="https://commentanalyzer.googleapis.com/$discovery/rest?version=v1alpha1",
  static_discovery=False,
)

analyze_request = {
  'comment': { 'text': 'Hello, world!' },
  'requestedAttributes': {'TOXICITY': {}}
}

response = client.comments().analyze(body=analyze_request).execute()
print(json.dumps(response, indent=2))
```

```
{
  "attributeScores": {
    "TOXICITY": {
      "spanScores": [
        {
          "begin": 0,
          "end": 13,
          "score": {
            "value": 0.01847211,
            "type": "PROBABILITY"
          }
        }
      ],
      "summaryScore": {
        "value": 0.01847211,
        "type": "PROBABILITY"
      }
    }
  },
  "languages": [
    "en"
  ],
  "detectedLanguages": [
    "en",
    "fil"
  ]
}
```

Applications and Examples

Industry (moderation):
NYTimes, Coral, Iscourse, Wordpress, etc.

Academia (identification):
Avalle, M., Di Marco, N., Etta, G., Sangiorgio, E., Alipour, S., Bonetti, A., ... & Quattrociocchi, W. (2024).
Persistent interaction patterns across social media platforms and over time. Nature, 628(8008), 582-589.

Lastly, some kind reminders:

No model is perfect and will make errors. It will be unable to detect patterns of toxicity it has not seen before.

Because of this, perspective is not intended for use cases such as fully automated moderation. Thus, Perspective can not completely replace human moderation. However, we can use perspective for several purposes: human-assisted moderation(time saver), authorship feedback; read better comments...

Personal thoughts on its usage for academic research:

*** Using more than one classifiers for triangulation to enhance robustness. Some tools sharing the similar function with Perspective API: HateBERT(PLM), Detoxify, and IMSYPP...