

# BDL Code Documentation

## Contents

<b>1_data_processing</b>	<b>2</b>
1_pull_documents . . . . .	2
2_doc_processing . . . . .	2
3_get_measures . . . . .	3
4_sample_selection . . . . .	3
5_control_variables . . . . .	4
6_LDA_topic_modeling . . . . .	4
7_merge . . . . .	4
<b>2_validation_test</b>	<b>4</b>
1_sample_selection_and_preparation . . . . .	4
2_run_through_GPTZero . . . . .	5
3_analysis . . . . .	5
<b>3_regression</b>	<b>5</b>



# 1\_data\_processing

## 1\_pull\_documents

- **pull\_sec\_filings.R**: pull SEC filing index data (10-K, 10-Q, 8-K, S-1) from WRDS SEC Analytics Suite and generate file lists for later downloading from WRDS server (see downloading instructions at the bottom of the file).
- **pull\_conference\_calls.py**: download earnings call transcripts from WRDS Capital IQ, keep the latest transcript version per event, and aggregate presenter speech into a single text file per call.

## 2\_doc\_processing

### 10k

- **doc\_processing\_10k.py**: clean and preprocess raw 10-K filings.
- **split\_into\_sections\_10k.py**: split cleaned 10-K filings into structured sections (e.g., Risk Factors and MD&A) based on filing headers.
- **remove\_sticky\_sentences\_10k.py**: identify and remove sticky sentence that contains a six-word phrase repeated from the prior year, isolating newly written disclosure content.

### 10q

- **doc\_processing\_10q.py**: clean and preprocess raw 10-Q filings.
- **split\_into\_sections\_10q.py**: split cleaned 10-Q filings into structured sections (e.g., Risk Factors and MD&A) based on filing headers.
- **remove\_sticky\_sentences\_10q.py**: identify and remove sticky sentence that contains a six-word phrase repeated from the prior year, isolating newly written disclosure content.

### 8k

- **doc\_processing\_8k.py**: clean and preprocess raw 8-K filings and identify Exhibit 99 attachments.
- **check\_exhibit\_99.py**: flag whether an Exhibit 99 is an earnings press release using regex-based keyword matching on exhibit text and metadata (filename, title, and description).
- **remove\_sticky\_sentences\_8k.py**: identify and remove sticky sentence that contains a six-word phrase repeated from the prior year, isolating newly written disclosure content.

### s1

- **doc\_processing\_s1.py**: clean and preprocess raw S-1 filings.
- **split\_into\_sections\_s1.py**: split cleaned S-1 filings into structured sections using table-of-contents (TOC) anchors.
- **combine\_sections\_s1.py**: map TOC headings to standardized section categories (e.g., Business Description, MD&A, Risk Factors, Prospectus Summary) and combine text within each category.

### conference\_calls

- **remove\_sticky\_sentences\_conference\_calls.py**: identify and remove sticky sentence that contains a six-word phrase repeated from the prior year, isolating newly written disclosure content.



### 3\_get\_measures

#### 10k

- **length\_fog\_by\_section\_10k.py**: calculate section-level (e.g., Risk Factors and MD&A) length (number of words) and readability (Fog index) for 10-K filings.
- **sentiment\_finbert\_by\_section\_10k.py**: calculate section-level tone measure for 10-K filings by labeling each sentence as positive, negative, or neutral using FinBERT.
- **specificity\_by\_section\_10k.py**: calculate section-level specificity measure for 10-K filings, which is the number of entities (locations, people, organizations, dollar amounts, percentages, dates, or times) identified by the Stanford Named Entity Recognizer (NER), scaled by total words. Ratio is multiplied by 1,000.

#### 10q

- **length\_fog\_by\_section\_10q.py**: calculate section-level (e.g., Risk Factors and MD&A) length (number of words) and readability (Fog index) for 10-Q filings.
- **sentiment\_finbert\_by\_section\_10q.py**: calculate section-level tone measure for 10-Q filings by labeling each sentence as positive, negative, or neutral using FinBERT.
- **specificity\_by\_section\_10q.py**: calculate section-level specificity measure for 10-Q filings, which is the number of entities (locations, people, organizations, dollar amounts, percentages, dates, or times) identified by the Stanford Named Entity Recognizer (NER), scaled by total words. Ratio is multiplied by 1,000.

