****

**ProntoNLP Functional API**

ProntoNLP is quickly becoming the favorite natural language processing framework for the investment world. Empowering the entire spectrum of financial professionals from portfolio managers, economists, and risk monitors to researchers, analysts, and quants; ProntoNLP delivers cutting-edge tools allowing users to research and develop complete NLP models for any text-based requirements. Efficiently move from theory to model, giving your strategy and clients total confidence. These data-driven strategies and analytics greatly outperform benchmark measures and can be further customized for maximal value.

Our model pipeline is a hybrid system using the best of our linguistic engine coupled with our deep learning sentiment model. This *Deep Linguistics* hybrid system takes full advantage of precise grammatical structures as well as the robust abstractions provided by deep learning models to deliver unparalleled performance. In a single system, users can create and then apply customized NLP models for full information extraction, event classification, sentiment analysis, and data structuring. By using the functional API, users can easily access three main endpoints in a programmatic way, providing for an even smoother integration into existing workflows. These are the main functions: Signal Generation, Extraction Analytics, Process User Documents.

Required Python Libraries:

* Python 3.6+
* gevent (pip install gevent)
* boto3 (pip install boto3)
* websocket-client (pip install websocket-client)

Required: Included ProntoNLP Python files

Best practice recommends first creating a virtual environment and then installing the above packages. Then, in a command terminal, navigate to the local ProntoNLP directory where the above python scripts are located.

**Signal Generation**

The ProntoNLP Signal is a CSV file containing the related event and sentiment counts aggregated at the document level. In that way, users can digest document level metrics derived from the NLP models. Additionally, overall sentiment and theme-related scores are included. The signal file is typically used as a direct input into investment making decisions.

To generate the Signal CSV use the following command line prompt:

python FIEFServerAWS\_DownloadCachedSignal.py [-h] -u USER -p PASSWORD -r RULESET -d DB [-s STARTDATE] [-e ENDDATE] [-g TAGS] outputCSV

* USER: your ProntoNLP username (the same one used to access the web platform)
* PASSWORD: your ProntoNLP password (the same one used to access the web platform)
* RULESET: name of a ruleset that exists in your account (e.g. Alpha, ESG, etc.)
* DB: name of a corpus (e.g. SnP\_ProntoStageDB\_First\_ParseCache.db3)
  + currently only the transcript corpus is relevant to producing signals
* STARTDATE: (optional) limit the signal to start from this date (YYYY-MM-DD)
* ENDDATE: (optional) limit the signal to end at this date (YYYY-MM-DD)
* TAGS: (optional) limit the signal to specific document characteristics (e.g. the Answer section of Earning Calls, companies in a specific sector, only text spoken by the CEO, etc.)
  + For a complete list of applicable tags, select a corpus in the web platform and select filter by tags
* outputCSV: path to store the generated Signal CSV

Example Usage:

python FIEFServerAWS\_DownloadCachedSignal.py -u demo -p demo1 -r Alpha -d SnP\_ProntoStageDB\_First\_ParseCache.db3 -s 2018-01-01 -e 2020-01-01 -g "#DocType\_EarningCall #Sector\_Utilities" mySignal.csv

This will generate the Signal CSV with the following characteristics:

* Values based on extractions from the Alpha model applied to the Transcript Corpus
* For the date range 2018 to 2020
* Limited to Earning Call transcripts and specifically the Utilities sector
* The signal will be saved to the local directory in a file called mySignal.csv

Example Usage:

python FIEFServerAWS\_DownloadCachedSignal.py -u demo -p demo1 -r Alpha -d SnP\_ProntoStageDB\_First\_ParseCache.db3 -g "#DocType\_EarningCall #SpeakerType\_Executives\_CEO #SpeakerType\_Executives\_CFO" mySignal.csv

Similar to above, but no date limitation and aggregating events appearing exclusively in what the CEO or CFO said.

