Forecasting Crude Oil Returns using News Sentiment and Machine Learning *

Tianyu Du †

Monday 13th April, 2020

Contents

*Compile Date: 21:26 Monday 13th April, 2020

†tianyu.du@mail.utoronto.ca

1 Introduction

This is the introduction.

2 Data

In order to identify the predictive power of sentiment data on crude oil returns, this study involves three major datasets, a the daily spot price of crude oil at the West Texas Intermediate (WTI) from which returns are computed, ii) a news sentiment dataset from Ravenpack News Analytics (RPNA), and iii) other macroeconomic indicators proxying the overall economic background.

2.1 The West Texas Intermediate (WTI) Crude Oil Dataset

West Texas Intermediate (WTI) is a class of light and sweet crude oil that serves as a benchmark for crude oil prices in the past few decades. Cushing, Oklahoma, where the Cushing oil field locates, has been the delivery point for commodities behind crude oil contracts traded at New York Mercantile Exchange (NYMEX). U.S. Energy Information Administration (EIA) provides a daily time series of spot prices of WTI crude oil delivered from Cushing. This time series can serve as a benchmark of measuring activities in the global crude oil market.

Because of the limited availability of the RavenPack dataset, this paper focuses only on crude oil prices after January 1, 2000. Analysis of the crude oil market Baumeister and Kilian 2016. shows the spot price is highly responsive to news and other macroeconomic shocks, which is exactly the tricky part of forecasting financial time series. If the proposed forecasting algorithm performs well on the crude oil dataset, such an algorithm is conceivably promising on other datasets as well.

2.2 Crude Oil Returns

One side goal of this paper is to identify to what extend machine learning techniques improves existing time series models. Moreover, this paper aims to examine whether machine learning techniques can better extract information from sentiment dataset.

The augmented Dickey-Fuller test on the raw price series gives a p-value of 0.26, which suggests the movement of crude oil prices exhibits significant non-stationarity. This non-stationarity confines classical time series models on this dataset, and makes the above-mentioned comparison between new and classical techniques infeasible. Besides, an accurate prediction of returns is more related to profitability in practice. Therefore, this paper focuses on forecasting returns of crude oil instead of raw prices.

The closing spot prices of crude oils are available at a daily frequency for weekdays only. Besides weekends, observations are missing on certain weekdays when the exchange market is closed. In subsequent sections, this article refers to these days with valid price data as *trading days*.

Table 1 reports dates that are most frequently associated with a missing data over the span of 20 years. The set of days with missing data is consistent over these years: the market is always closed on January 1, July 4 (Independence Day) and December 25 (Christmas). Because the entire dataset ranges from January 3, 2000 to October 31, 2019, missing data problems on December 25 are only reported 19 times in the table. Lastly, the group of dates in late November are responsible for missing data on Thanksgiving holidays since Thanksgiving holiday varies year by year.

Table 1: Top Days with Missing Data

Date	Number of Days with Missing Data
July 4	20
January 1	20
December 25	19
July 3	10
November 23	10
November 24	10
November 25	10
November 22	9
November 26	9

There are only ten weekdays with missing data problem each year on average (3.77% of the entire dataset). The insignificant percentage of missing data allows us to drop those dates without hurting the generalizability of models and experiments in subsequent sections.

On one particular trading day t, let Δ denotes the gap between date t and the previous

trading day, so that $t - \Delta$ is the last trading day before day t. Within a short time period such as the gap between two trading days, this paper assumes crude oil prices exhibits an exponential growth with constant daily growth rate of r_t. So that the following relationship quantifies the relationship between $p_{t-\Delta}$ and p_t:

$$p_t = e^{r_t \Delta} p_{t-\Delta} \tag{2.1}$$

This paper calculates crude oil returns on one particular day t by taking the difference in logged prices at t and the previous trading day and dividing it by the length of duration, Δ:

$$r_t = \frac{\ln(p_t) - \ln(p_{t-\Delta})}{\Delta} \tag{2.2}$$

Equivalently, r_t measures the daily return over the Δ day period. Because returns are closed to zero in most time, in order to avoid decimal issues, this paper converts all r_t's into percentage points.

As mentioned before, the time gap between two observed prices are not uniform. For instance, the return on a Monday can be computed by taking difference between the log close price on Monday and the previous Friday, if available. In this case, $\Delta = 3$. If the previous Friday was a holiday without valid price data, r_t will be $\ln(p_{\text{Mon}}) - \ln(p_{\text{Prev Thu}})$, and $\Delta = 4$.

Table 2: Distribution of Δ by Weekdays

Day of the week	Num. Days.	Num. Trading Days	Δ=1	2	3	4	5
Monday	1,034	931	0	0	887	33	11
Tuesday	1,035	1,023	926	0	0	97	0
Wednesday	1,035	1,027	1,016	5	0	0	6
Thursday	1,035	1,007	999	8	0	0	0
Friday	1,034	990	973	17	0	0	0
Saturday	1,035	0	0	0	0	0	0
Sunday	1,035	0	0	0	0	0	0
Total	7,243	4,978	3,914	30	887	130	17

2 summaries the distribution of Δ values. The Δ values for Mondays are at least 3 because weekend data are always unavailable. One extreme coincident case is that data are

missing on both Monday and Tuesday, so that the Δ value for the coming Wednesday would be 5. This happened in 6 weeks in total. The previous assumption of constant return are only likely to be true within a short time window, and in this kind of rare scenarios, the assumption becomes less convincing.

The movement of crude oil returns in the past two decades has exhibited volatile patterns. Figure 1 plots the pattern of returns, in which shaded areas indicate U.S. recessions (March 2001 to November 2001 and December 2007 to June 2009). a

cite data

Add more descriptions on the chronology.

Figure 1: Crude Oil Returns

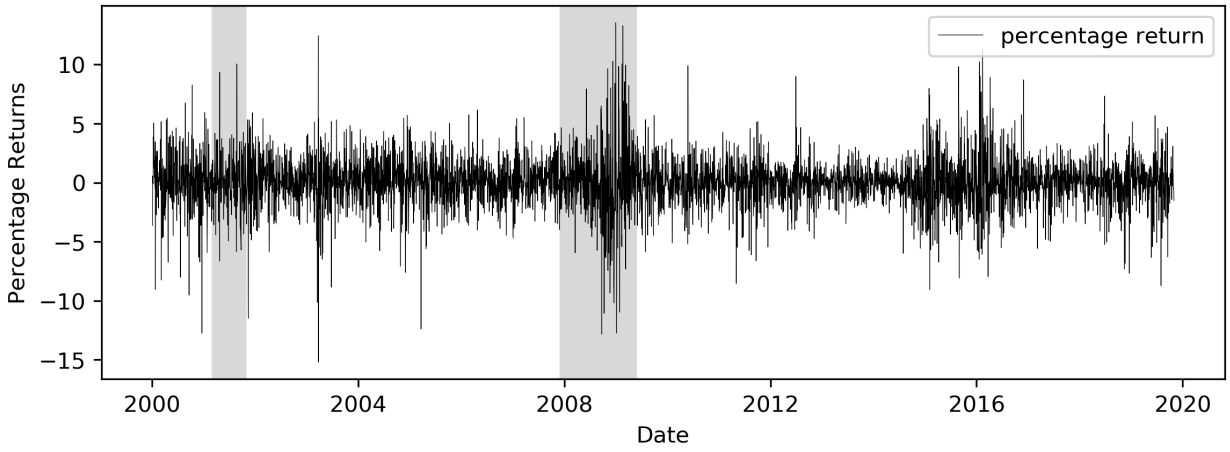


Table 3 provides summary statistics for the percentage crude oil returns, in which normalized skewness and excess kurtosis are defined as

$$\hat{m_3} := \frac{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^3}{\sigma_x^3} \text{ (normalized skewness)} \tag{2.3}$$

$$\hat{m_4} := \frac{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^4}{\sigma_x^4} - 3 \text{ (excess kurtosis)} \tag{2.4}$$

$$\text{where } \sigma_x := \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2} \tag{2.5}$$

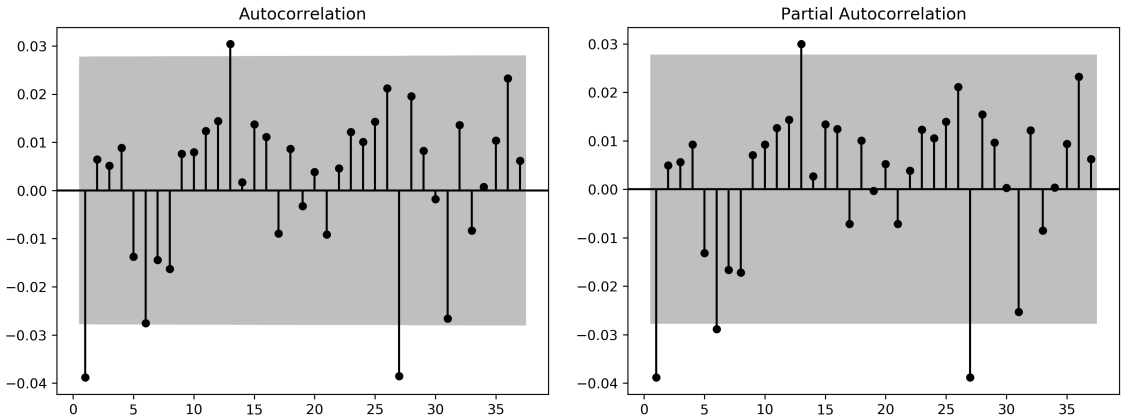
Summary statistics in the table suggest the mean return within each year are nearly zero, which agrees the conventional expectation that returns are zero on average. During periods of recessions, the average returns are below -0.2%,. Moreover, during same period, the

series becomes significantly more volatile as well. Given the high kurtosis between 2008 and 2009, one are more likely to encounter extreme returns, both positive and negative, during recession periods.

Table 3: Summary Statistics for Crude Oil Returns (Percentages)

Year	Obs.	Mean	Median	Std.	Min	Max	Normalized Skewness	Excess Kurtosis
2000	249	0.03433	0.20148	2.61996	-12.74152	8.26343	-0.92174	3.45580
2001	250	-0.02409	-0.04434	2.54058	-11.48581	10.05107	-0.06444	3.15304
2002	250	0.15535	0.15221	1.70283	-5.86460	5.43272	-0.22297	0.62431
2003	250	0.07861	0.13203	2.57315	-15.19090	12.44253	-0.89439	7.30189
2004	249	0.08918	0.11605	2.08792	-7.60501	5.70121	-0.38117	1.01395
2005	251	0.05257	0.11019	1.96717	-12.39009	5.02715	-1.04498	5.84007
2006	249	-0.00539	0.12995	1.58949	-4.45214	6.15402	0.13487	1.03258
2007	252	0.23400	0.09798	1.69800	-4.66915	5.51381	0.13705	0.65946
2008	253	-0.29945	-0.07920	3.34992	-12.82672	13.54551	-0.01650	2.60308
2009	252	0.26537	0.19157	2.92040	-12.74310	13.29544	0.29333	4.25972
2010	252	-0.02077	0.03198	1.74554	-5.18874	9.89802	0.39313	3.82001
2011	252	0.00583	0.10994	1.94170	-8.53498	5.18170	-0.69170	2.27400
2012	252	-0.04164	0.03600	1.51078	-4.76060	9.00091	0.54820	5.53225
2013	252	0.01455	0.04489	1.06690	-3.46951	3.20999	0.05495	0.67398
2014	252	-0.16510	-0.05343	1.36052	-5.98638	4.91592	-0.76983	3.16348
2015	252	-0.03610	-0.25616	2.63361	-9.05140	9.81397	0.24129	1.25225
2016	252	0.20931	0.00000	2.79698	-7.95603	11.28922	0.70466	2.11826
2017	250	0.06564	0.17286	1.40987	-5.56187	3.32016	-0.87368	2.07271
2018	249	-0.10076	0.07393	1.81925	-7.67683	7.33414	-0.64252	3.38603
2019	210	0.04359	0.10073	1.93931	-8.72444	5.67862	-0.66251	2.87153
Total	4978	0.02754	0.06307	2.15250	-15.19090	13.54551	-0.16152	5.12757

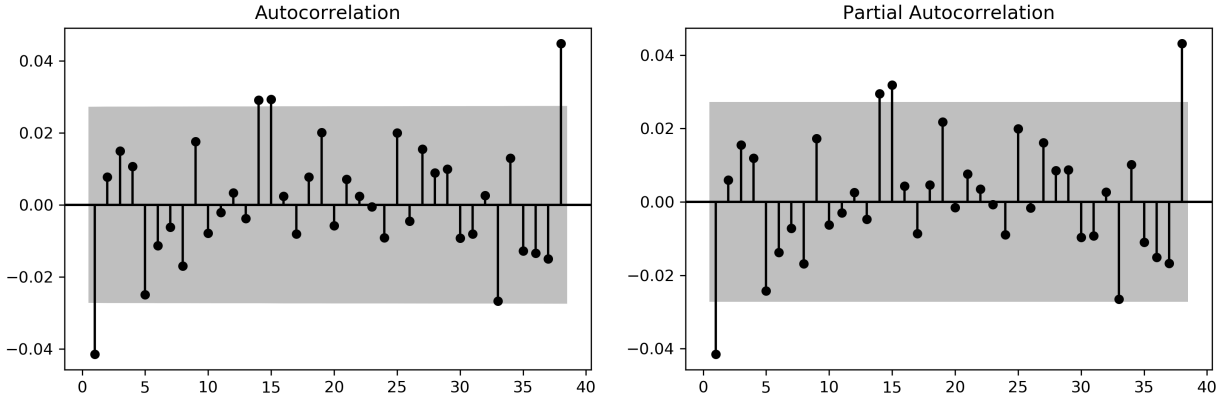
Figure 2: ACF and PACF for Crude Oil Returns (missing data dropped)



The autocorrelation function (ACF) and partial autocorrelation function (PACF) plots in figure 2 explore the inter-temporal correlation within the return series. Since only a few lags are statistically significant in the ACF and PACF plots, we do not expect linear time series models are capable to achieve high performances in this return prediction task.