#### 8k

- **length\_fog\_8k.py**: calculate length (number of words) and readability (Fog index) for press releases in 8-K filings.
- **sentiment\_finbert\_8k.py**: calculate tone measure for press releases in 8-K by labeling each sentence as positive, negative, or neutral using FinBERT.
- **specificity\_8k.py**: calculate specificity measure for press releases in 8-K, which is the number of entities (locations, people, organizations, dollar amounts, percentages, dates, or times) identified by the Stanford Named Entity Recognizer (NER), scaled by total words. Ratio is multiplied by 1,000.

#### conference calls

- **length\_fog\_conference\_calls.py**: calculate length (number of words) and readability (Fog index) for conference calls.
- **sentiment\_finbert\_conference\_calls.py**: calculate tone measure for conference calls by labeling each sentence as positive, negative, or neutral using FinBERT.
- **specificity\_conference\_calls.py**: calculate specificity measure for conference calls, which is the number of entities (locations, people, organizations, dollar amounts, percentages, dates, or times) identified by the Stanford Named Entity Recognizer (NER), scaled by total words. Ratio is multiplied by 1,000.

### 4\_sample\_selection

- **sample\_selection.Rmd**: construct the main analysis sample for Item 1A (Risk Factors), Item 7 (MD&A), conference calls, and press releases following the sample selection procedures in Table 1A.
- **edgar\_scraper\_file.ipynb**: scrape IPO filing data from the EDGAR website using SEC file numbers from LSEG's New Issue database (formerly SDC Platinum) for subsequent S-1 sample selection.
- **sample\_selection\_S1.Rmd**: construct the IPO S-1 filings sample.



## 5\_control\_variables

- **pull\_vwret\_by\_decile.ipynb**: calculate size-adjusted return, where size adjustment is based on decile of market value for all of CRSP.
- **pull\_control\_variables.Rmd**: pull and construct control variables from CRSP, Compustat, IBES and Audit Analytics, including firm size, net income, book-to-market, leverage, institutional ownership, analyst following, indicator for M&A, number of segments, abnormal returns and audit fees, and an indicator for an internal IR officer.
- **pull\_gics.Rmd**: pull and construct firm-level two-digit GICS industry classifications from Compustat for use in industry-level analyses.

## 6\_LDA\_topic\_modeling

- **LDA\_topic\_modeling\_MDA.py**: apply Latent Dirichlet Allocation (LDA) to MD&A text to identify disclosure topics at the sentence level, which are then used to examine variation in GAI usage across topics.
- **LDA\_topic\_modeling\_MDA\_industry\_45.py**: apply LDA topic modeling to MD&A text for firms in the Information Technology industry (two-digit GICS 45) to examine topic-level GAI usage within the industry with the highest adoption.
- **calculate\_change\_genscore.R**: calculate average post-period changes in GenScore by topic, using the outputs from the above two scripts.

## 7\_merge

- **clean\_gptzero\_results.R**: clean and restructure GPTZero output, parse disclosure identifiers from filenames, construct document-level GenScore (AI + mixed probabilities), and produce analysis-ready GenScore datasets for the four main disclosures and S-1 filings.
- **clean\_linguistic\_properties.R**: clean linguistic property measures (length, readability, tone, and specificity) across disclosures and output standardized datasets for analysis.
- **create\_final\_datasets.Rmd**: merge GenScore measures, linguistic properties, and control variables to construct the final analysis datasets used in the main regressions and tables.

