API Backend - Documentation and Examples

The Rest API documented here aims to serve as a backend for different public-facing ontology comparison tools in the Concept Integration Lab.

Technology. The Application Programming Interface (API) has been developed using the FastAPI toolkit. To increase speed, the data and models are held in RAM. Additionally, the ontology text embeddings are indexed using the `nmslib` library, which pre-computes k-nearest neighbors (k-NN) clusters according to angular distance.

Data. The examples below use the 163,596 constitutional segments from the Comparative Constitutions Project, but other corpora can be added later on.

API endpoints

▶ Code

/getEmbedding/

Return embeddings representation of a string.

Parameters:

- q: A string query

Example:

▼ Code

```
        r = requests.get(
            "http://127.0.0.1:8000/getEmbedding/",
            params={"q": "An independent Central Bank"}
        )
        pretty_print(r.text)
```

```
{
  "enc": [
    -0.01824432797729969,
    -0.04618113115429878,
    -0.06630924344062805,
    0.05848051607608795,
    0.06949634850025177,
    0.02356606163084507,
    0.010306106880307198,
    0.04741966351866722,
```

/getDistance/

Compute the angular distance between two strings.

Parameters:

- q1: A string query
- q2: A string query

Example:

▼ Code

```
        r = requests.get(
            "http://127.0.0.1:8000/getDistance/",
            params={"q1": "An independent Central Bank",
                    "q2": "An independent Federal Reserve"}
        )
        pretty_print(r.text)
```

```
{
  "dist": 0.769389842756842
}
```

/getTopMatches/

Return the k closest matches to a reference string in an ontology.

Parameters:

- q: A string query
- k: Number of matches to return
- method: Method to obtain matches. Can be “exact” or “fast” (the latter using the `nmslib` index)
- clusters: Whether to compute and return clusters from the matches
- threshold: (Optional) If returning clusters, what threshold to use

Example with results as-is (fast):

▶ Code

▼ Code

```
        r = requests.get(
            "http://127.0.0.1:8000/getTopMatches/",
            params={"q": "An independent Central Bank",
                    "k": 30,
                    "method": "fast",
                    "clusters": "false"}
        )
        pretty_print(r.text)
```

```
{
  "segment_ids": [
    "Sri_Lanka_2015/3490",
    "Chile_2021/367",
    "Gambia_2018/1864",
    "Papua_New_Guinea_2016/4438",
    "Swaziland_2005/2794",
    "Montenegro_2013/702",
    "New_Zealand_2014/8208",
    "Maldives_2008/401",
```

▶ Code

```
Elapsed time is 0.013942 seconds.
```

Example with results as-is (exact):

▶ Code

▼ Code

```
        r = requests.get(
            "http://127.0.0.1:8000/getTopMatches/",
            params={"q": "An independent Central Bank",
                    "k": 30,
                    "method": "exact",
                    "clusters": "false"}
        )
        pretty_print(r.text)
```

```
{
  "segment_ids": [
    "Sri_Lanka_2015/3490",
    "Chile_2021/367",
    "Gambia_2018/1864",
    "Papua_New_Guinea_2016/4438",
    "Swaziland_2005/2794",
    "Montenegro_2013/702",
    "New_Zealand_2014/8208",
    "Maldives_2008/401",
```

▶ Code

```
Elapsed time is 2.241845 seconds.
```

Example with results as clusters:

▼ Code

```
        r = requests.get(
            "http://127.0.0.1:8000/getTopMatches/",
            params={"q": "An independent Central Bank",
                    "k": 30,
                    "method": "fast",
                    "clusters": "true",
                    "threshold": 0.8}
        )
        pretty_print(r.text)
```

```
{
  "0": [
    "Sri_Lanka_2015/3490",
    "Papua_New_Guinea_2016/4438"
  ],
  "2": [
    "Gambia_2018/1864",
    "Swaziland_2005/2794"
  ],
  "3": [
```

/getClusters/

Compute and return the clusters in a set of strings.

Parameters:

- Request body: A valid JSON file in which values are string queries and keys are labels. One key-value pair should be the threshold, as shown below.

Examples:

▼ Code

```
        json_query = {
            "a": "An independent Central Bank",
            "b" : "An independent Federal Reserve",
            "c" : "An independent Electoral Registry",
            "threshold" : 0.7
        }
        r = requests.post(
            "http://127.0.0.1:8000/getClusters/",
            json=json_query
        )
        print(r.json())
```

```
{'0': ['a', 'b'], 'singletons': ['c']}
```

Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant Number 2315189. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.