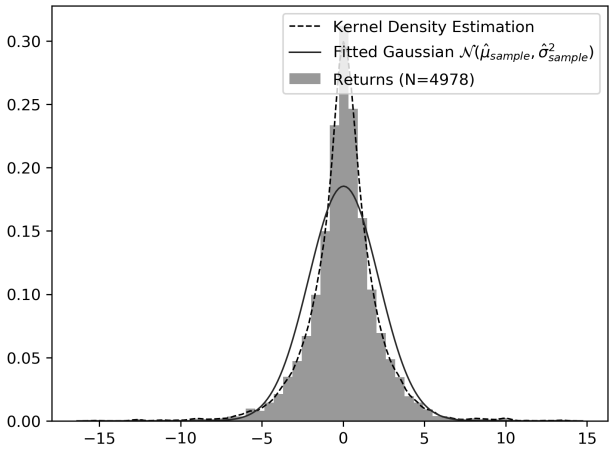
It is worth noticing that lag 1, 6, 13, 27 are significant in both ACF and PACF plots, which may indicate seasonalities with period of one week. However, regularity in missing data can lead to this observation as well. Hence, instead of dropping days with missing data, we fill up these missing data using random values from a Gaussian distribution parameterized by the mean and variance of the entire dataset. Figure 3 plots the ACF and PACF of return series with missing values filled using random noise, the significance at lag 6 disappeared, but the significance of bi-weekly lag persists and another spike at lag 36 emerges. This observation indicates that there might be seasonality with bi-weekly periods. In the experiment section, we are going to examine seasonal models with both period lengths.

Figure 3: ACF and PACF for Crude Oil Returns (missing data filled)



The histogram in figure 4 suggests that the empirical distribution of crude oil returns is much clustered near zero than a Gaussian distribution. With this clustering feature, conventional metric for evaluating regression models, such as mean squared error (MSE), will not be sufficient in this task. For instance, a dummy model consistently predicting zero will attain a fair MSE (to be specific, the variance of entire dataset). Therefore, in later sections, we introduce another directional accuracy to assess the fitness of models.

Figure 4: Distribution of Crude Oil Returns



2.3 Day of the Week Effect in Crude Oil Dataset

2.3.1 Difference in Returns across the Week

Gibbons and Hess examined returns on stocks from S&P 500, Dow Jones 30, and Treasury Bills. They found strong negative mean returns on Monday compared with other weekdays (1981). The seasonality persisted even after taking market adjustment measures, such as using mean-adjusted returns instead (Gibbons and Hess 1981). Analysis in this paper unveils a similar daily seasonality presents in crude oil returns as well. Panels in figure 5 demonstrate the empirical distributions of returns on each day of the week. Ns within parentheses in captions denote the number of observations. We can see that Mondays and Wednesdays have relatively larger variances, which again matches Gibbons and Hess' observations.

Figure 5: Distributions of Returns on Each Day of the Week

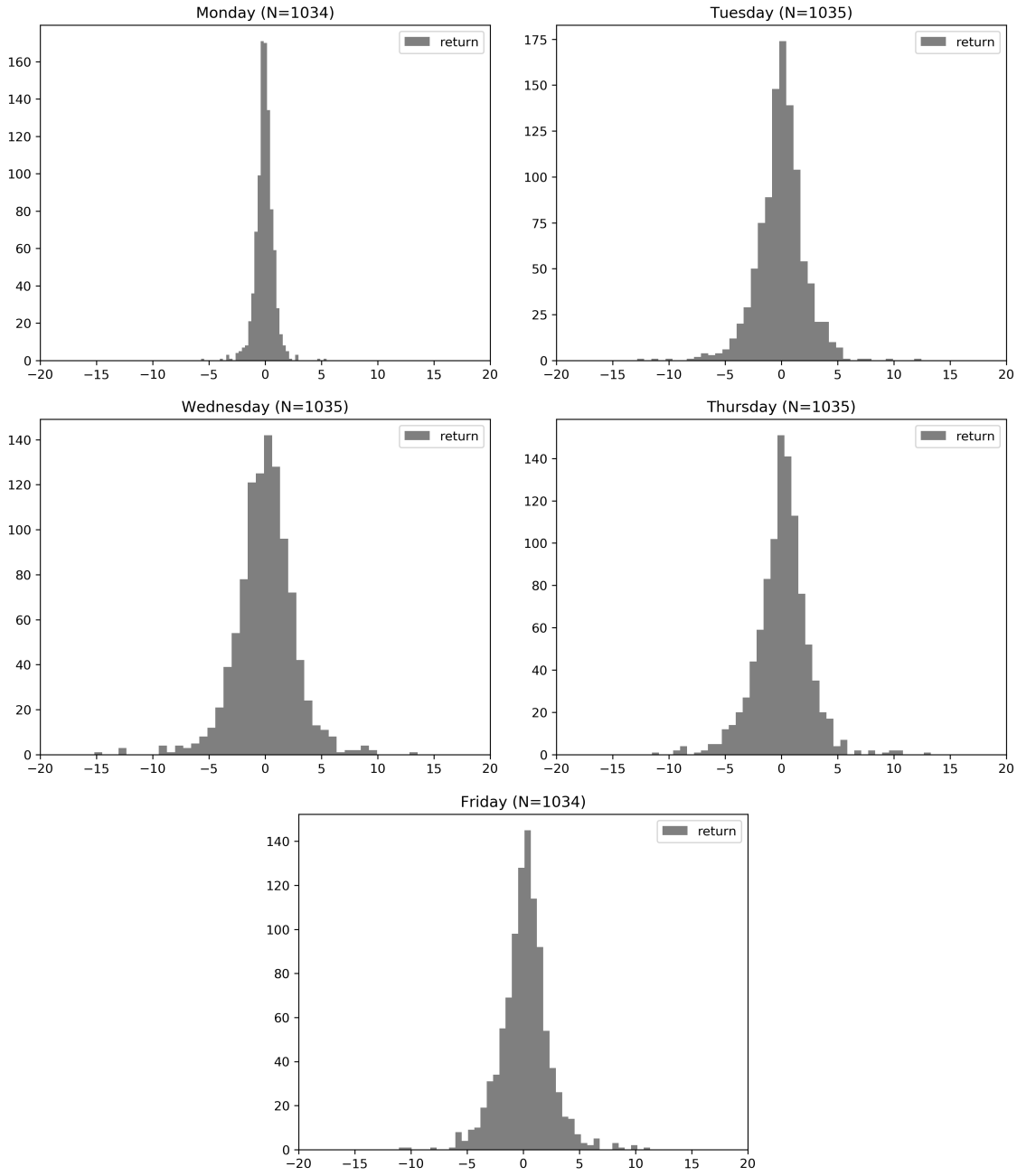


Table 4 below provide summary statistics for prices and returns on each day. [1] It turns out that at a significance level of 0.05, Monday and Friday are the only two weekdays with a mean return significantly different from than zero. And the t-test suggests Mondays are

[1] In table 4, a value of -0.000 indicates a negative value with magnitude less than 0.0005. P-values are calculated in a two-tailed t-test with null hypothesis $\mu_0 = 0$. Bold fonts indicate statistically significance at level $\alpha = 0.05$.

more likely to associate with negative returns, meanwhile, Friday is more often associated with positive returns.

Table 4: Summary Statistics of Crude Oil Returns on Each Day of Week

Day of the week	Num. Obs.	Mean (P-Value)	Std.	Normalized Skewness
Monday	931	**-0.055 (0.042)**	0.816	-0.134
Tuesday	1,023	-0.034 (0.615)	2.141	-0.335
Wednesday	1,027	-0.000 (0.998)	2.660	-0.325
Thursday	1,007	0.069 (0.361)	2.378	-0.041
Friday	990	**0.155 (0.026)**	2.194	0.128
Total	4,978			

2.3.2 Kolmogorov-Smirnov test for Distributional Similarities

Smirnov developed a non-parametric method of testing the equality between two continuous distributions, with CDFs $F(x)$ and $G(x)$ respectively (1939). Hodges' work provided more details on the Kolmogorov-Smirnov test and relevant methods (1958). I am using the two-tailed version of Kolmogorov-Smirnov test to check whether distributions of two different days are similar. Given two datasets, take returns on Mondays and Tuesdays for example, the null hypothesis says those two datasets are drawn from the same distribution, and the alternative says they are from different distributions. [2] Firstly, the Kolmogorov–Smirnov test constructs the empirical CDFs $F_{Mon,927}(x)$ and $F_{Tue,1018}(x)$ from the dataset. Then, the Kolmogorov–Smirnov statistic measures the maximum discrepancy between two distribution functions, which is

$$D := \sup_x |F_{Mon,927}(x) - F_{Tue,1018}(x)| \in [0,1] \tag{2.6}$$

A smaller D-statistic implies stronger distributional similarity between two distributions. For instance, when $F_{Mon,927}(x)$ and $F_{Tue,1018}(x)$ are exactly the same, the D-statistic is zero. In contrast, let $X = 0$ and $Y = 1$ be two "deterministic" random variables, in this case,

[2] Different alternative hypotheses can be used in Kolmogorov–Smirnov test: i) $H_1 : F(x) \geq G(x)$, ii) $H_1 : F(x) \leq G(x)$, and iii) $H_1 : F(x) \neq G(x)$. This paper is using the third (two-tailed) alternative hypothesis.

there distributions are completely different, and $D_{X,Y} = 1$.

The test rejects H_0 at a significance level of α if

$$D > \sqrt{-\frac{1}{2}\ln\frac{\alpha}{2}}\sqrt{\frac{n+m}{nm}} \tag{2.7}$$

where m and n denote sizes of two datasets.

Table 5: D-Statistics in Kolmogorov-Smirnov Tests

D-Statistic (P-Value)	Monday	Tuesday	Wednesday	Thursday	Friday
Monday	0.000(1.000)	**0.193(0.000)**	**0.243(0.000)**	**0.189(0.000)**	**0.180(0.000)**
Tuesday		0.000(1.000)	0.064(0.030)	0.064(0.030)	**0.071(0.010)**
Wednesday			0.000(1.000)	0.058(0.062)	**0.084(0.001)**
Thursday				0.000(1.000)	0.030(0.729)
Friday					0.000(1.000)

Table 6: The Kolmogorov-Smirnov D-Statistic for all pairs of distributions. Bold font indicates the null hypothesis is rejected at a significance level of 0.01, which implies discrepancy in distributions.

Table 5 presents the Kolmogorov-Smirnov D-Statistic for distributions of every pairs of days. At a significance level of 0.05, we can see that Mondays follow a distribution significantly different from distributions of other weekdays follow. Because the dataset does not contain weekend data, returns on Mondays is always computed using the difference between log prices on Monday and the previous Friday (Thursday if Friday is not a trading day and so on). Therefore, returns associated with Mondays pick the weekend effect. In fact, the distribution of returns on Mondays (over weekends) is the only one with negative mean among distributions of all five days.

2.4 News Sentiment Dataset

The event sentiment dataset from RavenPack News Analytics (RPNA) tracks and analyzes all information of companies, organizations, countries, commodities, and currencies from four major sources: Dow Jones Newswires, Wall Street Journal, Barron's and MarketWatch. This dataset covers events from January 1, 2000, to October 30, 2019. RavenPack records the exact date and coordinated universal time (UTC) when each news is published. Since the

crude oil prices are from New York Exchange (NYEX), which uses US Eastern time, this UTC time is later converted into Eastern time. For each piece of news, the dataset links it to a unique entity name attribute. To filter out noise data irrelevant to crude oil returns, this paper selects the subset of news with crude oil topic as the main source of news. It turns out that there are 106,960 entries from the original dataset left, lead to 17 events per day on average.

Table 7 provides summary statistics of numbers of daily news arrival by years.

Add more here

Table 7: Summary Statistics for Daily News Arrival by Years

Year	Mean	Median	Std.	Min	Max	Normalized Skewness	Excess Kurtosis
2000	8.665	8.000	8.315	0.000	48.000	1.228	1.939
2001	11.493	11.000	10.247	0.000	51.000	0.682	0.059
2002	3.542	3.000	3.642	0.000	19.000	1.403	2.368
2003	5.126	3.000	6.145	0.000	39.000	2.058	5.646
2004	20.776	19.000	17.680	0.000	84.000	0.728	0.193
2005	17.473	17.500	13.796	0.000	57.000	0.403	-0.460
2006	18.615	19.000	14.272	0.000	58.000	0.247	-0.862
2007	16.781	16.000	13.669	0.000	66.000	0.567	-0.187
2008	20.500	22.000	15.141	0.000	66.000	0.304	-0.562
2009	14.499	14.000	10.988	0.000	48.000	0.296	-0.761
2010	15.564	17.000	11.437	0.000	52.000	0.247	-0.753
2011	19.187	20.000	14.175	0.000	65.000	0.231	-0.610
2012	20.077	22.000	14.682	0.000	65.000	0.206	-0.688
2013	14.526	15.000	11.364	0.000	57.000	0.413	-0.374
2014	13.353	11.000	13.445	0.000	69.000	1.502	2.596
2015	18.663	18.000	15.974	0.000	80.000	0.738	0.188
2016	19.956	18.000	17.454	0.000	101.000	0.837	0.661
2017	12.479	11.000	10.927	0.000	58.000	0.797	0.619
2018	13.277	13.000	11.490	0.000	93.000	1.350	5.481
2019	10.505	9.000	10.608	0.000	65.000	1.067	1.569

In general, weekends are quiet period of news arrival, while much more news arrive in the middle of each week. Figure 6 below summarizes the average numbers of news on each day of the week, and table 8 summarizes the average number of news on each day in each year.

