**SUMMARY**

Wil Gonzales

**APPROACH:** Table extraction, then classification based on context (and features)

**STEPS**

PRELIMINARIES

1. Import packages (some examples)

* SEC edgar downloader
* BeautifulSoup
* TextWrap
* Html2text
* Pandas
* Tqdm
* NLTK
* Genism
* SpaCy
* pyLDAvis
* html

1. Download the 10-Q files using the list of tickers via SEC Downloader

EXPLORATORY ANALYSIS

1. Created dataframe that has company, year, report number, and report text as columns
2. Pre-processing of report text (e.g., removing html elements),
3. Created wordcloud from pre-processed report text
4. Attempted to do topic modelling/LDA; got coherence of 0.44

ANALYSIS PROPER

Data preparation

1. Extracted the html structure, html text, year information, company name of each file
2. Made list of cleaned tables (and table html text for searching and matching later) per document while filtering out tables that do not have country names in them (simplification)
3. Created a long format dataframe with tables as units instead of document (iterating through the list earlier and original dataframe)
4. Created variables that see whether table html text can be matched to the overall html text using lambda expression; found that there were a lot of errors
5. Correction of html text because not standardized (i.e., .replace({‘<br/>’: ‘<br>’, ‘\xa0’:’&#160;’, “\’Arial\’”:”\\\’Arial\\\’”})

Context extraction

1. Define function that extracts strings from left and right of table html text
2. Create string variable ‘context’ (left + right strings)

Topic modeling

1. Pre-processing for LDA: used htmllaundry package’s strip\_markup, sanitize, and preprocess\_text functions along with lambda functions to preprocess html text to get meaningful text (e.g., lemmatization, stemming, removing numbers, contraction expansion, stop words removal, etc.)
2. Create vocabulary of all words in data, create document term matrix
3. Use LDA to create topics along with the probability distribution for each word in the vocabulary
4. Identified two clusters (used two components) , the first of which is relevant (i.e., ['total', 'location', 'country', 'region', 'month', 'ended', 'geographic', 'customer', 'net', 'revenue'])

Feature engineering/extraction

1. Coded the clustering information as a binary variable [MAY BE USED AS OUTPUT]
2. Used sklearn’s CountVectorizer and nltk’s stemmer on the cleaned text columns to get a reduced and more concise document term frequency matrix (630 variables)
3. Used sklearn’s TfidfVectorizer to get relative frequency matrix (630 more variables)
4. Added these count frequencies and tfidf results as factors; made columns unique by creating function

Annotation

1. Manually annotated each table in the dataframe with a 1 (geographic segmentation) or 0 (non-geoseg). Ended up annotating 30 tables out of 250

Feature selection (to somehow address overfitting)

1. Used Boruta algorithm/random forests to narrow down the 1261 variables to 176 useful variables (parameters: 100 iterations, balanced class weight)

Training/ Supervised machine learning

1. Split the 30 annotated files into training and test sets
2. Fitted logistic regression model for classification (Accuracy = 1, f-1 = 1 due to small sample)
3. Prediction: used model to predict the rest of the unseen and unannotated table rows

Lookup table/JSON/CSV preparation

1. Subset dataframe: only get following variables: company, year, reportnumber, tableid, table, predicted geo segmentation, topic modelling cluster binary
2. Converted the subset dataframe to another long format dataframe that has figures as the row unit, with quarter information and figure number coded as two variables
3. Converted the dataframe to JSON format (by rows), to CSV

A picture containing text, black, scoreboard

Description automatically generatedExample:

{'Company': 'INFN',

'Year': 2020,

'ReportNumber': 65,

'tableid': 1,

'table': [['Three Months Ended'],

['March\xa028, 2020', 'March\xa030, 2019'],

Graphical user interface, application

Description automatically generated ['United States', '170,526', '132,522'],

['Other Americas', '19,688', '15,132'],

['Europe, Middle East and Africa', '88,578', '98,992'],

['Asia Pacific', '51,481', '46,061'],

['Total revenue', '330,273', '292,707']],

'predictedgeoseg': 0,

'topicfrommodelreal': 1,

'totalrevenues': ['Total revenue', '330,273', '292,707'],

'revenueheader': ['Three Months Ended'],

'figureid': 2,

'figure': '330273',….

LIMITATIONS

* Overfitting
* simplifying assumptions
* no accurate table parser
* small and unrandomized annotation set (and small training set)
* no cross-validation

ASSIGNED MAR 25, 2021, COMPLETED MAR 26, 2021