## 2\_validation\_test

### 1\_sample\_selection\_and\_preparation

- **1\_validation\_sample\_selection.R**: construct the validation samples for Item 1A, Item 7, conference calls, and press releases by restricting to years 2018-2020, stratifying firms into deciles of 2019 market value, and randomly selecting 50 firms per decile with balanced three-year panels, for a total of 500 firms and 1,500 reports per disclosure type.
- **2\_generate\_samples.py**: generate validation samples using the first 5,000 words from each report, rounding up to finish the last sentence.
- **3\_ChatGPT\_API\_batch\_sentence\_level.py**: use the ChatGPT API to generate sentence-level GAI-modified text for validation and power tests.
- **4\_split\_into\_chunks\_and\_ChatGPT\_API\_batch.py**: split disclosures into manageable text chunks (i.e., 500-word chunks) and submit them to the ChatGPT API in batches to generate GAI-modified content for validation analyses.
- **5\_check\_for\_stickiness.py**: identify sticky sentence that contains a six-word phrase repeated from the prior year, isolating newly written disclosure content.
- **6\_combine\_gptzero\_results.py**: combine and restructure GPTZero output for validation samples.



- **7\_overlap\_between\_original\_and\_modified\_reports.py**: calculate overlaps between original and GAI-modified reports.
- **8\_assemble\_validation\_dataset.py**: clean and prepare the final analysis datasets for validation samples.

## 2\_run\_through\_GPTZero

- **item\_1A\_with\_sticky.py**: generate synthetic Item 1A disclosures by randomly replacing a fixed share of sentences (0%, 1%, 2.5%, 5%, 10%, 25%, 50%, and 100%) with GAI-modified sentences, and the modified disclosures are evaluated using GPTZero.
- **item\_1A.py**: generate synthetic Item 1A disclosures by randomly replacing a fixed share of sentences (0%, 1%, 2.5%, 5%, 10%, 25%, 50%, and 100%) with GAI-modified sentences and then remove sticky sentences, and the modified disclosures are evaluated using GPTZero.
- **item\_7\_with\_sticky.py**: generate synthetic Item 7 disclosures by randomly replacing a fixed share of sentences (0%, 1%, 2.5%, 5%, 10%, 25%, 50%, and 100%) with GAI-modified sentences, and the modified disclosures are evaluated using GPTZero.
- **item\_7.py**: generate synthetic Item 7 disclosures by randomly replacing a fixed share of sentences (0%, 1%, 2.5%, 5%, 10%, 25%, 50%, and 100%) with GAI-modified sentences and then remove sticky sentences, and the modified disclosures are evaluated using GPTZero.
- **conference\_calls\_with\_sticky.py**: generate synthetic conference calls disclosures by randomly replacing a fixed share of sentences (0%, 1%, 2.5%, 5%, 10%, 25%, 50%, and 100%) with GAI-modified sentences, and the modified disclosures are evaluated using GPTZero.
- **conference\_calls.py**: generate synthetic conference calls disclosures by randomly replacing a fixed share of sentences (0%, 1%, 2.5%, 5%, 10%, 25%, 50%, and 100%) with GAI-modified sentences and then remove sticky sentences, and the modified disclosures are evaluated using GPTZero.
- **press\_release\_with\_sticky.py**: generate synthetic press release disclosures by randomly replacing a fixed share of sentences (0%, 1%, 2.5%, 5%, 10%, 25%, 50%, and 100%) with GAI-modified sentences, and the modified disclosures are evaluated using GPTZero.
- **press\_release.py**: generate synthetic press release disclosures by randomly replacing a fixed share of sentences (0%, 1%, 2.5%, 5%, 10%, 25%, 50%, and 100%) with GAI-modified sentences and then remove sticky sentences, and the modified disclosures are evaluated using GPTZero.
- **ai.py**: evaluate document-level replacement disclosures for all four disclosure types using GPTZero.

## 3\_analysis

- **GPTZero\_Analysis\_Part\_I.do**: produce summary statistics of GenScore for human-written reports and GAI-modified reports.
- **GPTZero\_Analysis\_Part\_II.do**: conduct power analysis for document-level and sentence-level replacement disclosures.
- **regressions\_lingOnValid.do**: summarize linguistic properties of human-written and GAI-modified reports.
- **confusion\_matrix.do**: construct confusion matrices to evaluate binary GenScore classification performance.

## 3\_regression

- **data\_prep.do**: clean and prepare the final analysis datasets and produce the inputs required for downstream regression and figure-generation scripts.
- **figures\_and\_regressions.do**: execute the main regression specifications and produce the figures and tables used in the primary results section.