Stopped here

Describe more, add references

Figure 6: Average Numbers of News on Each Day of Week

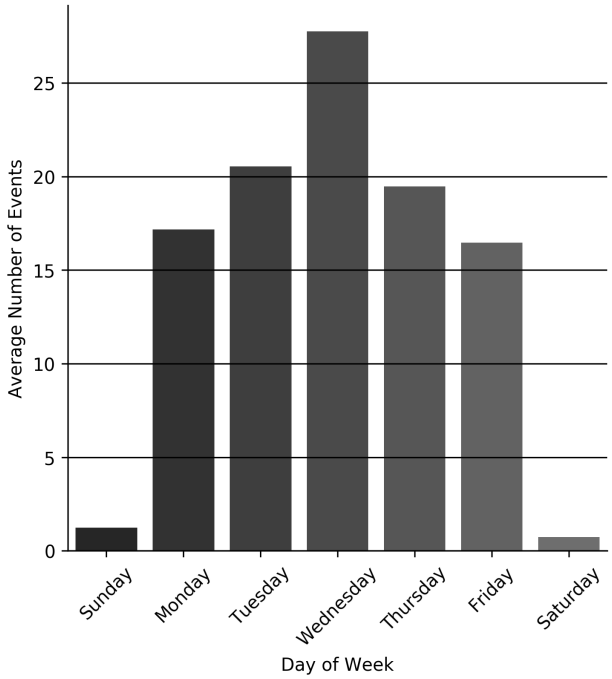


Table 8: Average Numbers of News on Each Day of Week in Each Year

Year	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
2000	11.157	14.135	13.077	11.885	9.769	1.643	1.500
2001	12.547	17.569	21.327	15.058	14.078	1.000	1.200
2002	5.771	5.019	5.224	3.980	5.469	1.200	1.600
2003	7.080	6.529	9.942	6.863	5.490	1.200	1.136
2004	24.058	28.981	39.250	28.660	22.302	2.182	2.240
2005	21.462	21.846	33.596	24.654	19.000	1.765	2.259
2006	22.981	24.885	35.904	24.846	19.731	1.346	2.161
2007	19.792	21.385	33.577	23.846	16.769	1.941	2.212
2008	24.788	26.415	36.415	26.269	25.250	2.207	3.065
2009	16.058	21.346	29.192	16.925	15.538	1.688	2.366
2010	16.327	23.058	28.654	20.596	17.135	2.261	2.932
2011	23.769	28.577	32.904	25.750	19.942	2.053	3.441
2012	22.340	26.654	36.423	26.981	25.118	3.783	2.756
2013	16.673	19.642	28.588	19.038	15.846	2.500	2.366
2014	15.510	18.846	25.113	16.923	15.529	2.167	2.467
2015	23.019	27.135	35.558	23.189	19.843	2.091	2.957
2016	23.333	29.192	38.462	24.808	23.077	2.190	2.105
2017	14.220	16.788	25.192	16.077	14.039	1.696	1.667
2018	13.654	19.059	24.712	18.635	15.235	2.586	2.143
2019	11.263	15.872	24.600	15.026	13.795	1.923	1.500

2.4.1 Event Sentiment Scores

To estimate the potential economic impact upon news arrival and afterwards, Ravenpack assigns each piece of news an **Event Sentiment Score** (ESS) between 0 and 100 using an algorithm combines results from surveying financial experts and pattern matching. An ESS of 100 indicates extreme positive short-term positive financial or economic impact. In contrast, an ESS with zero value indicates extreme negative impact. And a ESS of 50 indicates exact neutral news, which indicates noise.

For simplicity, raw scores (range from 0 to 100) are normalized by subtracting 50, so that the sign of normalized ESS matches the nature of news, and a zero score represents a neutral news.

The first panel in figure 7 plots the distribution of (normalized) ESS for all news about crude oil, while second and third panels focus on two tails of the distribution. From the histogram in figure 8, one can see that only a small portion of news is purely neutral with zero ESS (3,479 events, 3.25% of the entire dataset). Moreover, ESS scores of most news are clustered around -15 (39,347 and 36.8%) and 18 (34,574 and 32.3%). It turns out that these news are simply objective reports of past price/return movements of crude oil commodities and futures. Therefore, we do not expect these news to provide as much information on predicting returns as other breaking news like OPEC export restrictions. In order to emphasize fresh events other than reports of past price movements, models proposed in this paper will focus on extreme events by assigning them higher weights. Specifically, models are designed to pay more attention to news carrying sentiment scores with high absolute values, meanwhile, models actively discriminate news whose sentiment scores are near zero.

Add literature here

Figure 7: Distribution of Event Sentiment Scores (ENS)

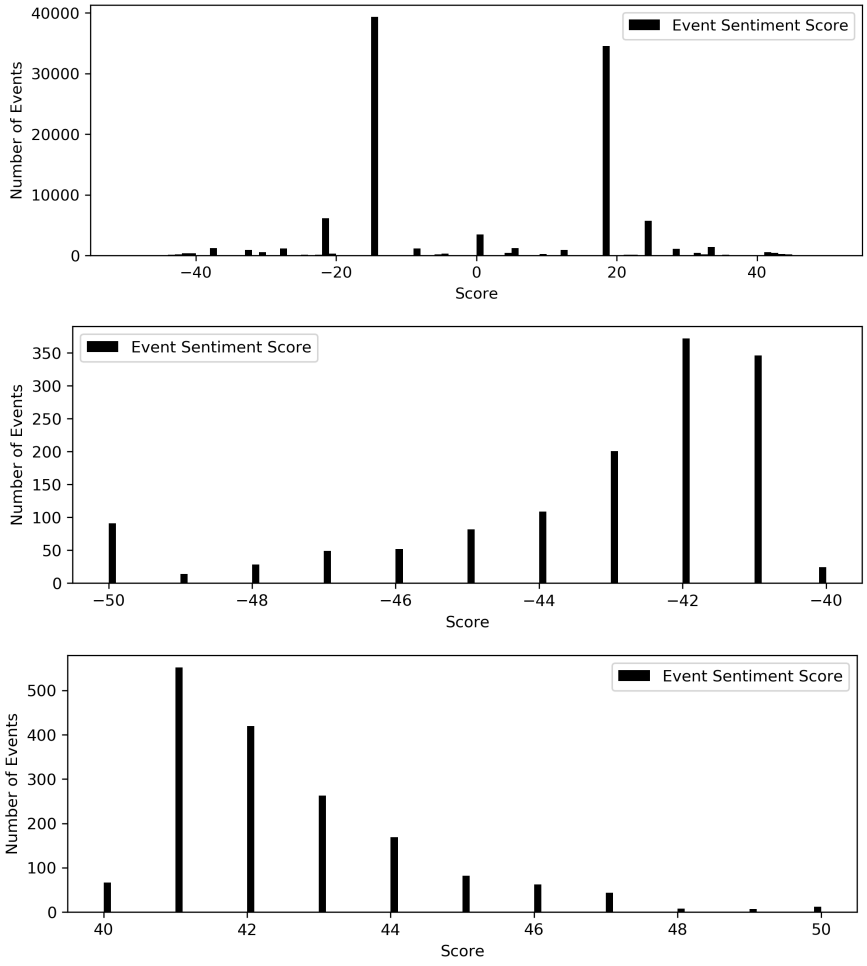
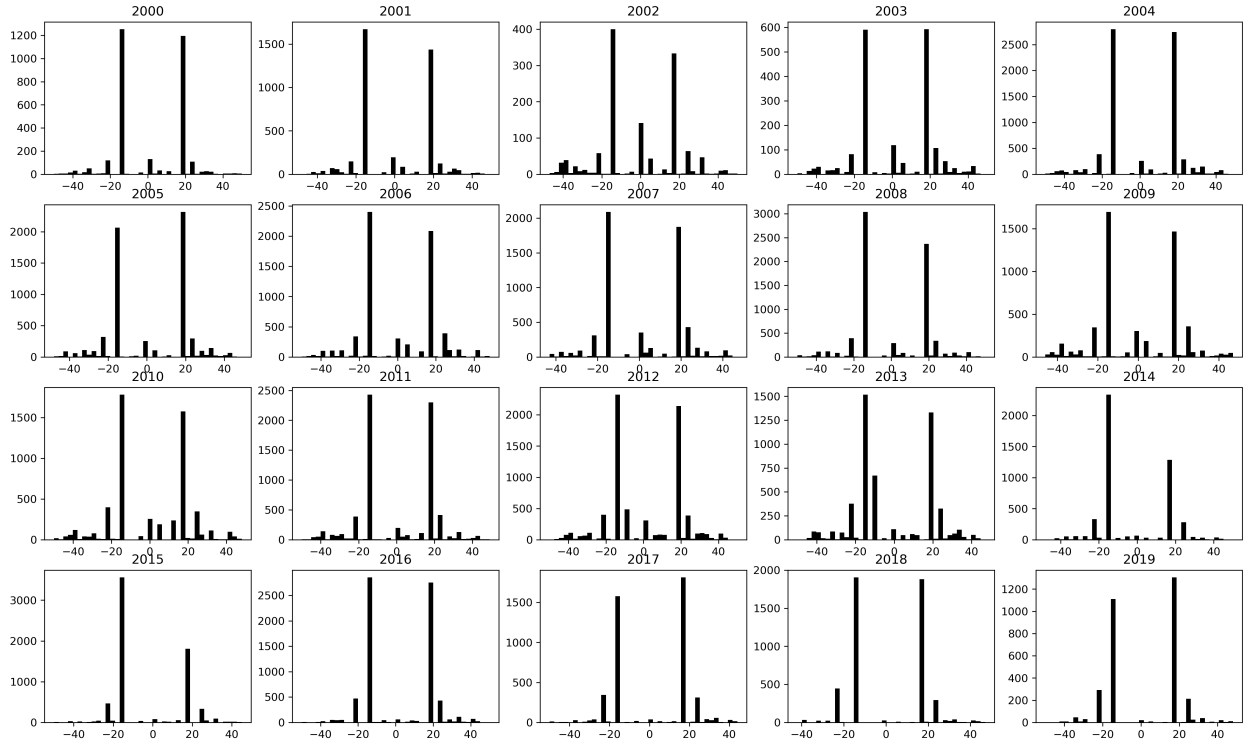


Figure 8 plots the distribution of ESS scores in each year. The pattern of clustering around -15 and 18 is pretty consistent over the span of 20 years: the majority of news are simply reporting the crude oil market instead of events outside the market.

Figure 8: Distribution of Event Sentiment Scores (ENS) each Year



2.4.2 Weighted Event Sentiment Scores

Often time, different news sources report the same event so that there are duplicate entries about the same event in this dataset, which aggravates the problem of noise. RPNA dataset computes an **Event Novelty Score** (ENS) to measure how novel a news story is within a 24-hour period.

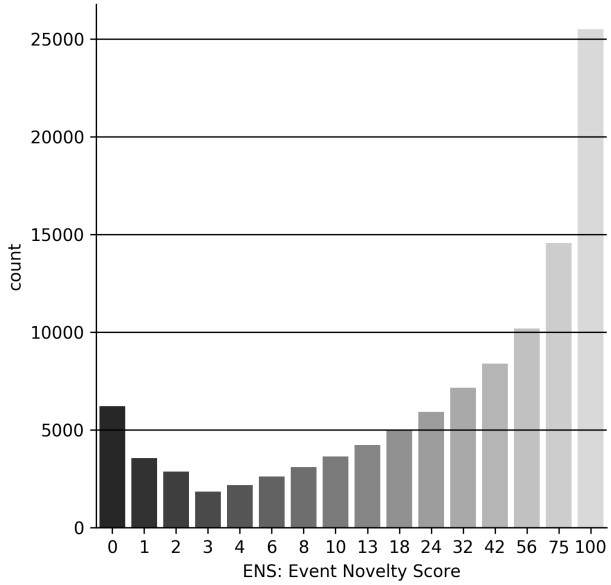
Suppose that OPEC announces an export resections after a conference finished at 11:00 a.m. January 10. After 30 minutes (11:30 a.m., January 10) one news source reports this export cut, and this news article is one entry in the dataset. To determine the ENS of this news article, the algorithm looks into the 24-hour period prior to the news arrival, that is, from 10.30 a.m. January 9 to 10.30 a.m. January 10. If there is no news about this export restriction in this period, then this news article is the first report of this export cut event and receives an ENS score of 100. Otherwise, if there are another two articles about the same events published before this articles, this article receives a decayed novelty score of 56 instead.

In general, the ENS decays exponentially as there are more news reporting the event. For an arbitrary news article i, if there are another k articles of the same topic published within the 24-hour period before article i arrives, article i is therefore the $k+1^{th}$ articles on this topic and would receive a novelty score of

$$\text{ENS}_i = 100 \times 0.75^k \tag{2.8}$$

Figure 9 plots the distribution of ENS, the histogram suggests that most news have relatively high novelty scores.

Figure 9: Distribution of Event Novelty Score



To address the duplication issue, this paper constructs an alternative metric of sentiment, **Weighted Event Sentiment Score** (WESS), from both ESS and ENS.

$$\text{WESS} := \frac{\text{ESS} \times \text{ENS}}{100} \tag{2.9}$$

We divide the product of ESS and ENS in equation (2.9) by 100 so that WESS ranges from -50 to 50 as well.

The constructed WESS scores have several advantages for modelling. Firstly, WESS discriminates against duplicate news articles. For example, if one extreme negative event

with an ESS of - 50 happened, many sources report this event within 24 hours after it happened. The sum of ESS of all these news would overestimate how bad the scenario is because the negative event only occurs once but it is reported for several times. Weighting ESS of articles using their novelty scores helps mitigate this problem so that WESS allows models to pay more attention on novel news rather than redundant ones. Secondly, WESS preserves the sign of ESS, so that an event carries positive sentiment, in terms of ESS, if and only if its WESS score is positive.

The histograms in figure 10 illustrate the overall distributions of WESS as well as the two tails of it. It turns out that the clustering pattern in figure 7 disappears and much more news are now with zero sentiment scores. Therefore, WESS provides a stricter filter to filtering out noises (i.e., news with zero sentiment scores) and better helps models to focus on meaningful news only.

