LevelFish

PARAPHRASE API GUIDE

May 2021

Table of Contents

Overview:

The LevelFish Paraphrase API allows you or your customers to generate multiple paraphrases from a single sentence. Designed to provide multiple choice options for Conversational AI or chatbot training data, the API can take a single or multiple utterance input and return a number of paraphrases that match the original intent.

LevelFish combines a number of AI NLG inference engines combined to generate the most relevant alternative phrasing. This is filtered through a set of quality filters to provide a high-quality paraphrase set. Some of these engines are designed to focus on word use, while others are designed to focus on originality of rephrasing.

Get an API token and permission to use an API.

To use any of the LevelFish APIs you will require:

- An account and an API token. Contact sales@levelfish.com if you don't have these.
- An API identifier - this is the name of the paraphrase engine you will use to generate your paraphrases.

Authentication:

Adding your API Token to your request headers will authorise the transaction. Any requests to the LevelFish services using your API Token will be attributed to your contract, allowance or costs. If you feel the API Token may have been compromised, please request a replacement as soon as possible.

To generate paraphrases, we have several alternative methods:

For demonstrations and simple use:

1. Synchronous, Single/Multiple-phrase, simple results

For production implementation:

2. Synchronous, Single/Multiple-phrase, detailed results
3. Asynchronous, Single/Multiple-phrase, detailed results

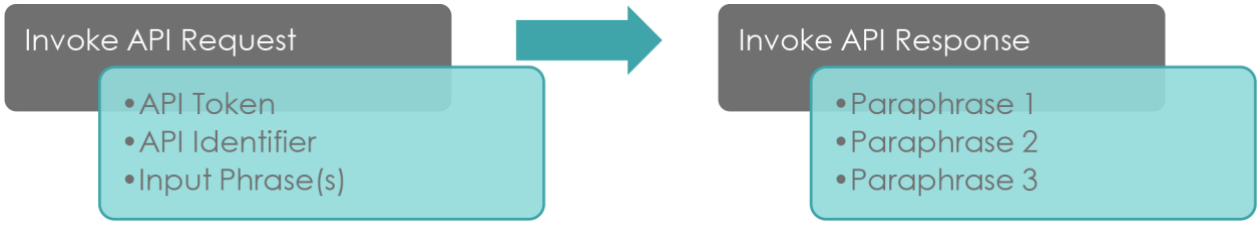
With an additional feedback loop

4. Synchronous, Single/Multiple-phrase, detailed results with feedback.
5. Asynchronous, Single/Multiple-phrase, detailed results with feedback.

Method 1: Synchronous, Single/Multiple-phrase, simple results.

Use: This is ideal for testing or demonstrating paraphrase examples.

This is not recommended for a production implementation as it cannot be used alongside the feedback API. This API call requires the API authentication token.

