Using YAML for Twitter Templates and Mapping Files

1. Template Details

In the context of a conference event, we categorize activities into three stages: Before Conference, During Conference, and 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 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).

•min_duration and max_duration define the time window for the tweet (3 to 3.5 months from a given time).

•frequency controls the tweet generation probability (0.5).

•body contains the actual tweet text with placeholders for dynamic content.

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 defined triples will be generated.