Figure 10: Distribution of Weighted Event Sentiment Scores (WESS)

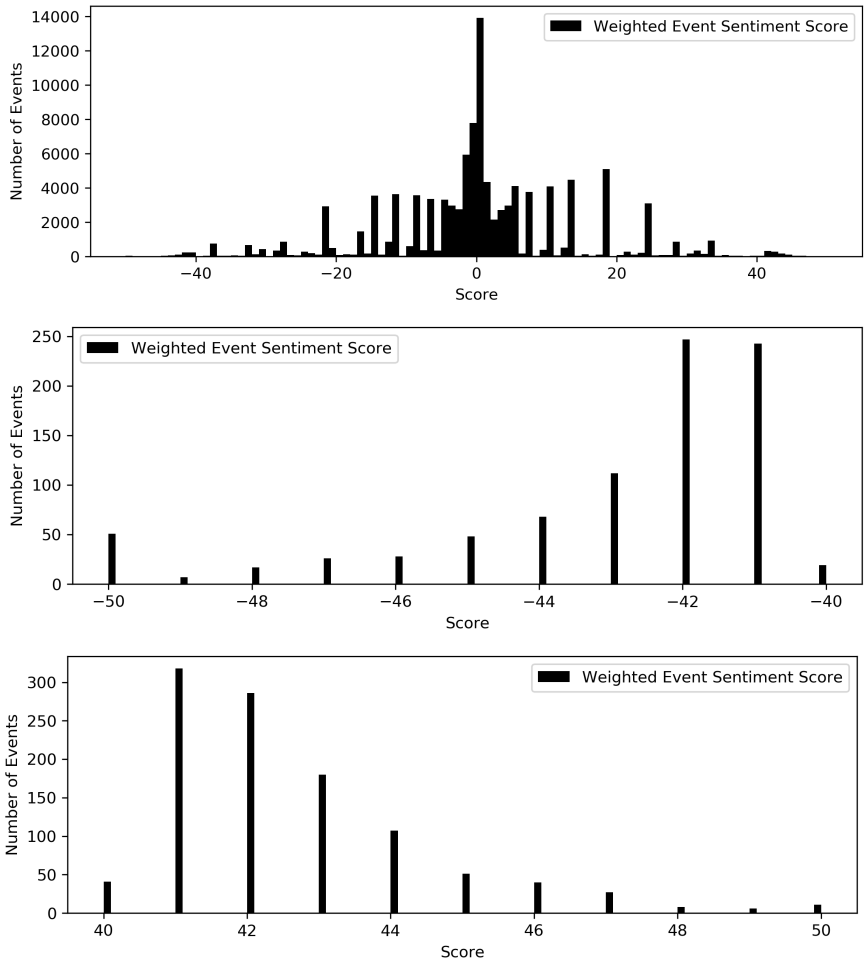
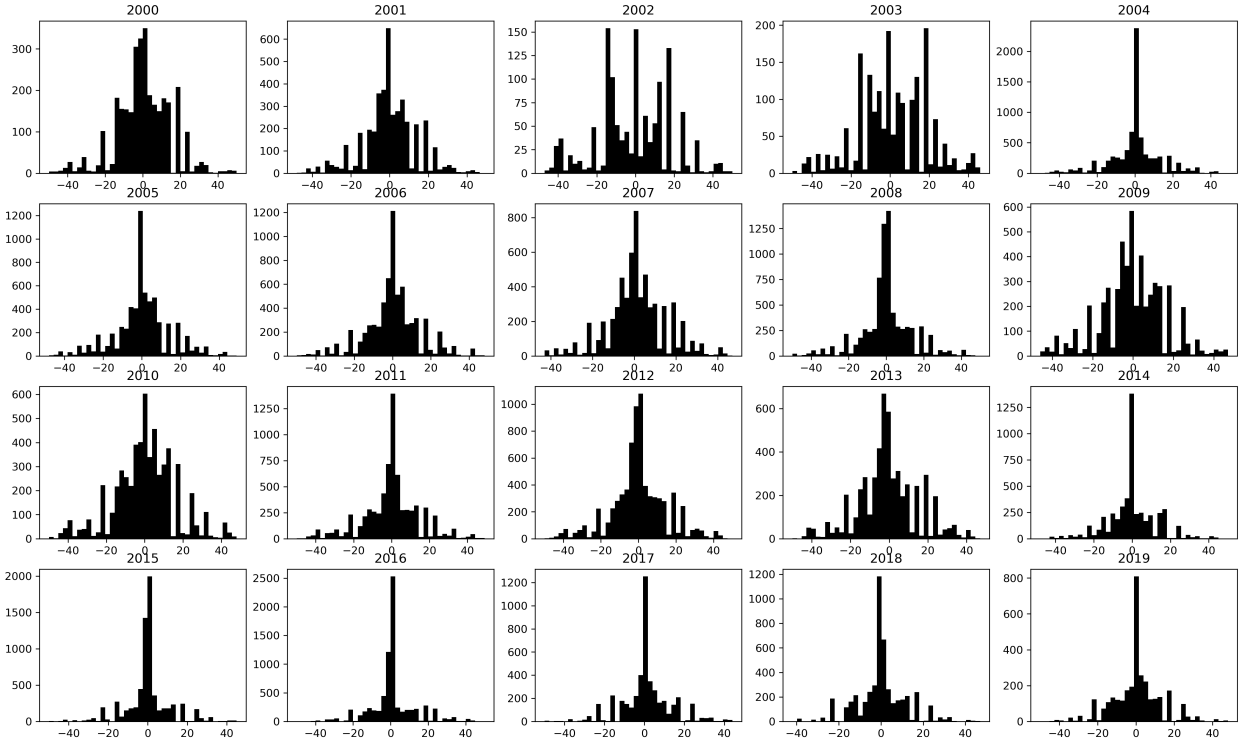


Figure 11 plots the yearly distributions of WESS. The yearly distributions suggest that there are more events with negative sentiments in 2001 as well as in 2008-2009, such observation matches the US recession records in figure 1.

Figure 11: Distribution of Weighted Event Sentiment Scores (WESS) each Year



2.4.3 News Arrival Time

Add citations here

The numbers of news arrived are not evenly distributed across the timeline, there are always busy hours as well as quiet hours. Figure 12 summarizes the average number of news arrives on each day over the period of 20 years. The trench at the end of February corresponds to leap years. Other trenches are in general correspond to holidays, for example, average numbers of news on the Independence Day and Christmas are significantly less than other days.

Figure 12: Average Number of Events on Each Day

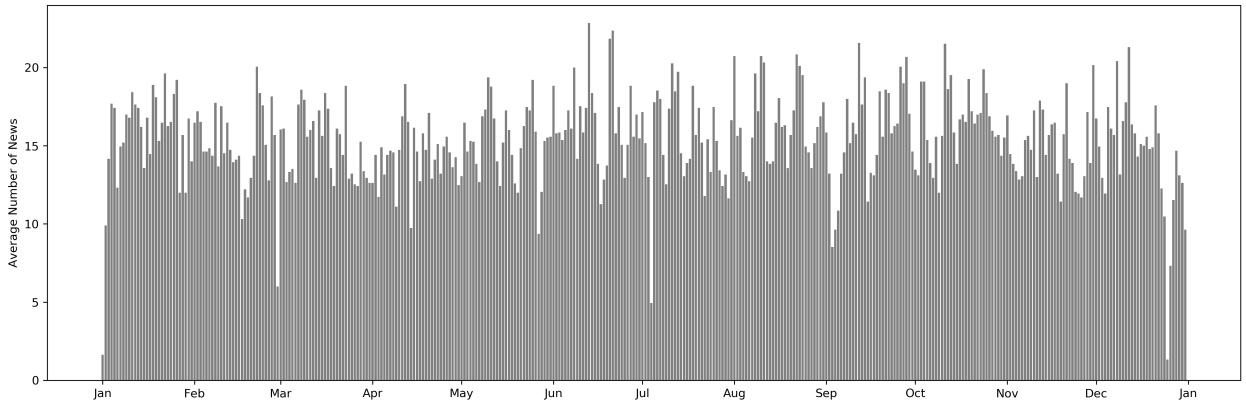
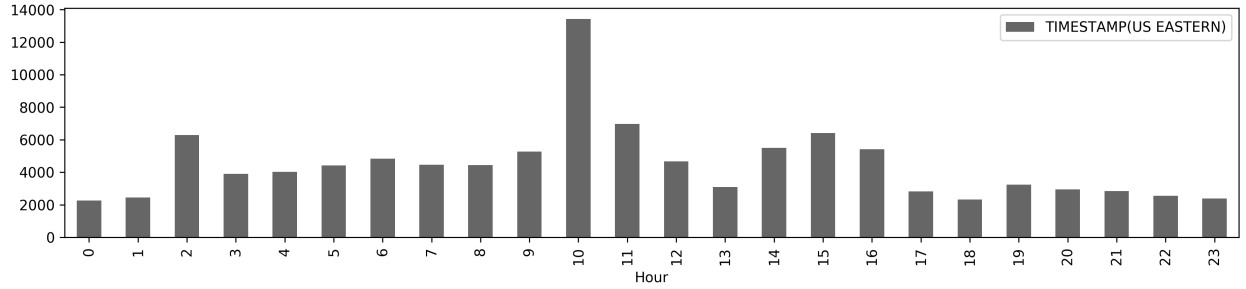


Figure 13 has a look at the distribution of news arrival within 24 hours. It is worth noticing that, in the RPNA dataset, the original timestamps recording when news arrives are using Universal Time Coordinated (UTC). To incorporate the crude oil dataset, we convert raw timestamps to Eastern Standard Time (EST) timezone[3], where crude oil commodities are traded. From the distribution of news arrival, one can see that most news arrive during day time between 10:00 and 16:00. There is an unusual spike at 2:00, this could correspond to morning news at 7:00 in British. But because all four news sources in RPNA dataset are U.S. based publishers, the news arrival process is quiet again between 3:00 and 9:00 as less reporters are actively writing during this time.

Figure 13: Total Number of News Arrived

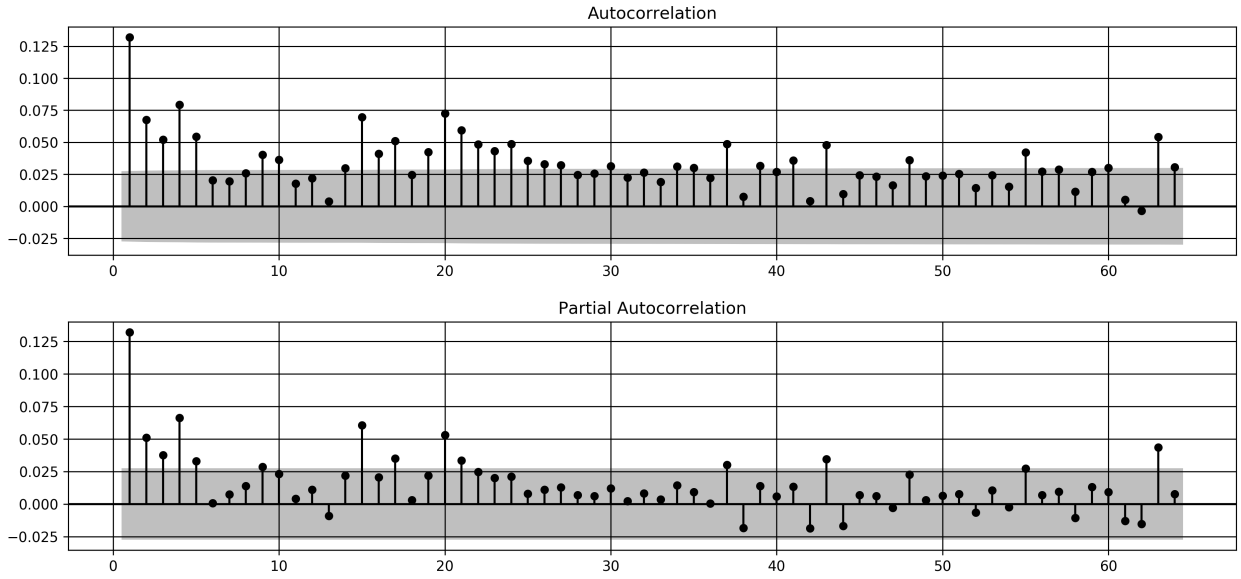


In order to have a closer look at the intertemporal correlation of event sentiment, this

[3]EST is five hours behind UTC during autumn and winter. During spring and summer (daylight saving time), EST is four hours behind UTC.

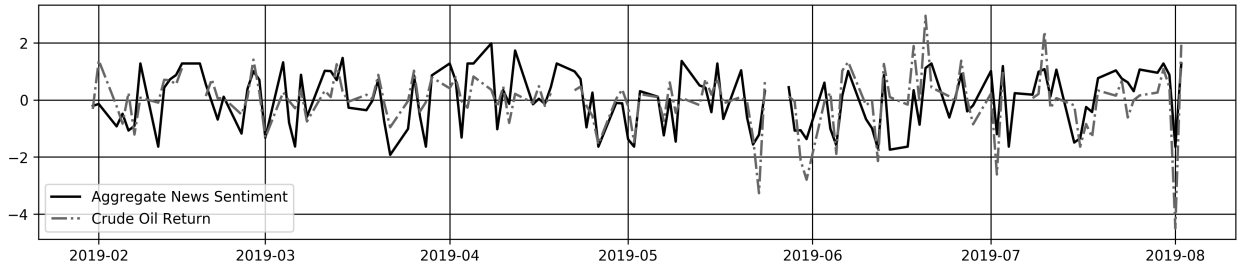
paper firstly compute the mean event sentiment score of all events within each day, denoted as $\overline{\text{ESS}}$ and $\overline{\text{WESS}}$ respectively, in the span of 20 years. The ACF and PACF plots of the daily average ESS in figure 14 suggest the intertemporal correlation here is much more salient than the series of returns, which has only a few significant lags.

Figure 14: ACF and PACF of $\overline{\text{ESS}}$



Moreover, there exists significant correlation between the price movement series and news sentiment. Figure 15 plots the trends of daily average sentiment and crude oil returns in 2019, in which these two series have shown significant co-movement pattern. It turns out that the Spearman correlation between these two series is 0.562 with p-value zero.