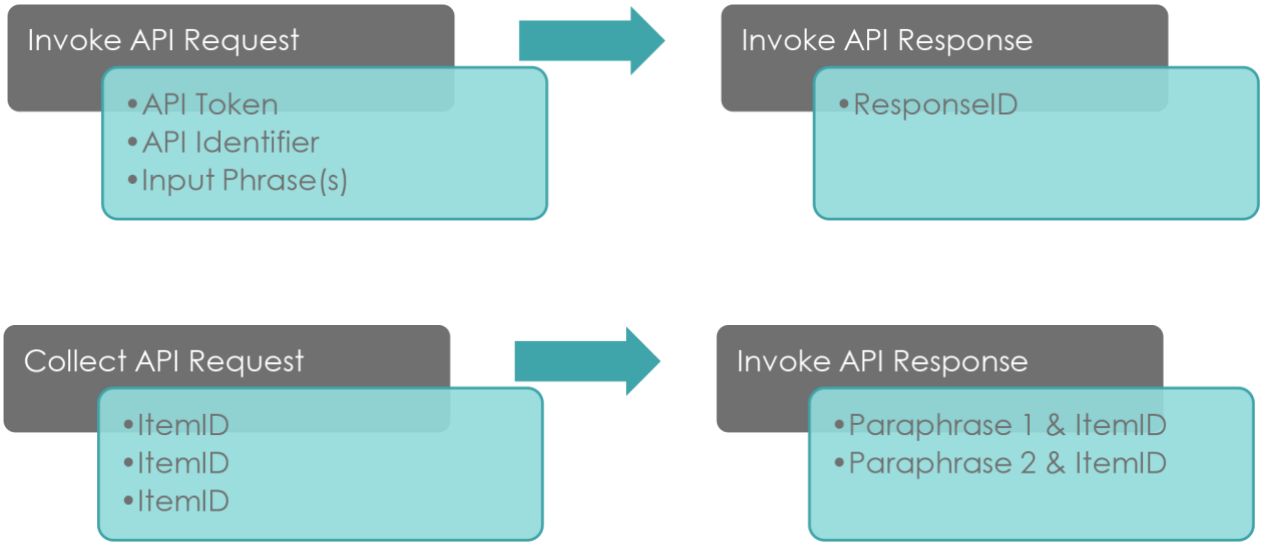


Method 2: Synchronous, Single/Multiple-phrase, detailed results.

This is ideal for a simple implementation and gives information about the source of all the Engines used to generate the paraphrases, an indication of what has been removed by filters and provides the item IDs needed for a feedback API implementation at some later date.

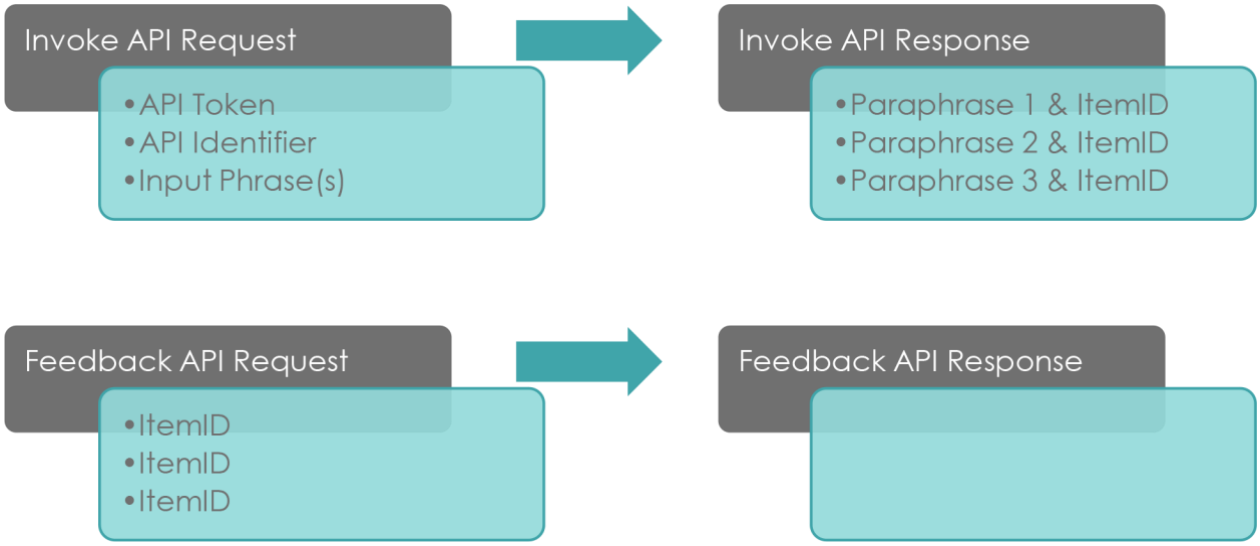
Method 3: Asynchronous, Single/Multiple-phrase, detailed results.

This is ideal for batch processing paraphrase requests. If you wish to send more than 5 phrases at a time it may be better to batch those requests and asynchronously pick up the results.



