機微情報の秘匿化APIチュートリアル［プレビュー版］

はじめに

本書は、本サービスが提供する機微情報の秘匿化APIによって、文書内の機微情報の単語を秘匿、復元するチュートリアルです。また、秘匿、復元の過程でシステムに保存されたデータを削除するAPIの使用方法についても説明します。

本書の対象読者は以下を想定しています。

- 本サービスと連携したシステムや製品開発を行う開発者

なお、本書ではPythonのサンプルコードを記載しておりますが、本サービスのAPIはREST形式のため、他の言語からもご利用いただけます。
サポートしているAPIとリクエスト・レスポンスの詳細については、APIリファレンスを参照してください。

機微情報の秘匿化APIとは

機微情報の秘匿化APIは以下の3つのAPIから構成されます。

秘匿API

オリジナル文書を受け取って、自然言語処理モデルによってオリジナル文書内の機微情報の単語を秘匿化します。　機微情報の種類に応じて、 例えば人名であれば「人名_001」、地名であれば「地名_001」のような形式に置き換えて秘匿化を行います。この時の「人名」や「地名」のことを「ラベル」、秘匿化前の単語を「秘匿化前単語」、秘匿化後の単語を「秘匿化後単語」と呼びます。　文書内に同じ種類で異なる秘匿化前単語が存在する場合(佐藤と山田など)、佐藤は「人名_001」、山田は「人名_002」のように、ナンバリングして別の秘匿化後単語に秘匿化します。

> 秘匿化できるカテゴリー一覧は以下となります。
>
> 名字、下の名前、人名、ミドルネーム、職業名、役職、地位社会的身分、国、都道府県州、市区町村、住所、郵便番号、番地、地域、その他地名、日付、生年月日、期間、時間、法人、部署、学校、病院、その他組織、その他施設、年齢、金額、電話番号、マイナンバー、免許証番号、パスポート番号、保険者番号、口座番号、クレジットカード番号、パスワード、ネットワーク情報、その他機微番号、メールアドレス、ID、URL、病名、症状、健診項目、測定値、処置、薬名、人種、宗教、信条、犯罪、国籍、言語、セクシャリティ

モデルによる秘匿結果は、辞書(抽出辞書、除外辞書)を使って修正をすることができます。

また文書を受け取り、文書を文章に分割し、分割された各文章に書かれている主題を抽出します。分割された文章の塊を「セグメント」と呼び、抽出された主題を「トピック」と呼びます。また、セグメントに分けることを「セグメンテーション」、トピックを抽出することを「トピック抽出」、これら一連の処理を「トピック推論」と呼びます。

秘匿時の内容をシステムに保存するために`x-nec-genai-client-id`とconversation_idを指定して実行します。

復元API

秘匿APIで秘匿化時に指定した`x-nec-genai-client-id`とconversation_idを指定して秘匿化された文書を元の状態に戻します。

秘匿履歴削除API

システムに保存された秘匿履歴を削除します。秘匿APIで秘匿化時に指定した`x-nec-genai-client-id`とconversation_idまたは期間を指定して、不要になった履歴を安全に削除できます。

チュートリアルの流れ

本チュートリアルの流れは以下です。

1. 秘匿APIのリクエスト方法および結果の確認
2. 復元APIのリクエスト方法および結果の確認
3. 秘匿履歴削除APIのリクエスト方法および結果の確認
4. （応用）Pythonプログラムから機微情報の秘匿化APIを利用する

秘匿APIのリクエスト方法および結果の確認

以下のcurlコマンドを用いて、APIを呼び出します。

※conversation_idはユーザー間の情報漏洩防止の観点からUUIDのような他のユーザーと重複しないものを推奨しています。

```
curl -X POST "https://ホスト名/genai-api/v1/concealed/hiding" ¥
  -H "Content-Type: application/json" ¥
  -H "x-nec-genai-client-id: sample-client-id" ¥
  -H "Authorization: Bearer APIキー" ¥
  -d '{
    "conversation_id": "e79cfea5-93fa-40ef-9163-6380890ea020",
    "conceal_flag": "true",
    "topic_flag": "true",
    "document": "佐藤花子さんは東京都品川区に住む35歳の会社員です。... (省略) ...",
    "extraction": [
        {"施設": "近所のスーパー"}
    ],
    "exclusion": ["東京都", "品川区"]
}'
```