Figure 15: Movements of $\overline{\text{ESS}}$ and Return in 2019



This co-movement provides justification of using the series news sentiment to predict

crude oil returns.

2.5 Classifying News Type

Based on the distributions shown in previous sections, we shall see that a great number of events carry nearly natural sentiment or are just description of past price movement. This paper wishes to allow models to differentiate different types of news instead of taking the average sentiment score of all news. As seen in the histograms of sentiment scores (figure 7 and 10), the distributions are pretty much symmetric about zero, therefore, for simplicity, this paper assumes the region of neutral news to be symmetric around zero. Specifically, the classification procedure firstly determines a radius $r \geq 0$. Afterward, the algorithm classifies all news based on their (weighted) event sentiment scores. News with score (W)ESS $\in [-50, -r)$ are negative news, and all news with (W)ESS $\in (r, 50]$ are positive news, and, news in $[-r, r]$ are neutral news. Figure 16 and figure 17 plots the composition of news types while classifying these news using two criterions, event sentiment scores and weighted event sentiment scores.

Figure 16: Composition of News Type based on ESS

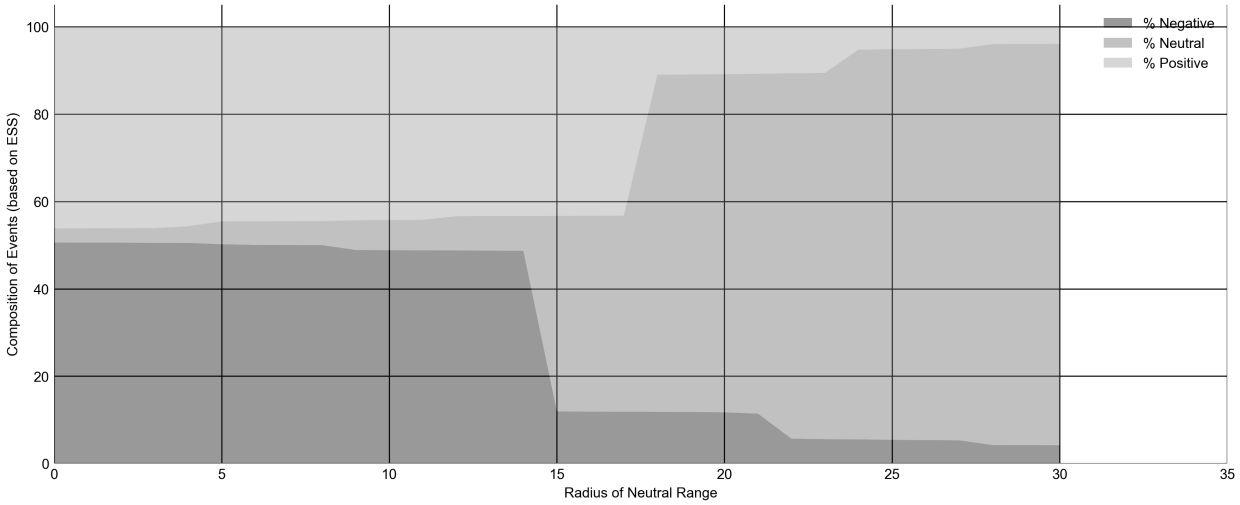
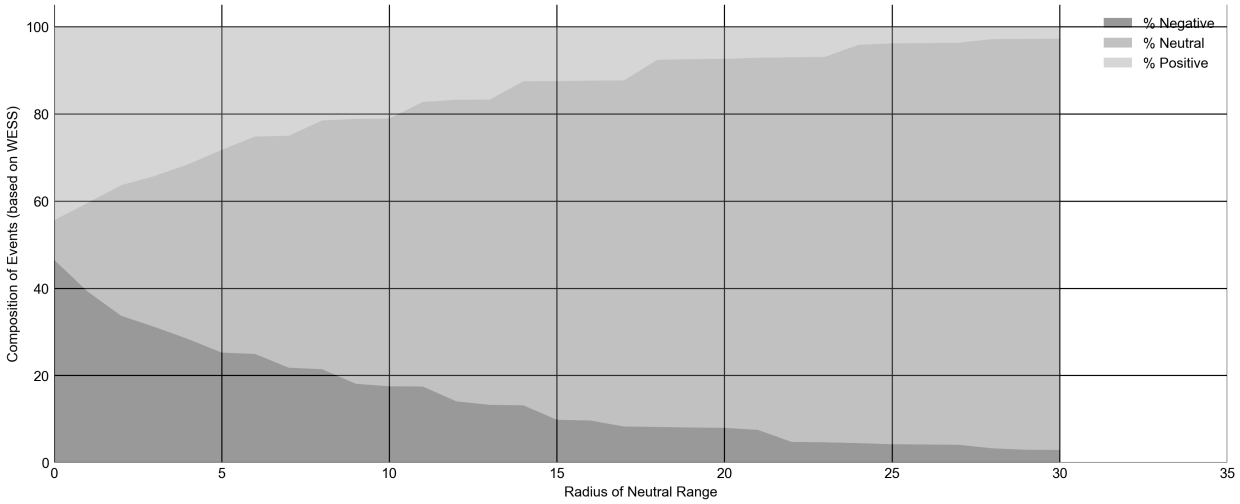


Figure 17: Composition of News Type based on WESS



In figure 16, the two sharp breaking points at $r = 15$ and $r = 18$ correspond to the two clusters of events with sentiment scores 35 and 68 observed in figure 7. Table 9 and table 10 summarizes how the portions of different classes of news change while applying different value of threshold.

The entry 89.43% (2,749.47%) in table 9 (last row, second column) means 89.43% of news are classified to be neutral using a threshold $r = 25$. Moreover, the number of neutral news under this threshold is 2,749.47% of the the number of neutral news under threshold $r = 0$. In the appendix, table 31 and 32 provide a more complete summary on the composition of news under various thresholds r. The threshold variable r is a hyper-parameter in our model, the optimal classification threshold depends on specific type of models used. In most experiments, we are using $r = 0$ for ESS scores and $r = 0.3$ for WESS scores.

Table 9: Composition of News Classes with Different Thresholds on ESS Scores

r	Num Negative	Num Neutral	Num Positive
0	50.59% (100.00%)	3.25% (100.00%)	46.15% (100.00%)
0.3	50.59% (100.00%)	3.25% (100.00%)	46.15% (100.00%)
1	50.57% (99.96%)	3.29% (101.24%)	46.13% (99.96%)
3	50.52% (99.85%)	3.39% (104.08%)	46.09% (99.87%)
5	50.20% (99.23%)	5.24% (161.14%)	44.55% (96.54%)
10	48.84% (96.53%)	6.91% (212.45%)	44.25% (95.88%)
15	11.93% (23.58%)	44.76% (1376.20%)	43.31% (93.83%)
20	11.73% (23.18%)	77.41% (2379.79%)	10.87% (23.55%)
25	5.41% (10.70%)	89.43% (2749.47%)	5.16% (11.17%)

Table 10: Composition of News Classes with Different Thresholds on WESS Scores

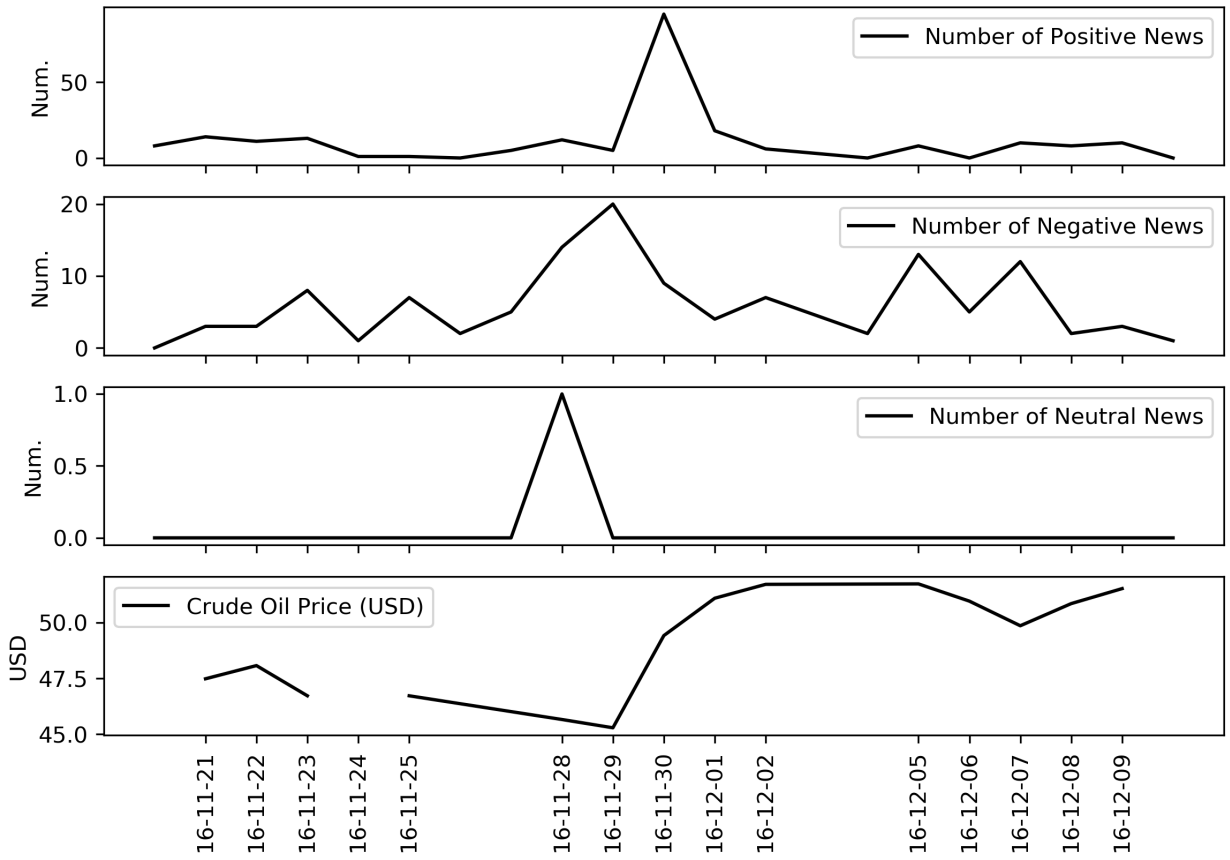
r	Num Negative	Num Neutral	Num Positive
0	46.54% (100.00%)	9.06% (100.00%)	44.40% (100.00%)
0.3	42.85% (92.08%)	13.99% (154.43%)	43.16% (97.19%)
1	39.25% (84.33%)	20.32% (224.32%)	40.43% (91.06%)
3	31.12% (66.87%)	34.62% (382.22%)	34.25% (77.14%)
5	25.24% (54.24%)	46.49% (513.20%)	28.27% (63.66%)
10	17.51% (37.64%)	61.39% (677.67%)	21.10% (47.52%)
15	9.83% (21.13%)	77.68% (857.58%)	12.48% (28.11%)
20	7.98% (17.15%)	84.63% (934.20%)	7.39% (16.65%)
25	4.22% (9.06%)	91.95% (1015.06%)	3.83% (8.63%)

2.6 Case Studies

2.6.1 November 30, 2016: Postive Spike

The first case study investigates the event of an expected production cut by OPEC. On the 30th of November, 2016. Reports concerning this shock were considered as positive news for crude oil price since upcoming negative supply shock generally leads to expectation on soaring prices.

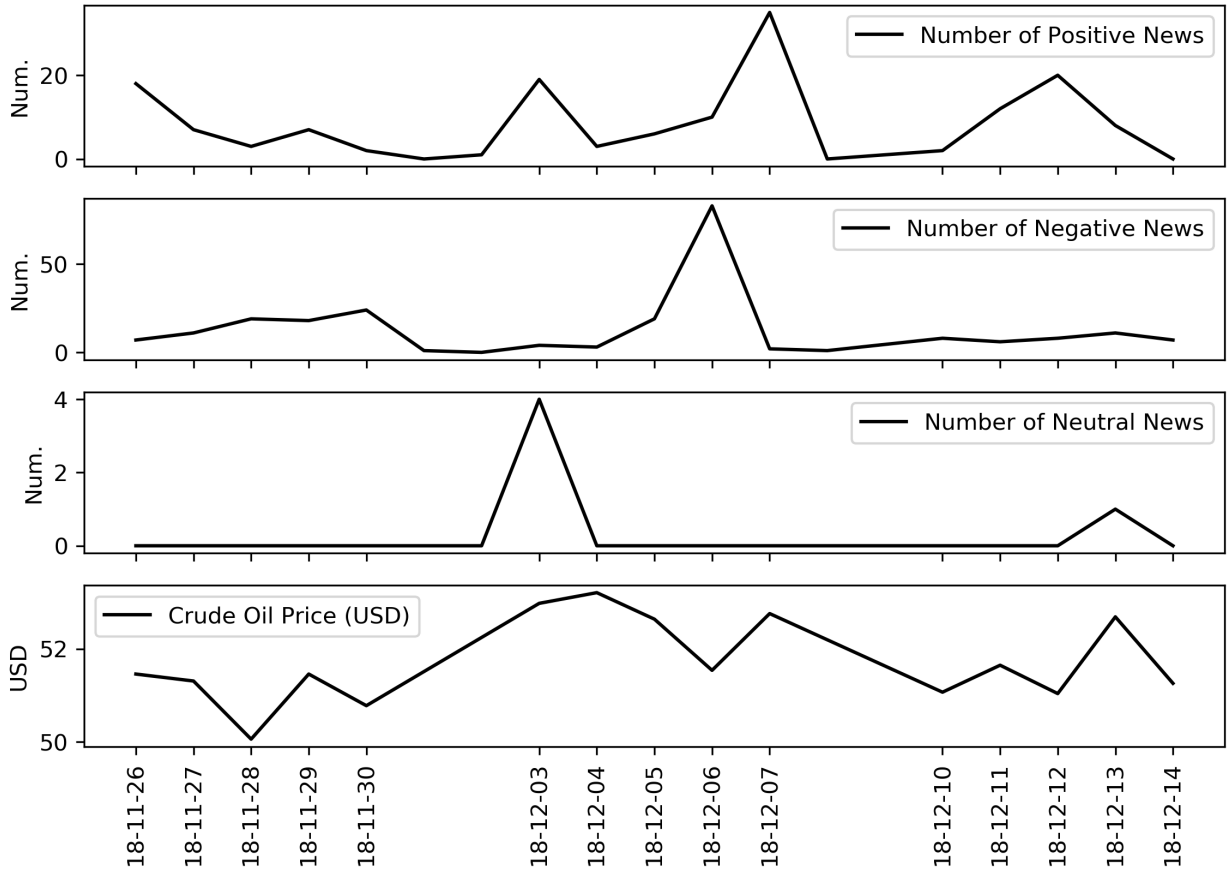
Figure 18: Crude Oil Price and Number of Events within 10 Days



2.6.2 December 6, 2018: Negative Spike

The US had become a net oil-exporting country in the week of Dec. 6, for the first time in 75 years (citation: Bloomberg). This major shift marks a potential negative shock in the demand side of the crude oil market and news reporting this fact was all considered as negative events for the crude oil price.

