Machine Learning Engineer Nanodegree

Capstone Proposal

Andrei Yarmak July 16th, 2019

Domain Background

When it comes to financial performance of a large company (especially the publicly traded ones), a lot of information is usually readily available for shareholders and general public. One can analyze annual and quarterly reports, press releases, analyst call transcripts, etc. However, if you want to aggregate any metrics across multiple companies (e.g., to look at marketing spend trends in a given industry over time), data gathering usually requires a lot of tedious manual work. Financial reports are usually posted in multiple different formats, include different sets of metrics, the same metrics can be called different names, a lot of the data is unstructured and buried within pages of text, etc. Even though multiple companies provide some aggregated data as a service, this service usually comes at a very steep subscription fee, which only large businesses can afford.

As an alternative, it might be possible to build a suite of machine learning models that would be able to automatically analyze annual reports of public companies, parse key performance metrics, and collect all necessary context (company name, industry, currency used, etc.) to enable the ease of aggregating these data points and making industry-wide comparisons.

Problem Statement

As a first step on a route to fully automated parsing of financial data from publicly available sources, in this capstone project, I want to build a model that locates a statement of operations within an annual financial report.

This problem is related to structuring unstructured or loosely structured data. It can be framed in several different ways: as a classification problem (i.e., if we split the text in words and then classify each word as either related or unrelated to statement of operations) or as a regression problem (i.e., using all contents of the text at the same time (including numerical positions of every word) trying to estimate the positions of upper and lower boundaries for the statement of operations within it.

In this sense, this problem is clearly:

1) Quantifiable: it can be framed with numerical inputs (word positions, word counts, share of digits in a text, etc.) and outputs (which in classification case would be the probability of the word being a part of statement of operations and in regression case - numerical positions of upper and lower boundaries of the statement of operation within the text)

2) Measurable: the problem can be measured by some metric (e.g., classification or regression accuracy)

3) Replicable: it can be reproduced in pretty much any annual report across most companies and industries

Datasets and Inputs

In this project, I will be using publicly available data from EDGAR (Electronic Data Gathering, Analysis, and Retrieval system)[1], a database of all public filings that companies make with US Securities and Exchange Commission (SEC). More specifically, EDGAR provides access to forms 10-K, which include most important company financial and operating results of the company for a year (annual report), as well a lot of management's commentary on company's performance.

As a preparatory work for this project, I collected and labeled all necessary data from the site. All relevant files that I created as part of that work are provided as part of this project proposal:

[1] https://www.sec.gov/edgar.shtml

1) Data collection phase:
    a. data_collection.ipynb includes a web crawler that I used to navigate EDGAR website and download all relevant files
    b. files.zip includes all the files that I created in the process, with one folder dedicated to each individual company
2) Label tagging phase:
    a. label_tagging.ipynb includes the code that I used to find and tag statements of operations in the text (via a combination of regex searches and manual checks / overrides)
3) Final dataset:
    a. labeled_reports.csv includes a table of 596 annual reports and statements of operations. All reports come from the filings made by different companies in the second quarter of 2019. The table includes 3 fields:
        i. dir - the company name
        ii. report - the whole 10-k annual report downloaded from EDGAR
        iii. pnl - statement of operations (also called profit and loss statement) parsed from the report

Solution Statement

While there are multiple ways to go about framing and solving this problem, I would primarily try to frame it as a classification one, and then use appropriate machine learning techniques to arrive to an answer.

More specifically, I will split each report in words, and then describe each word through characteristics (e.g., word counts, % of digits in the text, % of whitespace, specific word occurrences / counts, etc.) of:

- The text above it
- The text below it
- The text in its immediate proximity (e.g. within 100 words to each side)

This would enable me to then use common classification algorithms (Naive Bayes, SVMs, MLP, etc.) to estimate the probability that a point is above, below, or within statement of operations.

Benchmark Model

As a benchmark, I can potentially use a simple model that would just use the average position of the statement of operation (i.e., X% of total words from the top of the report and Y% of words from the bottom) across all 596 reports. This model is easily measurable through basic evaluation metrics (accuracy, etc.).

Evaluation Metrics

Average accuracy of classifying a data point (i.e. a word within a report) on the testing dataset might be a good candidate to be used for evaluation of both benchmark and solution models. More specifically, the accuracy will be defined in this case as a number of times a specific word was correctly classified as either being above, below, or within the statement of operations.

Project Design

Based on the description of the project I outlined above, its theoretical workflow should roughly follow the steps below:

1) **Preprocess the data**. This would include two main steps:
    a. *Preprocess model inputs*:
        i. Split each report into individual words

ii. Encode the required features that would describe each word. These features would describe text above, below, and in the near vicinity (e.g., the percent of numeric items as a share of total, number of occurrences of specific words, etc.) of the word into question, as well as the word itself (e.g., is it numerical, is it one of the encoded keywords, etc.)

b. *Preprocess the labels*. This would entail using the pre-parsed statements of operations (provided as a part of the input dataset) to automatically label each word as being above, below, or within the statement of operations

c. *Append all preprocessed reports and labels*. After this step, I will have the final dataset that can be used for training

2) **Split the data into training, validation, and testing sets**

3) **Build a classifier model.** The model should be trained to estimate the probability of a given word being above, below, or within a statement of operations (according the the labels coming out of step 1b above).

The two main candidate algorithms I would like to start with for this step are Naive Bayes and SVM models:

- Naive Bayes might potentially be a good fit due to its ease of implementation, scaling well with the data, being able to perform both binary and multi-class classification, working with both discrete and continuous data
- Support Vector Machines (SVM) might potentially be a good fit as they usually have very high accuracy, are very good at modeling non-linear decision boundaries, provide a wide range of kernels available to choose from, have robust theoretical protection against overfitting

If none of the above are able to produce satisfactory outcome, I might consider deep learning (e.g., an MLP) as another option to look at.