# Legal Verbiage and Topic Modeling

Anthony Chi & Chris Han

**May 21st, 2018**

## I. Definition

### Project Overview

Docket texts are up-kept court public records that detail historical events that happened on legal cases. These new texts/records are available when there’s an update/progress on the case. They are generated by clerks, and sent to the responsible attorneys, often via email. One of law practices’ day-to-day is to respond and react to these records. To help attorneys efficiently handle these records, we wanted to try to leverage technology and hopefully automate the docket text handling processes for law practices.

As Is process for many law practices, especially smaller practices: Lawyers spend time scanning through docket text emails, and determine whether or not to respond and react to the docket texts, and how to respond and react.

To be process: some logic scan the newly received docket texts and determine whether or not to respond and react. If needed, take the appropriate action such as draft up and email, schedule an appointment with the client, etc.

### Problem Statement

There are many difficulties with this exercise:

1. Difficult to acquire a large database: even though these texts are publicly available, they do come with a price. The process of acquiring data meant someone with access to the government court website would have to log in, search for cases that may or may not be relevant, and download the latest docket texts.
2. This is an unsupervised exercise: there is no target for this dataset, meaning the dataset does not come with answers on whether a lawyer should respond and react, or how to respond or react. Therefore, unless someone is willing to go through each individual texts to determine the targets for the dataset, this is an unsupervised classification exercise.
3. Lots of SME is required: to perform NLP or topic modeling for this exercise, we need to avoid GIGO (garbage in, garbage out). This meant that in each step, we needed experts to interpret our input and output results:
   1. During NLP, we need to clean enough to remove all the unwanted texts, but not too much so the origin meanings or relevant keywords were lost.
   2. After NLP, is there an easy way to filter the features to provide the correct targets?
   3. During Topic Modeling, we need to determine the best suited number of topics, as well as interpret the topics correctly.
   4. Testing: we also require experts to tell us whether our model is working for newly introduced datasets.
   5. After Topic Modeling, we also need experts to tell us if actions need to be taken, and what type of actions are relevant.
4. If topic modeling does not work, how should we proceed?
   1. Collect more data?
   2. Make this exercise a supervised classification exercise, and ask the expert to provide targets for each feature?

### Metrics

Since this is an unsupervised exercise, we rely on our expert to determine the accuracy on whether the the model able to classify topics correctly.

## II. Methodology

### Data Exploration

The goal here was to examine the available data, and achieve the following objectives:

Compare Data Sources: many data sources are becoming unavailable, turned into paid services, or only available on proprietary platforms. Many APIs were also inactivated, and webpages were inactivated or unable to be scrapped. I was only able to access three data sources: Quandl API, NASDAQ, and finance.yahoo.com.

After I compared the three data sources, there were apparent differences in Adjusted Closing Price and Adjusted Volume. There were also occasional missing data from Quandl API; the NASDAQ maximum historical data only spanned 10 years. With these difficulties, I relied soley on finance.yahoo.com’s data.

<insert the difference in data chart>

Using the available features, including open, close, high, low, volume and adj. close, I generated additional features:

* High-Low Range, as a percentage of previous day’s close: this described how volatile the day’s movements were
* Open-Close Range, as percentage of same day’s open: this described the general movement of the day’s activities
* Moving average (adjusted close & volume) for a specific time period.

The new and old features needed to answer the following questions or explore ideas:

* What am I trying to predict: specific price, change in price, simple direction, or general trend/direction? This will determine if this is a classification (bi or multi) or a regression exercise.
* Were there any relationship between the independent and dependent features?
* What did the feature distributions look like? Did I need to make any transformation?
* Were there any missing or outliers that I needed to look into?
* Did I needed to use scalers, or other feature engineering techniques such as PCA?
* Was there a model that can be applied to all equities? Or did I need to train for each equity?

### Exploratory Visualization

**Outliers**

<insert distribution plots for features>

**Missing Data**

<insert chart for missing feature>

**New Features created**

**Target**

<insert target distribution and transformation chart>

**Features Transformed**

**PCA**

### Benchmark

1. I used two benchmark models, namely XGBoost Classifier and Lasso Regression. The ultimate goal was to use LSTM to beat these benchmark models. Supposedly, LSTM should be able to incorporate long and short term memories to generate additional insights, and further improve the predictive power of resulting models.
2. During the LSTM optimization exercise, I attempted to explore how the hyperparameters such as window, epoch, batch size, and LSTM constructs affected the model performances. This surely helped me tune the model for better performances.
3. Also note this was a regression, as well as, a classification exercise.
   * 1. As the exercise became computationally expensive, I moved computational and storage needs onto AWS.

**III. Methodology**

**Data Preprocessing**

scalers

**Target**

**Outliers**

**Missing Data**

**Delete Feature**

**Feature Types Transformed**

**One-hot Features**

**New Features created**

**Features Transformed**

**PCA**

### Implementation

### Refinement

### **Lasso Regression** **(benchmark)**

### **XGBoost (benchmark)**

### **LSTM**

## IV. Results

### Model Evaluation and Validation

### Justification

## V. Conclusion

### Free-Form Visualization

### Reflection

# Findings and questions

# Questions:

# Predicting next day price may not be the best thing to do. We may want to predict price in 5 days, or even much more ahead? What’s a reasonable window to predict price before randomness takes over?

# Should I predict volume?

# Should I include other features to predict price?

# Does dropout improve performance?

# Does multiple layers improve performance?

# Should we actually try to predict price? Or should we predict a range? For example, -1~1% is neutral, 1~5% is good, 5%+ is excellent, etc.

# Findings:

# Best results is with window 15 (about 3 weeks), epoch 2000, and batch-size 50-250.

# There may be upward bias, meaning all that we have seen is upward trend (most of the time).

# When selecting hyperparameters, training predict time, testing predict time, training error eval time, and testing error eval time are negligible.

# The larger the window, the longer the training time. The higher the epoch, and the smaller the batch size, the longer the training time.

# Best training error achieved is with window 15, batch size 10, epoch 500: 2.5616e-06

# Best testing error achieved is with window 10, batch size 500, epoch 2000: 7.58083e-05

### Citation and Sources

**Relevant Files and Folders**