応答はJson形式で返却されます。

文書が3つのセグメントに分割され、それぞれのセグメントにトピックが付与されていることがわかります。セグメント内のtextに秘匿された文章が格納されています。

```
{
  "status": 200,
  "data": {
    "segments": [
      {
        "segment_no": 1,
        "topics": [
          {
            "topic_name": "基本個人情報",
            "is_personal_information": true,
            "is_corporate_secret": false
          },
          ... (省略) ...
        ],
        "text": "[名字_001][下の名前_001]さんは東京都品川区に住む[年齢_001]の[職業名_001]です。... (省略) ...",
        "risk_words": []
      },
      ... (省略) ...
    ],
    "hidden_words": [
      {
        "[名字_001]": {
          "佐藤": "名字"
        }
      },
      {
        "[下の名前_001]": {
          "花子": "下の名前"
        }
      },
      {
        "[年齢_001]": {
          "35歳": "年齢"
        }
      },
      "... (省略) ..."
    ],
    "labelcount": [
      {
        "名字": 4
      },
      {
        "下の名前": 4
      },
      {
        "人名": 0
      },
      "... (省略) ..."
    ]
  }
}
```

秘匿APIのリクエストヘッダおよびリクエストボディの詳細は以下のとおりです。

リクエストヘッダ

```
Authorization: Bearer [APIキー]
Content-Type: application/json
x-nec-genai-client-id: [クライアントID]
```

リクエストボディ

パラメータ名	型	デフォルト	必須	説明

conversation_id	string	-	○	対話ID (※UUIDのような他のユーザーと重複しないものを推奨)
document	string	-	○	オリジナル文書
conceal_flag	bool	true	-	秘匿要否(true/false)
topic_flag	bool	false	-	トピック推論要否(true/false)
extraction	array	-	-	抽出辞書 モデルで秘匿化できない単語を秘匿化するための辞書
exclusion	array	-	-	除外辞書 秘匿化した単語を元の単語に戻すための辞書

レスポンスボディの各項目の詳細は以下のとおりです。

パラメータ名				型	デフォルト	説明
status				string	-	ステータスコード
data					-	応答メッセージ
	segments			array	-	トピック推論、秘匿化結果 トピック推論しない場合は要素数1の配列で返す
		segment_no		string	-	セグメント毎の連番
		topics		array	-	トピック情報 トピック推論しない場合は空配列
			topic_name	text	-	トピック名
			is_personal_information	boolean	-	個人情報 (true:該当,false:非該当)
			is_corporate_secret	boolean	-	企業秘密 (true:該当,false:非該当)
		text		string	-	秘匿化した結果の文書 秘匿しない場合は原文
		risk_words		array	-	リスクのある単語情報 リスク単語が無ければ空配列
			word	string	-	単語
			label	string	-	ラベル名
			start	integer	-	秘匿化した結果の文書（data->segments->text）における 単語の開始位置
			end	integer	-	秘匿化した結果の文書（data->segments->text）における 単語の終了位置
			score	float	-	リスクスコア
	hidden_words			array	-	秘匿化前後の単語とラベル名
	labelcount			array	-	秘匿化したラベルと件数

復元APIのリクエスト方法および結果の確認

以下のcurlコマンドを用いて、APIを呼び出します。

conversation_idには秘匿時に指定したものと同じ値を指定します。

```
curl -X POST "https://ホスト名/genai-api/v1/concealed/revealing" ¥
  -H "Content-Type: application/json" ¥
  -H "x-nec-genai-client-id: sample-client-id" ¥
  -H "Authorization: Bearer APIキー" ¥
  -d '{
    "conversation_id": "e79cfea5-93fa-40ef-9163-6380890ea020",
    "document": "[名字_001][下の名前_001]さんは東京都品川区に住む[年齢_001]の[職業名_001]です。...（省略）..."
}'
```