Figure 19: Crude Oil Price and Number of Events within 10 Days



2.6.3 June 12-13 Positive Spike in Down Period

Unlike in the two previous cases, the third case investigates the impact of positive news spike in a period with falling oil prices.

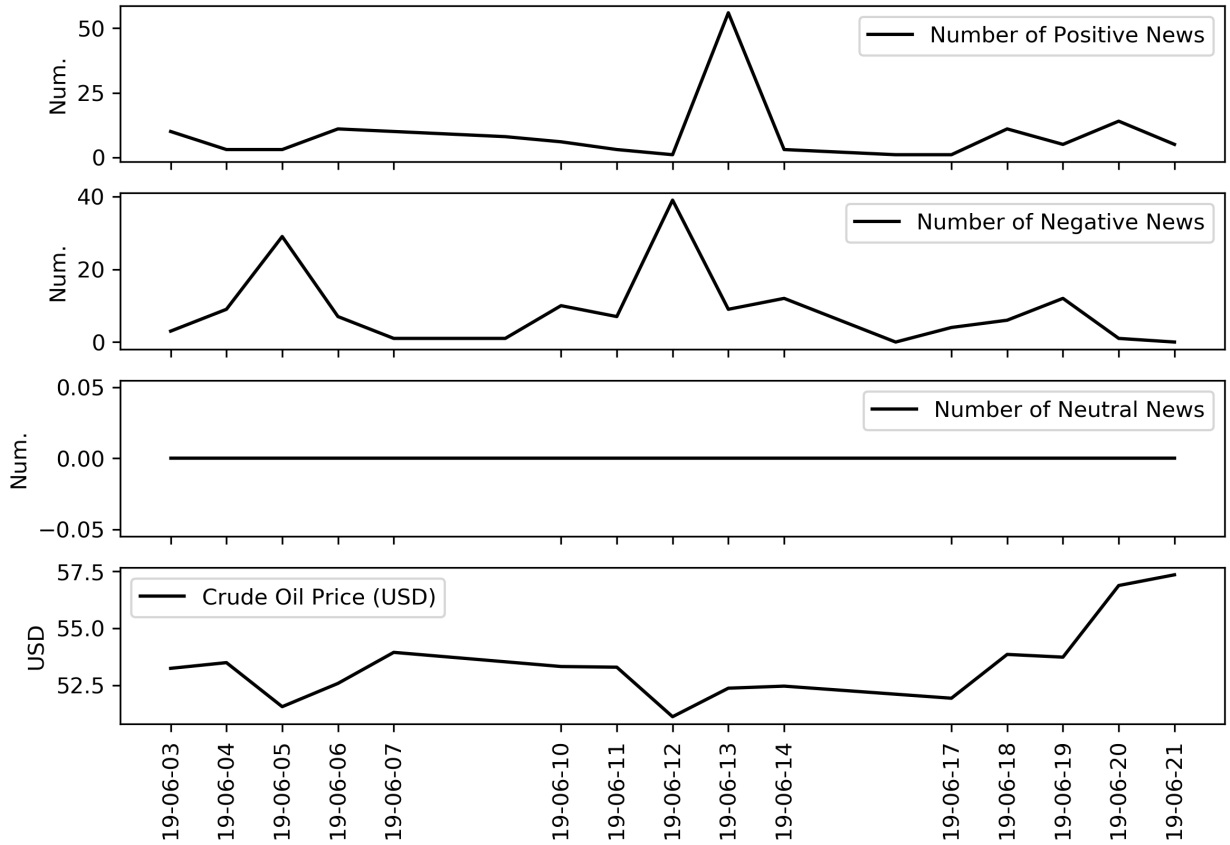
Insert Bloomberg source

The crude oil price had been decreasing since April 2019. The tension between the US and Iran had been accumulating since the US withdrew from the Joint Comprehensive Plan of Action on the 8th of May and alleged Iran for the first Gulf of Oman incident occurred on the 12th. In response, Iran had threatened to close the Strait of Hormuz, which is an important channel for international oil shipping. However, as we can see from the figure 20, before Jun. 12, the major theme of news available was positive, this can be explained by the stable oil supply from Saudi Arabia and other oil-exporting countries.

The story changed on Jun. 13, when the second Gulf of Oman incident happened. Two oil tankers were attacked while passing the Gulf of Oman, which further escalated the tension

between the US and Iran, and the market had sufficient reason to expect a negative supply shock. With the arrival of such a cluster of positive news (positive for crude oil prices), the price increased significantly after the incident but returned to its normal decreasing trend after approximately one week.

Figure 20: Crude Oil Price and Number of Events within 10 Days



3 Model

3.1 Framework

Figure 21 illustrates the framework of our model as a directed acyclic graph (DAG). In a DAG, an arrow $X \to Y$ indicates $p(X, Y) = p(Y|X)p(X)$. Intuitively, one may interpret $X \to Y$ as X is causing Y.

On each day t, the real state of world is denoted as ω_t, which is a high dimensional variable describing everything happening in the world on day t.

The first component in this model consists of a batch of news subscriptions to various news sources. Each of these news sources summarize ω_t as a collection of news articles, which are literally a collection of texts. Some sources provide summary on these articles as well as analysis on the potential economic impacts from events mentioned in these articles. We define the collection of news articles altogether with any summary and analysis from the news provider to be the information flow (conditioned on subscriptions). For example, consider one trader who only read Wall Street Journal in his office everyday, then his perception of the state of world is formed by (therefore, a function of) those news articles and analysis of news on Wall Street Journal. In this case all those articles and analysis is precisely the information flow received by this trader.

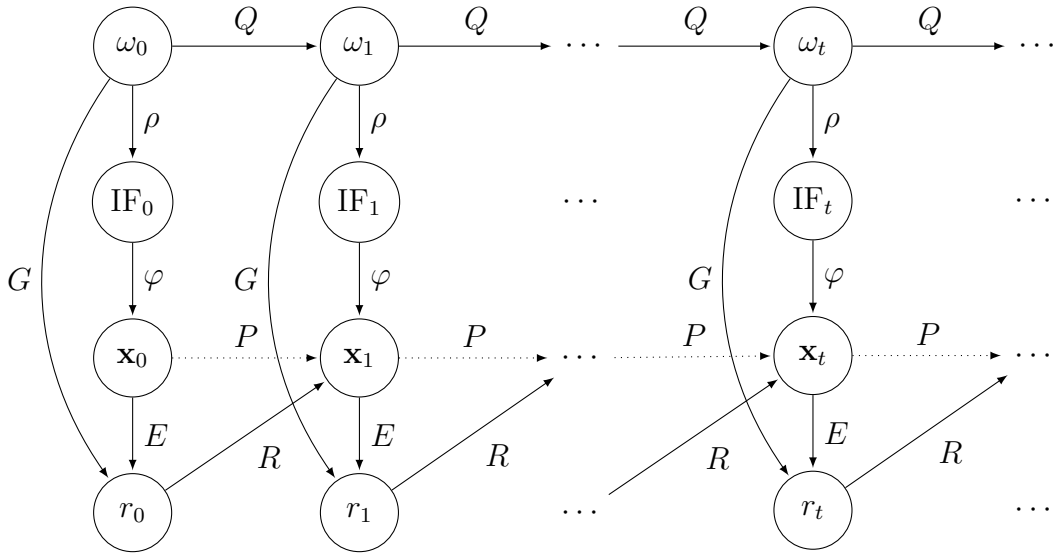
Given an arbitrary time period, say one day, the information flow within this period is simply the collection of news reported within this time period. We denote the information flow on day t as IF_t. Therefore, IF_t provides a summary of ω_t.

The information flow consists of texts and summaries of news articles, one needs to quantify the information flow before applying quantitative models to it. This paper works on a daily basis prediction task, therefore, we would need to quantify the abstract information flow on each day t as a real-valued vector, $\mathbf{x}_t$. Ideally, $\mathbf{x}_t$ should provide a finer summary, especially sentiments, of the IF_t as well as ω_t. Sometime we refer to $\mathbf{x}$ as the quantified information flow and IF as the abstract information flow.

The last component is the realized return r_t. The true state ω_t is determining the actual realized return r_t on day t. Moreover, since traders are often reacting to news reports, so the quantified information flow $\mathbf{x}_t$ is affecting r_t as well.

3.2 Formal Framework

Figure 21: The Framework



3.2.1 Timestamps

In the following discussion, this paper uses non-negative real numbers, $t \in \mathbb{R}_+$, to indicate timestamps accurate to seconds. Specifically, 12 am of first day in dataset (January 3, 2000) corresponds to $t = 0$ and the length of 24 hours is normalized to one. Using this timestamp convention, an integer t indicates the beginning of the t^{th} trading day in the dataset. For example, the time stamp $t = 10$ represents 12:00 am of the 10^{th} trading day in our dataset, which was January 18, 2000. Similarly, $t = 10 + \frac{10}{24}$ denotes 10:00 am of January 18, 2000.

Each news article in the dataset has one timestamp τ corresponding to the time when this piece of news is published. Because this paper works on a daily basis prediction task, we need to discretize the continuous timestamp of t into integers and convert the frequency into daily frequency. clarify

3.2.2 States of World

Figure 21 illustrates the framework of our model as a directed acyclic graph (DAG). In a DAG, an arrow $X \to Y$ indicates $p(X, Y) = p(Y|X)p(X)$. Intuitively, one may interpret $X \to Y$ as X is causing Y.

On each day $t \in \mathbb{Z}_+$, the real state of world is denoted as $\omega_t \in \Omega$, which is a latent variable describing everything happening in the world on day t. The latent of possible states of world Ω is left unspecified because we do not need to interpret ω_t explicitly for this prediction task.

The dynamics of $\{\omega_t\}$ is a stochastic process governed a transition probability Q. The process evolves following equation (3.1):

$$\omega_t \sim Q(\omega_t | \omega_{t-1}, \omega_{t-2}, \cdots, \omega_0) \tag{3.1}$$

3.2.3 Information Flow

The first component in this model consists of a batch of news subscriptions to various news sources. Each of these news sources summarize ω_t as a collection of news articles, which are literally a collection of texts. Some sources provide summary on these articles as well as analysis on the potential economic impacts from events mentioned in these articles. We define the collection of news articles altogether with any summary and analysis from the news provider to be the **information flow** (conditioned on subscriptions), denoted as $\rho(\omega_t)$. The functional form suggests the information flow received by an individual depends on both the state of world ω_t and another subscription function ρ characterizing how many resources this individual has access to. One individual with subscription function ρ_1 has access to more resources than another one with ρ_2 if

$$\rho_2(\omega) \subseteq \rho_1(\omega) \quad \forall \omega \in \Omega \tag{3.2}$$

For example, consider one trader who only read Wall Street Journal in his office everyday, then his perception of the state of world is formed by (therefore, a function of) those news articles and analysis of news on Wall Street Journal. In this case all those articles and analysis is precisely the information flow received by this trader.

Given an arbitrary time period, say one day, the information flow within this period is simply the collection of news reported within this time period. We denote the information flow on day t as IF_t. Therefore, IF_t provides a summary of ω_t.

In this study, our subscription function ρ is defined by the RPNA dataset, which consists

of four sources: Dow Jones Newswires, Wall Street Journal, Barron's, and MarketWatch. So the information flow used in this paper is the collection of news articles and relevant analysis from the above-mentioned four sources.

Formally, let N denote total number of news articles about crude oils in the RPNA dataset, specifically, $N = 106,960$. One may firstly sort all N news based on the timestamp when each news arrived. Then each piece of news in the dataset can be uniquely indexed using an integer $n \in \{1, 2, \cdots, N\}$. For example, news article n is the n^{th} news in the dataset. Let τ_n denote the time when the n^{th} news article was reported.

