

要約API
仕様書
Ver 1.0
2021.10.26

一般社団法人AI-Lab

このAPI は、文章を要約するAPIです。
 圧縮方式/パラメータを指定して要約を行います。
 要約結果の他に、需要と判定された文章、キーワード出現率を出力する事ができます。

- URL
 https://api.ai-tanteki.com/V1/tanteki

2. リクエストヘッダ

KEY	型	省略	内容
Content-Type	文字列	不可	"application/json" 固定
Authorization	文字列	不可	APIの利用に必要なトークン

3. リクエスト JSON型式

名称	KEY	型	省略	内容									
ID	id	文字列	不可	このリクエストを識別するためのID。レスポンスのIDに返す。									
テキスト	text	文字列	不可	要約対象のテキスト									
圧縮方式	compressionType	文字列	可	'パーセント' または '文字数' <table border="1"> <thead> <tr> <th>値</th> <th>意味</th> </tr> </thead> <tbody> <tr> <td>パーセント</td> <td>%で指定する</td> </tr> <tr> <td>文字数</td> <td>文字数で指定する</td> </tr> </tbody> </table> (*)省略した場合は、「%で指定する」の「50%」	値	意味	パーセント	%で指定する	文字数	文字数で指定する			
値	意味												
パーセント	%で指定する												
文字数	文字数で指定する												
圧縮方式パラメータ	compressionArg	数値	可	圧縮方式のパラメータを指定する。 この値を目安として圧縮を行います。 <table border="1"> <thead> <tr> <th>圧縮方式</th> <th>値</th> <th>省略した場合</th> </tr> </thead> <tbody> <tr> <td>パーセント</td> <td>10%~90%</td> <td>50</td> </tr> <tr> <td>文字数</td> <td>70文字~2000文字</td> <td>200</td> </tr> </tbody> </table>	圧縮方式	値	省略した場合	パーセント	10%~90%	50	文字数	70文字~2000文字	200
圧縮方式	値	省略した場合											
パーセント	10%~90%	50											
文字数	70文字~2000文字	200											
重要と判定された文章フラグ	important	bool値	可	trueの時、「重要と判定された文章」をレスポンスに含む (*)省略した場合は、false扱い。									
キーワード出現率フラグ	keyword	bool値	可	trueの時、「キーワード出現率」をレスポンスに含む (*)省略した場合は、false扱い。									

4. レスポンス JSON型式

名称	KEY	型	省略	内容												
バージョン	version	文字列	不可	APIのバージョン												
ID	id	文字列	不可	このリクエストで設定されたIDを返す。												
圧縮方式	compressionType	文字列	不可	要約で使用した「圧縮方式」を返す。												
圧縮方式パラメータ	compressionArg	数値	不可	要約で使用した「圧縮方式パラメータ」を返す。												
要約結果テキスト	compressedText	文字列	不可	要約結果のテキスト												
重要と判定された文書フラグ	important	bool値	不可	要約で使用した「重要と判定された文章フラグ」を返す。												
重要と判定された文書	importantTexts	配列	可	「重要と判定された文章フラグ」がTRUEの時、「重要と判定された文章」を、文字列の配列で返す。												
キーワード出現率フラグ	keyword	bool値	不可	要約で使用した「キーワード出現率フラグ」を返す。												
名詞キーワード出現率	keywordNouns	配列	可	「キーワード出現率フラグ」がtrueの時、「名詞キーワード出現率」をオブジェクトの配列として返す。 オブジェクトの内容は下記の通り。 <table border="1"> <thead> <tr> <th>KEY</th> <th>型</th> <th>内容</th> </tr> </thead> <tbody> <tr> <td>noun</td> <td>文字列</td> <td>名詞</td> </tr> <tr> <td>freq</td> <td>数値</td> <td>出現頻度</td> </tr> <tr> <td>ratio</td> <td>数値</td> <td>出現率</td> </tr> </tbody> </table>	KEY	型	内容	noun	文字列	名詞	freq	数値	出現頻度	ratio	数値	出現率
KEY	型	内容														
noun	文字列	名詞														
freq	数値	出現頻度														
ratio	数値	出現率														
動詞状態キーワード出現率	keywordVerbs	配列	可	「キーワード出現率フラグ」がtrueの時、「動詞状態キーワード出現率」をオブジェクトの配列として返す。 オブジェクトの内容は下記の通り。 <table border="1"> <thead> <tr> <th>KEY</th> <th>型</th> <th>内容</th> </tr> </thead> <tbody> <tr> <td>verb</td> <td>文字列</td> <td>動詞状態</td> </tr> <tr> <td>freq</td> <td>数値</td> <td>出現頻度</td> </tr> <tr> <td>ratio</td> <td>数値</td> <td>出現率</td> </tr> </tbody> </table>	KEY	型	内容	verb	文字列	動詞状態	freq	数値	出現頻度	ratio	数値	出現率
KEY	型	内容														
verb	文字列	動詞状態														
freq	数値	出現頻度														
ratio	数値	出現率														