応答がJson形式で返却されます。data.answerに復元された文書が格納されています。

```
{
  "status": 200,
  "data": {
    "answer": "佐藤花子さんは東京都品川区に住む35歳の会社員です。...（省略）..."
  }
}
```

復元APIのリクエストヘッダおよびリクエストボディの詳細は以下のとおりです。

リクエストヘッダ

```
Authorization: Bearer [APIキー]
Content-Type: application/json
x-nec-genai-client-id: [クライアントID]
```

リクエストボディ

パラメータ名	型	デフォルト	必須	説明
conversation_id	string	-	○	対話ID (秘匿APIで指定したID)
document	string	-	○	秘匿化文書

秘匿履歴削除APIのリクエスト方法および結果の確認

以下のcurlコマンドを用いて、APIを呼び出します。

```
curl -X DELETE "https://ホスト名/genai-api/v1/concealed/hiding_data" ¥
  -H "Content-Type: application/json" ¥
  -H "x-nec-genai-client-id: sample-client-id" ¥
  -H "Authorization: Bearer APIキー" ¥
  -d '{
    "conversation_id": "e79cfea5-93fa-40ef-9163-6380890ea020"
}'
```

応答がJson形式で返却されます。delete_idsに削除されたレコードのconversation_idが格納されます。

```
{
  "status": 200,
  "data": {
    "deleted_ids": [
      "e79cfea5-93fa-40ef-9163-6380890ea020"
    ]
  }
}
```

以下のように日付を指定して削除することも可能です。

```
curl -X DELETE "https://ホスト名/genai-api/v1/concealed/hiding_data" ¥
     -H "Content-Type: application/json" ¥
     -H "x-nec-genai-client-id: sample-client-id" ¥
     -H "Authorization: Bearer APIキー" ¥
  -d '{
    "start_date": "2025-11-10T00:00:00Z",
    "end_date": "2025-11-17T00:00:00Z"
}'
```

conversation_id指定の時と同様に応答がJson形式で返却されます。delete_idsに削除されたレコードのconversation_idが格納されます。

秘匿履歴削除APIのリクエストヘッダおよびリクエストボディの詳細は以下のとおりです。

リクエストヘッダ

```
Authorization: Bearer [APIキー]
```

```
Content-Type: application/json
x-nec-genai-client-id: [クライアントID]
```

リクエストボディ

パラメータ名	型	デフォルト	必須	説明
conversation_id	string	-	※	対話ID （秘匿APIで指定したID）
start_date	string	-	※	削除期間の開始日（ISO 8601（YYYY-MM-DDTHH:MM:SSZ）） タイムゾーンはUTCとする
end_date	string	-	※	削除期間の終了日（ISO 8601（YYYY-MM-DDTHH:MM:SSZ）） タイムゾーンはUTCとする

※対話IDが指定されている場合は、削除期間(開始日、終了日)が指定されていても無視します。
※対話ID、削除期間の開始日、削除期間の終了日のうち１つは必須となります。