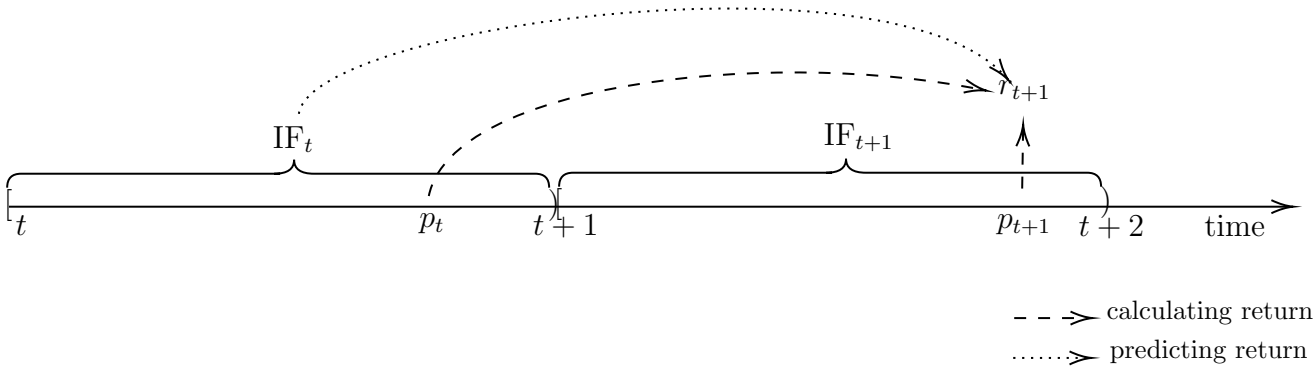
Using this index method, one may express an information flow of day t as a set of integers instead, so that IF_t contains the integer indices of all news arrived on day t. That is,

$$\text{IF}_t = \{n : \tau_n \in [t, t+1)\} \tag{3.3}$$

Figure 22 summarizes the workflow of the prediction task in a minimal setting: predicting r_{t+1} using information on the day t only. Each of t, $t+1$ and $t+2$ denotes the beginning of the day, p_t and p_{t+1} indicate the time when the closing price is computed. Firstly, the return on day $t+1$ is computed from p_{t+1} and p_t using equation (2.2). Then, some predictive models are used to predict r_{t+1} using IF_t.

The workflow suggests the right end of $[t, t+1)$ must be open so that IF_t does not contain any information at time $t+1$. So that while the model is predicting r_{t+1} using IF_t, the model is not peaking into the future.

Figure 22: Workflow of the Prediction Task



3.2.4 Characteristic Function

In this paper, we are approaching the prediction task using auto-regressive models, which means we are using lagged values to predict the future. Let $\mathcal{M}$ denote a predictive model uses information in the past to the future crude oil return.

For instance, if one is building a model conducting one step ahead forecasting based on historical information (both information flow of past days and historical returns) up to ℓ days in the past, $\mathcal{M}$ is a map from information in the past ℓ days to a prediction:

$$\mathcal{M}\left(\begin{bmatrix} \mathrm{IF}_{t-\ell+1} \\ r_{t-\ell+1} \end{bmatrix}, \begin{bmatrix} \mathrm{IF}_{t-\ell+2} \\ r_{t-\ell+2} \end{bmatrix}, \cdots, \begin{bmatrix} \mathrm{IF}_{t} \\ r_{t} \end{bmatrix}\right) = \hat{r}_{t+1} \tag{3.4}$$

Then the model $\mathcal{M}$ is evaluated based on how close r_{t+1} and $\hat{r}_{t+1}$ are. However, each information flow IF_t is a set of (indices of) news articles, these indices remain abstract and do not have any meaning on themselves. Simply feeding these indices to a machine learning model will not generate any meaningful result, one needs to extract some features from those news articles for predictive models. In order to quantify these abstract information flow, we propose a mapping called characteristic function. Let $\mathcal{T}$ be an arbitrary time period, such as the whole day t:

$$\mathcal{T} := [t, t+1) \tag{3.5}$$

We define a characteristic function φ [4] as a mapping from a time period $\mathcal{T}$ to a real-valued vector $\varphi(\mathcal{T}) \in \mathbb{R}^d$, where d denotes the number of features constructed. Ideally, $\varphi(\mathcal{T})$ should provide a quantitative summary from various aspects of news articles in $\mathrm{IF}_{\mathcal{T}}$.

For example, one valid characteristic function φ can construct the following two features: 1) the the number of news articles (events) in period $\mathcal{T}$ and 2) the average event sentiment

[4] In the context of probability, the characteristic function of a distribution fully describes the distribution, refer to (Ushakov 1999) for a review of this topic. Here we define the characteristic function to be the function mapping a collection of news to a vector $\mathbf{x}$ of summary statistic of these news.

score of these news:

$$\varphi(\mathcal{T}) = \begin{bmatrix} |\{n : \tau_n \in \mathcal{T}\}| \\ \frac{1}{|\{n:\tau_n \in \mathcal{T}\}|} \sum_{n \ s.t. \tau_n \in \mathcal{T}} \text{ESS}_n \end{bmatrix} = \begin{bmatrix} \text{Number of News on Day } t \\ \text{Average ESS Score of News on Day } t \end{bmatrix} \in \mathbb{R}^2 \quad (3.6)$$

Note that in subsequent sections, the characteristic function actually used has far more features than the example in equation (3.6).

With a chosen characteristic function φ, one can summarize the information flow on each day t using a real-valued vector $\mathbf{x}_t$:

$$\mathbf{x}_t := \varphi([t, t+1)) \quad (3.7)$$

In this paper, the characteristic function used provides a summary on the sentiment scores provided by RPNA dataset, therefore, we define $\mathbf{x}_t$ to be the **sentiment** of the information flow on day t.

Using characteristic functions, the predictive model in equation (3.4) can be equivalently expressed as

$$\mathcal{M}\left(\begin{bmatrix} \varphi([t-\ell+1, t-\ell+2)) \\ r_{t-\ell+1} \end{bmatrix}, \begin{bmatrix} \varphi([t-\ell+2, t-\ell+3)) \\ r_{t-\ell+2} \end{bmatrix}, \cdots, \begin{bmatrix} \varphi([t, t+1)) \\ r_t \end{bmatrix}\right) \quad (3.8)$$

$$= \mathcal{M}\left(\mathbf{x}_{t-\ell+1}, r_{t-\ell+1}, \mathbf{x}_{t-\ell+2}, r_{t-\ell+2}, \cdots, \mathbf{x}_t, r_t\right) = \hat{r}_{t+1} \quad (3.9)$$

Gathering First or Summarizing First Exchanging the order of applying characteristic function and aggregating information induces subtle difference in the model. Note that we may aggregate information flow first , that is, take $\mathcal{T}' = [t-\ell+1, t+1)$. Then, we can construct a summary of all news arrive in the past ℓ days by applying φ on $\mathcal{T}'$:

$$\mathbf{x}_{t-\ell+1:t} = \varphi(\mathcal{T}') \quad (3.10)$$

Therefore, the following alternative formulation is equally valid,

$$\mathcal{M}\left(\varphi([t-\ell+1, t+1)), r_{t-\ell+1}, r_{r-\ell+2}, \cdots, r_t\right) \tag{3.11}$$

$$= \mathcal{M}\left(\mathbf{x}_{t-\ell+1:t}, r_{t-\ell+1}, r_{r-\ell+2}, \cdots, r_t\right) = \hat{r}_{t+1} \tag{3.12}$$

In the following parts of this paper, we call equation (3.9) the **summarizing-first** formulation since it apply φ on news arrived in each day first, then feeds the collection of summaries to a predictive model. And, we call (3.12) the **gathering-first** formulation, since it firstly collects all news arrive in the time period of consideration, and apply the characteristic function on all these news.

Gathering-first approach provides more news samples to φ to construct sentiment $\mathbf{x}$ of news arrive in the period of consideration. However, no matter how large ℓ is, gathering-first approach only returns one copy of sentiment. In contrast, summarizing-first approach returns ℓ copies of sentiments, and provides algorithms with more features to learn from. Therefore, we are using the summarizing-first approach in most cases to fully leverage the capability of machine learning models.

3.2.5 Inter-temporal Dependency

The state of world is changing from ω_t to ω_{t+1} between two consecutive trading days, and ω_{t+1} depends on ω_t. Moreover, $\mathbf{x}_t$ affects $\mathbf{x}_{t+1}$ as well since certain type of events are reported continuously for more than one day, and the news sentiments exhibits inter-temporal correlation. For example, export restrictions by OPEC countries is a negative news for crude oil prices. An article about this OPEC meeting is reported on the first day and an analysis report of this restriction is published on the second day.

Lastly, some news on day $t+1$ are simply reporting the return on the previous day so that r_t impacts $\mathbf{x}_{t+1}$ as well.

3.3 Empirical Model

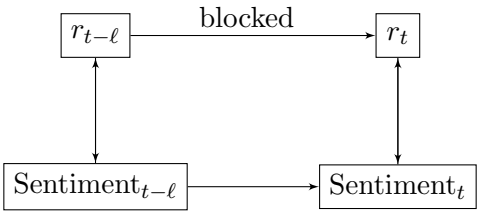
The empirical models to be estimated in this paper are focusing on the dynamics of sentiments $\{\mathbf{x}_t\}_t$ and returns $\{r_t\}_t$.

As examined in the data section, the inter-temporal correlation among returns is weak, hence, predicting current r_t using lagged value of returns is far too challenging for simple models. The framework proposed by this paper aims to use series of sentiments $\{\mathbf{x}_t\}_t$ constructed, which have stronger inter-temporal correlation, to bridge the gap.

Specifically, in figure 21, each pair of $\mathbf{x}_t$ and $\mathbf{x}_{t+1}$ are correlation through the chain consisting of $\mathbf{x}_t$, ω_t, ω_{t+1}, and $\mathbf{x}_{t+1}$. In order to model the dynamic from $\mathbf{x}_t$ to $\mathbf{x}_{t+1}$, we have to construct models estimating P, E, and R in the graph first.

Figure 23 illustrates the idea of forecasting returns via sentiments. Suppose we wish to predict returns on day t, r_t, using only information at time $t-\ell$ (i.e., both $r_{t-\ell}$ and $\mathbf{x}_{t-\ell}$) for some integer $\ell > 1$. The ACF and PACF of return series have only a few significant lags, so that the arrow from $r_{t-\ell}$ to r_t is blocked by this weak correlation. This prevents one from forecasting r_t using $r_{t-\ell}$ with directly a simple model like autoregression integrated moving average (ARIMA). However, the strong inter-temporal correlation in sentiment series allows one to predict Sentiment_t using $\text{Sentiment}_{t-\ell}$. Secondly, the correlation between sentiment and return enables the model to estimate r_t from the prediction of Sentiment_t. The composite of two steps above provides an indirect approach of forecasting r_t using information at time $t-\ell$.

Figure 23: Framework



Literature have been using hidden Markov models (HMMs) for time series forecasting. As mentioned before, $\mathbf{x}_{t+1}$ are determined by the collection of news on day $t+1$ (the information flow) via a characteristic function. However, many of those news in IF_{t+1} are simply reporting past price movements of crude oil, that is, r_t. Therefore, the proposed framework extends the hidden Markov framework by allowing the directed edge from r_t to $\mathbf{x}_{t+1}$, the edge R, to explicitly model the impact of historical price movements on future news sentiment.

Literature here

We model $\{\mathbf{x}_t\}_t$ as a stochastic process whose dynamics is governed by the **transition**

probability, P, and the **reporting probability**, R, plus random noises:

$$\mathbf{x}_t = \mathbf{x}_t^A + \mathbf{x}_t^B + \varepsilon_t \tag{3.13}$$

$$\text{where} \quad \mathbf{x}_t^A \sim P(\mathbf{x}_t^A | \mathbf{x}_{t-1}) \tag{3.14}$$

$$\mathbf{x}_t^B \sim R(\mathbf{x}_t^A | r_{t-1}) \tag{3.15}$$

The transition probability P models the impact of past news sentiments on the future news sentiment, and $\mathbf{x}_t^A$ is the portion of sentiment $\mathbf{x}_t$ solely determined by the inter-temporal correlation among $\mathbf{x}$. In addition, another reporting probability R models the impact of past returns on future news sentiment. Hence, $\mathbf{x}_t^B$ is the part of $\mathbf{x}_t$ responses to past price movement. For example, $\mathbf{x}_t^B$ could be the average of sentiment scores assigned to articles simply reporting the return on the previous day. More generally, two parts of news sentiments can be merged using one joint distribution PR:

$$\mathbf{x}_t \sim PR(\mathbf{x}_t | \mathbf{x}_{t-1}, r_{t-1}) \tag{3.16}$$

Expanding equation (3.16) recursively shows that $\mathbf{x}_t$ is in fact impacted by all historical values of $\mathbf{x}_t$ and r_t. Therefore, the entire history of $\{\mathbf{x}_{t-1}, \mathbf{x}_{t-2}, \cdots, \mathbf{x}_0\}$ and $\{r_{t-1}, r_{t-2}, \cdots, r_0\}$ contribute to the distribution of $\mathbf{x}_t$

$$\mathbf{x}_t \sim F(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{x}_{t-2}, \cdots, \mathbf{x}_0, r_{t-1}, r_{t-2}, \cdots, r_0) \tag{3.17}$$

The **order** of a Markov model determines the length of its memory, a Markov model has order ℓ if the distribution of an arbitrary random variable Y_t only depends on the past ℓ values, that is, for every $t > \ell$,

$$P(Y_t | Y_{t-1}, Y_{t-2}, \cdots, Y_0) = P(Y_t | Y_{t-1}, Y_{t-2}, \cdots, Y_{t-\ell}) \tag{3.18}$$

In most cases, the impact of observations in the distant past, say $\mathbf{x}_{t-1,000}$, on the current observation $\mathbf{x}_t$ is negligible. Therefore, for simplicity, we assume the chain in equation (3.17)

is assumed to have a finite order ℓ. Hence,

$$\mathbf{x}_t \sim F(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{x}_{t-2}, \cdots, \mathbf{x}_{t-\ell}, r_{t-1}, r_{t-2}, \cdots, r_{t-\ell}) \tag{3.19}$$

As mentioned before, the actual return r_t is determined by multiple factors. Firstly, all those events happening on day t, that is, ω_t affects r_t. Moreover, traders are often time response to news published, and traders' behaviours affects the market and therefore r_t. The distribution of r_t follows an **emission probability** distribution E. In particular, E maps the current state of world ω_t and current news sentiment $\mathbf{x}_t$ to a distribution of realized returns r_t. Hence, the distribution of r_t depends on both the true state of world and the news sentiment.

$$r_t \sim E(r|\omega_t, \mathbf{x}_t, r_{t-1}, r_{t-2}, \cdots, r_{t-\ell}) \tag{3.20}$$

For generality, even not shown in figure 21, we assume the distribution of r_t depends on historical returns too. Therefore, the distribution in equation (3.20) includes the lagged values of returns as well. However the impact of adding historical returns should be insignificant given previous analysis on autocorrelations. Moreover, since we assume the Markov chain of returns has order ℓ, so that historical return values before $r - \ell$ are discarded.

