This document provides details about Twitter Templates and Mapping Files, explaining how to adjust parameters for generating data that simulates different scenarios and peaks. The included parameters also enable control over the speed of data generation, allowing for the replay of the same dataset at various speeds.

We have used YAML for the twitter templates and triple mapping files. This setting allows to add or modify templates in the future.

1. Template Details

In the context of a conference event, we categorize activities into three stages:

1. Before Conference
2. During Conference
3. After Conference

Each of these stages is defined by two separate YAML files for Twitter templates: sequence.yaml and random.yaml. The templates in sequence.yaml are processed sequentially, while those in random.yaml can be executed in any order multiple times.

The figure below gives an idea of different type of tweets in each of the categories-

Tweet Category	Before Conference	During Conference	After Conference
Announcements △	Main Conference ▲ Call for Papers △ Keynotes, Panelists and Sponsors △	Awards ▲	Next Conference ▲
Reminders ○	Submission ● Registration ○ Student Grants and Volunteer Applications ●	Upcoming Sessions ● Social Gatherings ○	Surveys ●
Notifications □	Accepted Papers ■ Schedule Changes □	Ongoing Sessions ■	-
Insights ✧	Based on Accepted Papers or Emerging Trends ✧	Based on Presentations ✧	Recap and Reflections ✧
Others ◇	Excitement for the Conference ◆ Gratitude for the Grants and Volunteer Opportunities ◆	Job Postings ◇	Acknowledgment and Gratitude ◆

A few examples of natural language based twitter templates are provided in the TwitterExampleTemplates.docx file in the '**Mappings**' folder.

The format of each template is structured as follows:

```yaml
template_name: CallForPapersTemplate
account_type: ConferenceAccount
min_duration: 3
max_duration: 3.5
frequency: 0.5
```

body: | We extend a warm invitation to researchers and scholars to participate in [Conference] by submitting your papers to different tracks. Share your pioneering research and valuable insights with the academic community. Don't miss this opportunity to make a meaningful contribution to the success of the conference! The submission deadline is approaching soon.

Explanation:
•account_type signifies that the tweet is from a conference account (e.g., @iswc_conf) or a person account (e.g.,@gsingh)
•min_duration and max_duration define the time window for the tweet such as between 3 to 3.5 months from a given time and in a particular category.
•frequency controls the tweet generation probability (0.5).
•body contains the actual tweet text with placeholders for dynamic content such as [Conference].

From the above template the below triples will be generated. Each tweet has metadata associated with it with respect to the twitter account and then there will be additional triples depending on the placeholders in the body.

2. Mapping Files

The mapping.yaml file is utilized based on the placeholders present in each tweet template. It consists of triples associated with each placeholder. For example:

```
yaml
- ConferenceAccount:
    - subject: ?postinguser
      predicate: rdf:type
      object: ConferenceAccount
    - subject: ?postinguser
      predicate: posts
      object: ?tweet
    - subject: ?tweet
      predicate: rdf:type
      object: Tweet
    - subject: ?tweet
      predicate: hasInformation
      object: ?information
    - subject: ?tweet
      predicate: hasDateTimestamp
      object: _timestamp
    - subject: ?tweet
      predicate: isAbout
      object: ?conferenceinstance
    - subject: ?tweet
```

```
    predicate: mentions
    object: ?otherusers
  - subject: ?tweet
    predicate: hasHashtag
    object: ?domain
  - subject: ?tweet
    predicate: hasTweetID
    object: ?tweet
  - subject: ?postinguser
    predicate: hasUserName
    object: ?username
  - subject: ?postinguser
    predicate: hasUserID
    object: ?postinguser
- Conference:
  - subject: ?conferenceinstance
    predicate: rdf:type
    object: :Conference
  - subject: ?conferenceinstance
    predicate: hasConferenceName
    object: ?conferenceName
  - subject: ?conferenceinstance
    predicate: hasId
    object: ?conferenceid
  - subject: ?conferenceinstance
    predicate: hasMode
    object: ?conferenceMode
  - subject: ?conferenceinstance
    predicate: hasPaperTrack
    object: ?paperTrack
  - subject: ?conferenceinstance
    predicate: hasWebsiteURL
    object: _url
  - subject: ?conferenceinstance
    predicate: hasLocation
    object: ?location
```

For a specific tweet, all the above defined triples will be generated. Since the triples could be object property assertions, data property assertions or class assertions, we distinguish them with '?', "_', and 'Capital Letter' in the object field respectively. The instances for each entity in the subject or object are either picked from the csv files or generated synthetically.

3. Controlling Speed of Data Generated

Once we have the tweet files, these can be replayed as per user requirements. In order to have a faster execution and avoid overhead in reading timestamps from files, using SPARQL queries, each of the tweet file names have timestamp associated with it. So, the data can be replayed as required by the users. The repository consists of the java file **ObtainTimestamp.java** that can be used to read the time-stamps in order to replay the datasets. Also, having the triples associated with each tweet in separate files, allow for faster execution of sparql queries.