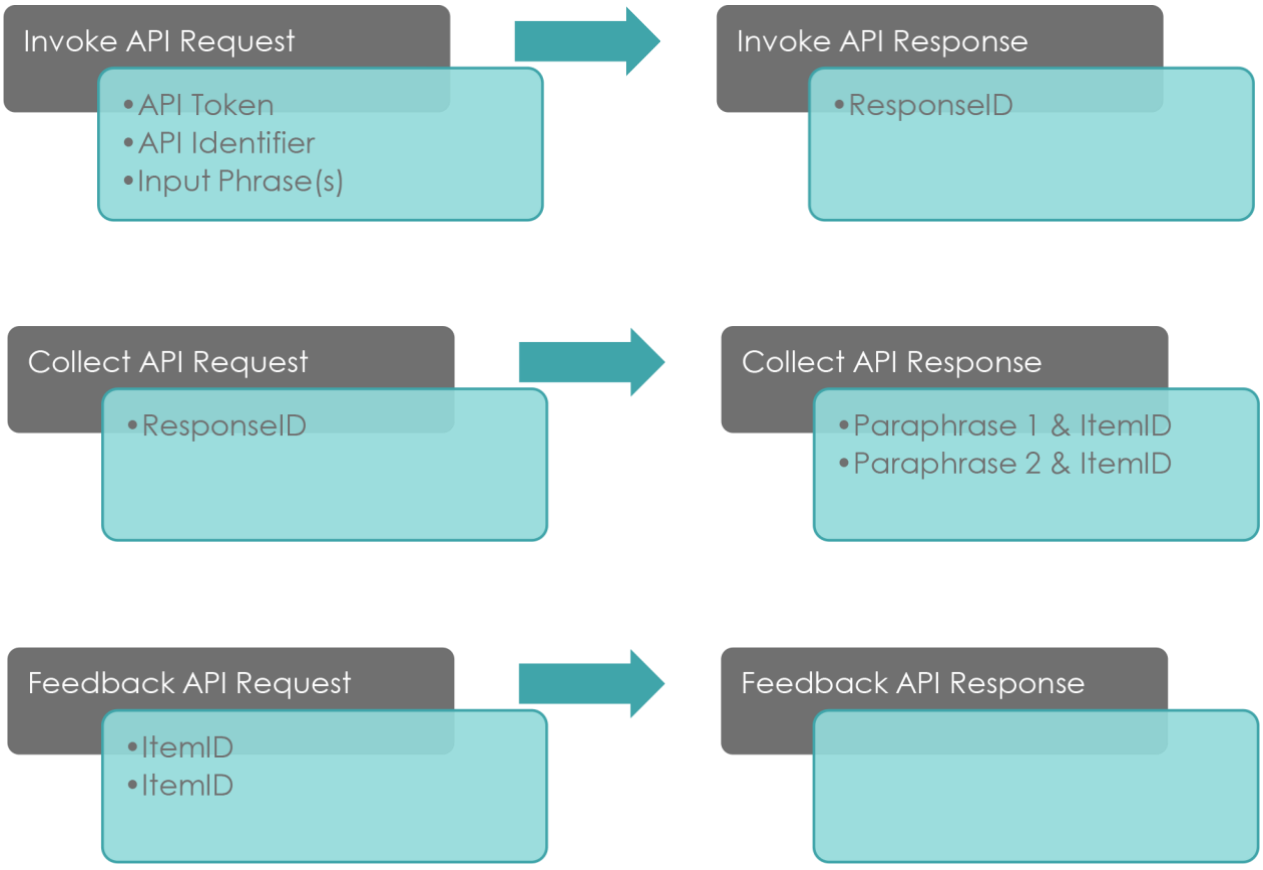
Method 4: Synchronous, Single/Multiple-phrase, detailed results with feedback.

This method is the recommended implementation as it is a synchronous request which makes the reaction real time. In addition, it provides the Item IDs needed to be able to give feedback which paraphrases the user has selected.



Method 5: Asynchronous, Single/Multiple-phrase, detailed results with feedback.

This method can be used for batch processing tasks as above, but with the added feedback call.



API Definition:

The critical APIs used for requesting paraphrases is the INVOCATION API set. This comprises three APIs. Each API accepts a valid JSON document as part of an HTTP POST request.

Authentication/Authorization

Every request to the API must be authenticated and authorized. You will have been supplied with an API token by your support manager. You must provide this as part of every request, by way of an HTTP header. It is important that if you detect your token has been compromised, you should notify us immediately so that we can disable the token and issue you a new one. When calling our API, you should provide the headers listed below.

Header Name	Required	Example Value
Authorization	TRUE	Api <MY-API-TOKEN>
Content-Type	TRUE	application/json
Example HTTP Headers **Note the api token below is an example and cannot be used.*		
"Authorization" : "Api hG8x5Tb8SSMKKUJ0zlvo6myonx2rg1qXD7lhJey4TFdJh57gWD0nt7wpAz5ZpeW7noNBrm"		
"Content-Type" : "application/json"		

Web Service Response

Each request to the API will result in a response with an http status line containing an HTTP status code and message. The result may also contain a body, which will be in the form of application/json data. Each API will return a different JSON body, but the API response code meaning will be the same across all services. The response codes you will most often encounter are listed below.