5. エラー

名称	KEY	型	省略	内容						
エラー	error	オブジェクト	不可	エラーオブジェクト エラーは下記のオブジェクト構造を持つ <table border="1"> <thead> <tr> <th>KEY</th> <th>型</th> <th>内容</th> </tr> </thead> <tbody> <tr> <td>message</td> <td>文字列</td> <td>エラーの内容</td> </tr> </tbody> </table>	KEY	型	内容	message	文字列	エラーの内容
KEY	型	内容								
message	文字列	エラーの内容								

6. リクエストヘッダ例

```
{
  "Content-Type": "application/json",
  "Authorization": "Your Token"
}
```

7-1. リクエストレスポンス例 1

```
リクエスト
{
  "id": "0001",
  "text": "文章が理解できる人工知能〜〜略〜〜と考えております。",
  "compressionType": "パーセント",
  "compressionArg": 50,
  "important": true,
  "keyword": true
}
```

レスポンス

```
{
  "version": "1",
  "id": "0001",
  "compressionType": "パーセント",
  "compressionArg": 50,
  "compressedText": "その人工知能は人や他の人工知能と自由に~~~略~~~仕事をしてくれる社会へとつながって行きます。",
  "important": true,
  "importantTexts": [
    "文章が理解できる人工知能ができれば~~~略~~~仕事をしてくれる社会へとつながって行きます。",
    "バズグラフ社はそんなサービスを実現し提供していきたいと考えております。"
  ],
  "keyword": true,
  "keywordNouns": [
    { "noun": "その人工知能", "freq": 2, "ratio": 9.09 },
    { "noun": "バズグラフ社", "freq": 1, "ratio": 4.55 },
    { "noun": "弱者", "freq": 1, "ratio": 4.55 },
    ~ ~ 略 ~ ~
    { "noun": "それ", "freq": 1, "ratio": 4.55 },
    { "noun": "究極", "freq": 1, "ratio": 4.55 },
    { "noun": "実現", "freq": 1, "ratio": 4.55 }
  ],
  "keywordVerbs": [
    { "verb": "できれば", "freq": 1, "ratio": 5.56 },
    { "verb": "つながって", "freq": 1, "ratio": 5.56 },
    { "verb": "あります", "freq": 1, "ratio": 5.56 },
    ~ ~ 略 ~ ~
    { "verb": "実現し", "freq": 1, "ratio": 5.56 },
    { "verb": "いきたいと", "freq": 1, "ratio": 5.56 },
    { "verb": "おります", "freq": 1, "ratio": 5.56 }
  ]
}
```

7-2. リクエストレスポンス例2

リクエスト

```
{
  "id": "0002",
  "text": "文章が理解できる人工知能~~~略~~~と考えております。",
  "compressionType": "文字数"
}
```

レスポンス

```
{
  "version": "1",
  "id": "0002",
  "compressionType": "文字数",
  "compressionArg": 200,
  "compressedText": "文章が理解できる人工知能ができれば、~~~略~~~と考えております。",
  "important": false,
  "keyword": false
}
```

7-3. リクエストレスポンス例3

リクエスト

```
{
  "id": "0003",
  "text": "文章が理解できる人工知能~~~略~~~と考えております。"
}
```

レスポンス

```
{
  "version": "1",
  "id": "0003",
  "compressionType": "パーセント",
  "compressionArg": 50,
  "compressedText": "その人工知能は人や他の人工知能と自由に ~~~略~~~ とつながって行きます。",
  "important": false,
  "keyword": false
}
```

8. エラー例

```
{
  "error": {
    "message": "Your Request was Forbidden"
  }
}
```

9. pythonでのサンプル

```
import requests
import json

# URL
api_url = "https://api.ai-tanteki.com/V1/tanteki"

# Request Header
headers = {
  "Content-Type": "application/json",
  "Authorization": "Your Tanteki API Token"
}

text = '''
文章が理解できる人工知能ができれば、その人工知能は人や他の人工知能と自由に
コミュニケーションをとることができるようになり、その人工知能が自分の分身
として情報収集や取引・仕事をしてくれる社会へとつながって行きます。
'''
```

それは弱者に優しい究極のコミュニケーション社会でもあります。
バスグラフ社はそんなサービスを実現し提供していきたいと考えております。

```
postData = {  
    "id": "0001",  
    "text": text,  
    "compressionType": "パーセント",  
    "compressionArg": 50,  
    "important": True,  
    "keyword": True  
}
```

```
r_post = requests.post(api_url, headers=headers, json=postData)  
print(r_post.status_code)  
jsonText = json.dumps(r_post.json(), indent=4, ensure_ascii=False)  
print(jsonText)
```