**Extraction Analytics**

The ProntoNLP Extraction Analytics is a CSV file containing the actual extractions on an individual level. Each sentence and related event with all of the various properties produced by the NLP model are returned. In that way, users can easily transform raw unstructured text data into organized events and properties. The Extraction Analytics file is typically used to feed a BI tool or as the basis of downstream analysis.

To generate the Extraction CSV use the following command line prompt:

python FIEFServerAWS\_FindMatches.py [-h] -u USER -p PASSWORD -r RULESET -v EVENTS -d DB [-s STARTDATE] [-e ENDDATE] [-g TAGS] [-m] outputCSV

* USER: your ProntoNLP username (the same one used to access the web platform)
* PASSWORD: your ProntoNLP password (the same one used to access the web platform)
* RULESET: name of a ruleset that exists in your account (e.g. Alpha, ESG, etc.)
* EVENTS: Regex specifying events from Ruleset to extract (e.g. ".\*" or "Dividend|Margin")
* DB: name of a corpus (e.g. SnP\_ProntoStageDB\_First\_ParseCache.db3)
* STARTDATE: (optional) limit the query to start from this date (YYYY-MM-DD)
* ENDDATE: (optional) limit the query to end at this date (YYYY-MM-DD)
* TAGS: (optional) limit the query to specific document characteristics (e.g. the Answer section of Earning Calls, companies in a specific sector, only text spoken by the CEO, etc.)
  + For a complete list of applicable tags, select a corpus in the web platform and select filter by tags
* M: Flag to download extra metadata
* outputCSV: path to store the generated Extraction CSV

Example Usage:

python FIEFServerAWS\_FindMatches.py -u demo -p demo1 -r Alpha -v "Margin|Dividend" -d SnP\_ProntoStageDB\_First\_ParseCache.db3 -s 2018-01-01 -e 2020-01-01 -g "#DocType\_EarningCall #Sector\_Utilities" myExtractions.csv

This will generate the Extraction CSV with the following characteristics:

* Extractions and their properties which matched on either a Margin or Dividend event from the Alpha model in the Transcript Corpus
* For the date range 2018 to 2020
* Limited to Earning Call transcripts and specifically the Utilities sector

Example Usage:

python FIEFServerAWS\_FindMatches.py -u demo -p demo1 -r Alpha -v "Excited|Disappointed" -d SnP\_ProntoStageDB\_First\_ParseCache.db3 -g "#SpeakerType\_Executives\_CEO #SpeakerType\_Executives\_CFO" myExtractions.csv

Similar to above, but no date limitation and showing extractions for Excited and Disappointed events exclusively in what the CEO or CFO said.

**Process User Documents**

This function allows users to upload documents to be processed by a ProntoNLP model of choice. The outputted format is chosen by the user and can be similar to either option above (i.e. actual extractions or aggregated event counts). The input documents must be a CSV file where each row contains a document. The system will run the selected model on the documents and add the result as an additional column to the input. This allows users to run ProntoNLP models on private documents to create a trading signal or generate insights.

To Process User Docs use the following command line prompt:

python FIEFServerAWS\_ProcessCorpus.py [-h] -u USER -p PASSWORD -r RULESET -o OUTPUTTYPE -n NUMTHREADS inputCSV outputCSV

* USER: your ProntoNLP username (the same one used to access the web platform)
* PASSWORD: your ProntoNLP password (the same one used to access the web platform)
* RULESET: name of a ruleset that exists in your account (e.g. Alpha, ESG, etc.)
* OUTPUTTYPE: type of the processing output (‘XML’, ‘JSON’, ‘events’)
* NUMTHREADS: number of concurrent processing threads
* inputCSV: path to CSV containing the user documents
* outputCSV: path to store the generated CSV

Example Usage:

python FIEFServerAWS\_ProcessCorpus.py -u demo -p demo1 -r Alpha -o events customDocs.csv customDocs\_analyzed.csv