HTTP Status Code	Http Result	Description
200	Success	The request was received by the server, and processed as expected without error. You should check the body of the response for accompanying JSON data.
400	Bad Request	The request was received by the server and rejected for some reason, most likely the format of the input was invalid or did not validate. You should check the body of the response for accompanying JSON data which will contain a reason for the rejection.
401	Authentication	The server could not authenticate your API token or you attempted to perform an unauthorised action.
404	Not Found	The API or resource you requested was not found.
429	Too Many Requests	The API detected you were calling too often. The response may also contain a **Retry-After** header, which when present, denotes when we will next consider your request

		successfully. Restrictions are set by your account manager. Please speak to support for more information.
500	Server Error	The API encountered an unexpected error. This can happen from time to time and you should cater for this with error handling to retry your request.
503	Bad Gateway	Occasionally our API will automatically scale up or down according to global usage patterns. If you encounter a 503 response code, our gateways have been unable to route your request to a server to handle it. You should back off and retry using an exponential backoff algorithm.

Invoke Web Service

POST /api/v1/invocation/invoke - *makes the paraphrase request and either returns the paraphrase results or an apiResponseIdentifier that is used to collect the results asynchronously.*

Attribute Name	Type	Required	Description
apiIdentifier	String	TRUE	This is the ID of the paraphrase engine to be requested. This should be provided to you by your account manager.
Input	Object	TRUE	The input object contains the data you would like to pass to the API to have processed.
input > inputValues	String Array	TRUE	An array of 1 or more input statement(s) to be passed into the paraphrase engine.
options	Object	FALSE	An object that can be used to customise your request and the format of the response you would like.
options > emailAddresses	String	FALSE	A comma-separated or semi-colon separated list of email addresses. If specified, each address shall receive an email on completion detailing a full breakdown of the results.
options > async	Boolean	FALSE	**default false.** TRUE if you would like to use the collect API to collect the results or receive a result email, FALSE if you would like the webservice to block until the results are ready.
options > simpleResults	Boolean	FALSE	**default false.** TRUE if you would like the format of the response object to be a simple array of input -> output, FALSE if you would like a full complex response model.

Example JSON Input

```
{
  "apiIdentifier": "API-LR123-12312",
  "input": {
    "inputValues": [
      "This is the input statement which I would like to send to the engine"
    ]
  },
```

```
  "options": {
    "emailAddresses": "example1@acme.com,example2@acme.com",
    "async": false,
    "simpleResults": false
  }
}
```

Example Python Script

```
import json
import requests

headers = {"Content-Type" : "application/json",
        "Authorization": "Api YOUR-API-KEY"
}
API_URL = "https://api.levelfish.com/api/v1/invocation/invoke"

def query(payload):
  data = json.dumps(payload)
  print (data)
  response = requests.request("POST", API_URL, headers=headers, data=data)
  return json.loads(response.content.decode("utf-8"))

data = query(
  {
    "apiIdentifier": "API-ACEA5-47232",
    "input": {
      "inputValues" : [
        "Black cars tend to hide dirt better than lighter colors"
      ]
    },
    "options" : {
      "async" : False,
      "simpleResults" : True
    }
  }
)
print (json.dumps(data))
```

Response (Detailed) - *the detailed version of the response contains information on the sources' of the results, and associated metadata.*

Attribute Name	Type	Required	Description
httpResult	String	TRUE	Please refer to the ***Web Service Response*** section.
object	Object	TRUE	The data associated with this response.
object > request	Object	TRUE	The original request object you submitted.
object > status	String	TRUE	• **PENDING** - your request has not finished processing and is not yet ready for collection. • **COMPLETE** - your request has completed and the results are included. • **PARTIAL** - your request was processed, not all operations in the pipeline could be considered in the generation of your results.

			• **ERROR** - a fundamental error was encountered whilst generating your results. You should refer to the messages for more information.
object > apiResponseIdentifier	String	TRUE	Uniquely identifies this request. This should be used if you would like to collect your results at a later time.
object > responseTimeMilliseconds	Integer	TRUE	Time in milliseconds that the API took to complete your response.
object > resultCount	Integer	FALSE	When results are included in the response, the resultCount will denote how many results are contained in the response.
object > results	Object Array	FALSE	If the request is complete or partial, will contain objects with your result data. You will receive a result object per inputValue you supplied in the request.
object > results > inputText	String	TRUE	The original test you supplied in the request.
object > results > itemCount	Integer	TRUE	The number of items in this part of the results.
object > results > items	Object Array	FALSE	An array of result items containing suggestions relating to the inputValue in the request.
object > results > items > itemIdentifier	String	TRUE	A unique identifier used to reference the suggestion. When using the feedback service, this is the identifier you would use to denote your selection of the suggestion.
object > results > items > label	String	TRUE	A label denotes the provenance of each of the items in your results.
object > results > items > value	String	TRUE	This is the value returned by the Level Fish engine.

