stanford-corenlp-node

Design notes, v1

Mike Hewett
Shamod Lacoul

October 2012

Goals

- Provide a Node.js library that uses a Stanford CoreNLP server to provide NLP services
- Provide a REST-based web service API that uses the above library to enable access to the Stanford CoreNLP software
- Provide adequate documentation and examples
- Provide automated and manual test cases for the web services and the library

NLP Server lifecycle (client view)

1. Create one or more configurations (*offline*)
2. Start Web Services server (stanford-corenlp-node)
3. Start CoreNLP server (using a configuration)
4. Send text for processing (repeat...)
5. Stop CoreNLP server
6. (optional) Go to Step 3
7. Stop Web Services server

Non-obvious features

- Multiple NLP servers can be started at the same time.
- The same instance of the Node.js server can be used to interact with all of the servers.
- For now, the server will not return 202 statuses. It will wait on long-running NLP processes.
- The NLP results will be returned as-is, enclosed in a JSON envelope.

CoreNLP Configurations

- JSON objects that are used to set parameters when starting the Stanford CoreNLP server
- User can define multiple configurations
- Specify a configuration when starting the CoreNLP server
- Several example configurations are included

Web Services

- POST http://{host}/api/1/server/stanford-corenlp
  - config={path-to-config-file}
  - Returns JSON object, including serverId
  - Can start multiple servers
- POST http://{host}/api/1/nlp/{serverId}
  - text={text...}
  - clientId={user-generated-id}
  - Returns JSON object with results and clientId
- DELETE http://{host}/api/1/server/{serverId}
- GET http://{host}/api/1/status/{serverId}
  - Returns JSON object with the server status