（応用）Pythonプログラムから機微情報の秘匿化APIを利用する

requestsライブラリのインストール

インストールコマンドの例

```
pip install requests
```

Pythonコードを実装する

Pythonコードの例

エディタを開き、下記のコードを記述します。ファイル名を「hiding_maing.py」として保存します。

```
import requests
import json
import uuid
# 共通のヘッダー
headers = {
    "Content-Type": "application/json",
    "x-nec-genai-client-id": "sample-client-id",
    "Authorization": "Bearer APIキー"
}
# UUIDから対話IDを生成
conversation_id = str(uuid.uuid4())
base_url = "https://ホスト名"
print(f"Generated Conversation ID: {conversation_id}¥n")
# 1. 秘匿API呼び出し
print("=== 秘匿API ===")
hiding_data = {
    "conversation_id": conversation_id,
    "document": "佐藤花子さんは東京都品川区に住む35歳の会社員です。...（省略）..."
}
hiding_response = requests.post(
    f"{base_url}/genai-api/v1/concealed/hiding",
    headers=headers,
    json=hiding_data)
print(f"Status Code: {hiding_response.status_code}")
hiding_result = hiding_response.json()
concealed_text = hiding_result['data']['segments'][0]['text']
print(f"秘匿化文書: {concealed_text}")
# 2. 復元API呼び出し
print("¥n=== 復元API ===")
revealing_data = {
    "conversation_id": conversation_id,
    "document": concealed_text
}
revealing_response = requests.post(
    f"{base_url}/genai-api/v1/concealed/revealing",
    headers=headers,
    json=revealing_data)
print(f"Status Code: {revealing_response.status_code}")
revealing_result = revealing_response.json()
print(f"復元文書: {revealing_result['data']['answer']}")
# 3. 秘匿履歴削除API呼び出し
print("¥n=== 秘匿履歴削除API ===")
delete_data = {
    "conversation_id": conversation_id
}
delete_response = requests.delete(
```

```
    f"{base_url}/genai-api/v1/concealed/hiding_data",
    headers=headers,
    json=delete_data)
print(f"Status Code: {delete_response.status_code}")
delete_result = delete_response.json()
print(f"削除された対話ID: {delete_result['data']['deleted_ids']}")
```

Pythonの実行

実行コマンド

```
python hiding_main.py
```

実行結果

```
=== 秘匿API ===
Status Code: 200
秘匿化文書: [名字_001][下の名前_001]さんは[都道府県州_001][市区町村_001]に住む[年齢_001]の[職業名_001]です。...（省略）...
=== 復元API ===
Status Code: 200
復元文書: 佐藤花子さんは東京都品川区に住む35歳の会社員です。...（省略）...
=== 秘匿履歴削除API ===
Status Code: 200
削除された対話ID: ['15e6532a-eb37-46ba-b34a-a024cda2ea46']
```

注意事項

処理時間

秘匿APIは秘匿化処理において自然言語処理モデルを使用しているため、処理の実行に時間がかかる場合があります。

処理時間の目安は以下の通りです：

- 通常時：文書の長さに応じて数秒～数十秒程度
- アクセス集中時：200秒程度かかる場合があります。入力文書の長さに比例してさらに処理時間が延長される可能性があります。

そのため、APIを呼び出す際は適切なタイムアウト設定や時間を空けてのリトライ処理の導入を検討してください。

秘匿化処理

- 秘匿化後の情報も個人情報として適切に扱う必要があります。使用する際には、事前に取得している利用目的の範囲内でお使いください。
- 本機能が秘匿化する対象は個人情報のみです。最高機密事項や極秘事項、お客様情報を秘匿化するものではありません。
- 秘匿化した文字列を生成AIに送信する場合、秘匿化前の文字列は生成AIには伝わらないため、その単語に関する質問や分析を依頼しても回答できません。（例：「阿部さんについて教えて」は「［名字_001］について教えて」として伝わるため）
- 秘匿化の判断は完全ではありません。必ず必要な秘匿化が行われているかを確認してからご利用ください。
- 秘匿化対象は日本語のみとなります。
- 秘匿化対象の個人情報は、文脈や組み合わせによって個人情報になりうるものを幅広く定義しています。そのため過剰な秘匿化を行っているいる印象があるかもしれません。

付録

秘匿APIで秘匿化した文書を生成AIで利用する際のシステムプロンプトの例を記載します。

`入力プロンプトでは、個人識別情報の一部が仮名化タグ（例:［名字_001］,［機微番号_001］）で置き換えられています。仮名化タグがある箇所は、具体的な情報が秘匿された場所であることを認識し、重要な情報である場合は変形や削除をせず、出力に仮名化タグを必ずそのまま反映してください。また、出力に新しい仮名化タグを生成しないようにし、原文に存在するタグのみを使用してください。`