Example JSON Output


```
{
  "httpResult": "success",
  "object": {
    "request": {
      "apiIdentifier": "API-ACEA5-47232",
      "input": {
        "inputValues": [
          "Hi, I recently bought a device from your company but it is not working as advertised and I would like to get reimbursed!"
        ]
      },
      "options": {
```

```
      "async": false
    }
  },
  "status": "COMPLETE",
  "apiResponseIdentifier": "RSP-HIRBA-WKMRBL3I05",
  "responseTimeMilliseconds": 126,
  "results": [
    {
      "inputText": "Hi, I recently bought a device from your company but it is not working as
advertised and I would like to get reimbursed!",
      "items": [
        {
          "itemIdentifier": "ITM-IBADF-EPCZFBOBVZAFH0O",
          "label": "NG 8-2",
          "value": "I bought a device from your company and it is not working as advertised, so I
would like to get reimbursed."
        }
      ],
      "itemCount": 1
    }
  ],
  "resultCount": 1
 }
}
```


Collect Web Service

POST /api/v1/invocation/collect - *if you have made an asynchronous INVOKE request then using the apiResponseIdentifier this API will return the results.*

Attribute Name	Type	Required	Description
apiResponseIdentifier	String	TRUE	This is the ID that was returned from the invoke call, and can be used at any point after the initial request to recall the request.

Example JSON Input

```
{
  "apiResponseIdentifier": "RSP-HIRBA-AY3F37MOCR",
}
```

Example Python Script

```
import json
import requests

headers = {"Content-Type" : "application/json",
        "Authorization": "Api YOUR-API-KEY"
}
API_URL = "https://api.levelfish.com/api/v1/invocation/collect"

def query(payload):
  data = json.dumps(payload)
  print (data)
  response = requests.request("POST", API_URL, headers=headers, data=data)
  return json.loads(response.content.decode("utf-8"))
```

```
data = query(
  {
    "apiResponseIdentifier": "RSP-HIRBA-AY3F37MOCR",
  }
)
print (json.dumps(data))
```

Response - *please refer to the **invoke** webservice, the response model is the same across the 2 services.*

Feedback Web Service

POST /api/v1/invocation/feedback *- once you have received your results. The Feedback API is used to register which paraphrases your end user has selected for use. The feedback API, while not essential to deploy DOES have a direct bearing on the results for all responses back to your service as this data provides more or less paraphrases from each end point inference engine depending on the frequency of user selection, in other words, as the chatbot users select the most apt responses, they get more of that type and less from another engine which they favour responses less from. This API automatically prioritises the type of data which users select to optimise the optimise the output results.*

Attribute Name	Type	Required	Description
apiResponseIdentifier	String	TRUE	This is the ID that was returned from the invoke call, and can be used at any point after the initial request to feedback usage.
itemIdentifier	String Array	TRUE	An array of Strings containing the item identifier(s) retrieved from the response object of the items your end user selected for use.

Example JSON Input

```
{
  "apiResponseIdentifier": "RSP-HIRBA-AY3F37MOCR",
  "itemIdentifier" : ["ITM-IBADF-OD5ADKWRRD5EERU"]
}
```

Example Python Script

```
import json
import requests

headers = {"Content-Type" : "application/json",
        "Authorization": "Api YOUR-API-KEY"
}
API_URL = "https://api.levelfish.com/api/v1/invocation/feedback"

def query(payload):
  data = json.dumps(payload)
  print (data)
  response = requests.request("POST", API_URL, headers=headers, data=data)
  return json.loads(response.content.decode("utf-8"))

data = query(
  {
```

```
    "apiResponseIdentifier": "RSP-HIRBA-AY3F37MOCR",
    "itemIdentifier" : ["ITM-IBADF-OD5ADKWRRD5EERU"]
  }
)
print (json.dumps(data))
```

Response - *the feedback service merely contains an httpResult denoting the success of the call.*