Note that the series of $\{\omega_t\}$ is latent, the model has only observations on $\{\mathbf{x}_t\}$ and $\{r_t\}$. For each day t, we can now construct two predictors of return, r_t,

$$\hat{r}_t^{\text{raw}} = \mathbb{E}[r_t|r_{t-1}, r_{t-2}, \cdots, r_{t-\ell}] \tag{3.21}$$

$$= \mathcal{M}^{\text{raw}}(r_{t-1}, r_{t-2}, \cdots, r_{t-\ell}) \tag{3.22}$$

$$\hat{r}_t^{\text{senti}} = \mathbb{E}[r_t|\mathbf{x}_{t-1}, \mathbf{x}_{t-2}, \cdots, \mathbf{x}_{t-\ell}, r_{t-1}, r_{t-2}, \cdots, r_{t-\ell}] \tag{3.23}$$

$$= \mathcal{M}^{\text{senti}}(\mathbf{x}_{t-1}, \mathbf{x}_{t-2}, \cdots, \mathbf{x}_{t-\ell}, r_{t-1}, r_{t-2}, \cdots, r_{t-\ell}) \tag{3.24}$$

The first model $\mathcal{M}^{\text{raw}}$ is predicting the return following a classical auto-regressive manner, that is, using lagged values of return only. In contrast, the second model $\mathcal{M}^{\text{senti}}$ incorporates the sentiment series as well.

Using this framework, we may formulate the original question of interest as whether $\hat{r}_t^{\text{senti}}$ is a significantly better prediction of r_{t+1} than $\hat{r}_t^{\text{raw}}$, in terms of prediction accuracy. Recall the workflow in figure 22, since the closing price p_t is realized near the end of day and the return r_t can be computed at the same time. Based on figure 13, most information within IF_t will arrive before r_t is realized, if the market is efficient, a great portion of IF_t should have already been reflected in r_t. In this case, IF_t, and therefore $\mathbf{x}_t$, will not provide meaningful information to the prediction of r_t. Therefore, if the efficient market hypothesis holds true, the performance of two above-mentioned predictions should be similar. And we can conclude that adding the news sentiment series does not help predict returns. In contrast, if $\hat{r}_t^{\text{senti}}$ outperformed $\hat{r}_t^{\text{raw}}$ a lot, then we can claim that the news sentiment series is capable of improving return predictions.

In experiment section of this paper, this paper uses different types of statistical learning models to estimate $\hat{r}_t^{\text{raw}}$ and $\hat{r}_t^{\text{senti}}$, then compare their performances.

4 Experiments

4.1 Procedures

The empirical model in section 3.3 gives a brief description on how to assess the predictive power of models with and without sentiment dataset. We need to choose a specific characteristic function φ and predictive model $\mathcal{M}$ in order to generate quantitive metrics and compare performances.

4.1.1 Feature Constructions

The previous section described the rough idea of using a characteristic function to quantify an information flow. Before implementing any statistical model, the first step is to choose a specific characteristic function which extracts quantitive summary statistics from a given information flow. In order to maximize the number of features extracted, the proposed characteristic function utilizes features from both the gathering first and summarizing first paradigms.

Let $\ell \in \mathbb{Z}_+$ denote the length of model's memory. That is, while predicting r_t, the model can only use information from day $t-\ell$ to day $t-1$. This paper chooses $\ell = 31$ so that the model uses information within a whole month to predict one return.

Firstly, for each day from day $t-\ell$ to day $t-1$, the characteristic function computes all variables in table 11. Even though `NUM_EVENTS` can be deduced from `ESS_MEAN` and `ESS_TOTAL`, we decided to include it as a proxy of the volatility of news networks.

Table 11: Daily Summary Statistics from Summarizing-First Paradigm (1)

Code Name	Variable	Code Name	Variable
`ESS_MEAN`	Average ESS	`WESS_MEAN`	Average WESS
`ESS_TOTAL`	Sum of ESS	`WESS_TOTAL`	Sum of WESS
`NUM_EVENTS`	Number of Events		

Moreover, following the methodology in section 2.5, the characteristic function classifies positive (negative / neutral) news using their ESS (WESS) scores and a predefined threshold r. The characteristic function added the number of news in each class to the daily summary.

Table 12: Daily Summary Statistics from Summarizing-First Paradigm (2)

Code Name	Variable	Code Name	Variable
`NUM_POSITIVE_ESS`	# news s.t. ESS $> r$	`NUM_POSITIVE_WESS`	# news s.t. WESS $> r$
`NUM_NEGATIVE_ESS`	# news s.t. ESS $< -r$	`NUM_NEGATIVE_WESS`	# news s.t. WESS $< -r$
`NUM_NEUTRAL_ESS`	# news s.t. ESS $\in [-r, r]$	`NUM_NEUTRAL_WESS`	# news s.t. WESS $\in [-r, r]$

Table 11 and table 12 together provide the daily summary for the sentiment dataset from day $t-\ell$ to day $t-1$, and concatenating them gives 11ℓ features in total. We denote the characteristic function computing daily summary as φ_{daily}, for a given information flow on day t, $\varphi_{\text{daily}}(\text{IF}_t)$ calculates 11 summary statistics of IF_t.

Studies on the gold future market suggests that negative news sentiments tend to invoke greater responses from the market (Smales 2014). It is likely for this observation to be true in crude oil market as well since gold market and crude oil market share many similar features. One way to separate impacts from positive and negative news is to split all news into a positive and a negative group (neutral news are dropped). Then, applying φ_{daily}

on these two subsets of information flow gives two copies of summaries, one for positive news and one for negative news. However, distinguishing positive and negative news while constructing daily summary doubles the number of features generated (from 11ℓ to 22ℓ), and can potentially lead to the curse of dimensionality especially when ℓ is large (Friedman 1997). Therefore, for the daily summary, the proposed characteristic function only counts the number of news in each class but does not calculate detailed summary statistics (e.g., standard deviation and percentiles).

In contrast, while processing the aggregated information flow in the period of consideration, $\text{IF}_{[t-\ell,t)}$ (i.e., all news from day $t-\ell$ to day $t-1$), instead of creating ℓ copies of daily summaries for ℓ days, the number of features constructed no longer depends on ℓ. This allows us to choose more complicated characteristic function for the aggregate information flow. Therefore, we may choose a characteristic function distinguishing positive and negative news and computes more detailed summary statistics.

Let $\varphi_{\text{aggregate}}$ denote the second characteristic function extracting features from $\text{IF}_{[t-\ell,t)}$. Table 13 enumerates 8 types of summary statistics used. Firstly, $\varphi_{\text{aggregate}}$ computes the the 8 statistics in table 13 for ESS and WESS of all news in $\text{IF}_{[t-\ell,t)}$ (16 features in total).

Table 13: Summary Statistics from Gathering-First Paradigm (1)

Code Name	Variable
`x_count`	Number of Samples X
`x_mean`	Average of X
`x_std`	Standard Deviation
`x_min`	Minimum
`x_25%`	25^{th} Percentile
`x_50%`	Median
`x_75%`	75^{th} Percentile
`x_max`	Maximum

To emphasize extreme events more, we then compute the 8 summary statistics in table 13 for the squared ESS and WESS scores as well (16 features in total). Note that ESS and WESS scores range from -50 to 50, the squared scores are defined following equation (4.1)

so that their signs are preserved.

$$\begin{aligned} \text{ESS}^2 &:= \text{sign}(\text{ESS}) \times \text{ESS}^2 \\ \text{WESS}^2 &:= \text{sign}(\text{WESS}) \times \text{WESS}^2 \end{aligned} \tag{4.1}$$

Afterwards, we split $\text{IF}_{[t-\ell,t)}$ into the positive group, $\text{IF}^{+}_{[t-\ell,t)}$, and the negative group, $\text{IF}^{-}_{[t-\ell,t)}$, according to the sign of each news' ESS score (news with zero ESS score are discarded). Note that by the definition of WESS, signs of ESS and WESS are always the same, hence, splitting $\text{IF}_{[t-\ell,t)}$ based on ESS and WESS always gives the same outcome. Then, $\varphi_{\text{aggregate}}$ summarizes the number of news, the average ESS and the average WESS for each of $\text{IF}^{+}_{[t-\ell,t)}$ and $\text{IF}^{-}_{[t-\ell,t)}$ (6 features in total).

Lastly, as we noticed in the data exploration, there are two clusters of ESS scores (at -15 and 18). Moreover, news article assigned with these two values of ESS scores are in general reports of past price movements and carry little information about future price changes. To address this issue, we define $\text{IF}^{++}_{[t-\ell,t)}$ (extremely positive news) to be the subset of $\text{IF}_{[t-\ell,t)}$ with ESS strictly greater than 18, and $\text{IF}^{--}_{[t-\ell,t)}$ (extremely negative news) to be these news with ESS strictly less than -15. Afterwards, $\varphi_{\text{aggregate}}$ summarizes the number of extremely positive (negative), the average ESS and the average WESS of news in $\text{IF}^{++}_{[t-\ell,t)}$ and $\text{IF}^{--}_{[t-\ell,t)}$ (6 features in total).

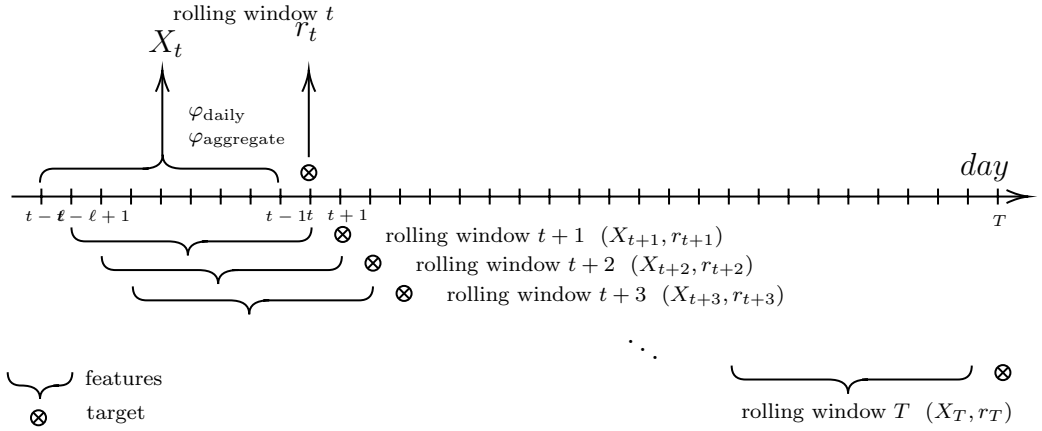
Overall, $\varphi_{\text{aggregate}}$ constructs 44 features to summarize each information flow $\text{IF}_{[t-\ell,t)}$. Altogether with the 11ℓ features from φ_{daily}, $11\ell + 44$ features are used to predict the return r_t. For example, there would be 385 features if one chose ℓ to be 31. Given there are around 5,000 daily returns to train the model, using approximate 400 independent variable is reasonable.

4.1.2 Rolling-Window Method

The second step of the pipeline is to construct a batch of feature-target pairs (called a sample), (X_t, r_t), so that we can evaluate model $\mathcal{M}$ based on how close $\mathcal{M}(X_t)$ and r_t are. Let $\ell = 31$ for now, returns in the first month are discarded from the dataset since we do not have sufficient number of days to construct these features required. Afterwards, a

rolling-window method generates a training set from the series of returns and news dataset as illustrated in figure 24.

Figure 24: Using Rolling Window to Construct (X_t, r_t) pairs



For each day t with valid return r_t (those days with missing returns are discarded), the set of features X_t consists of 31 daily summary from φ_{daily}, one aggregate summary from $\varphi_{\text{aggregate}}$ and 31 lagged values of returns.

$$X_t^{\ell=31} := \left\{\varphi_{\text{daily}}(\text{IF}_{t-31}), \cdots, \varphi_{\text{daily}}(\text{IF}_{t-1}), \varphi_{\text{aggregate}}(\text{IF}_{[t-31,t)}), r_{t-31}, \cdots, r_{t-1}\right\} \tag{4.2}$$

Therefore, $X_t^{\ell=31}$ consists of 416 real-valued features used to predict r_t. Afterwards, the rolling window constructor move to $t+1$ (if available, otherwise move to the next day with valid returns) and generate another pair of feature and target (X_{t+1}, r_{t+1}).

Finally, the rolling window generates 4,934 pairs of (X_t, r_t), in which t ranges from January 1, 2000 to September 30, 2019. Among the 4,934 samples, each X_t is a 416 dimensional real-valued vector and r_t is a real-valued scalar. This paper uses samples (X_t, r_t) with t before January 1, 2019 as training set (4,747 samples) and the rest of samples are taken as test set (187 samples).

$$\mathcal{D}^{\text{train}} := \{(X_t, r_t) : t \leq \text{December 31, 2018}\} \tag{4.3}$$

$$\mathcal{D}^{\text{test}} := \{(X_t, r_t) : t \geq \text{January 1, 2019}\} \tag{4.4}$$

After assessing models' performances on the test set, $\mathcal{D}^{\text{train}}$ and $\mathcal{D}^{\text{test}}$ are further split into 5

subsets [5] to explore the effectiveness of models on each day of the week.

4.1.3 Performance Metrics

In order to quantify the performance of a predictive model, we have to specify a performance metric measuring the proximity between predictions and actual values. Let $\hat{r}_t$ denote the predicted value of r_t, the performance metric should reflect the proximity between predicted and true returns. The primary performance metric used in this paper is the mean squared error (MSE). The MSE of a model aiming to predict $\{r_1, r_2, \cdots, r_T\}$ is defined as

$$MSE := \frac{1}{T}\sum_{t=1}^{T}(r_t - \hat{r}_t)^2 \tag{4.5}$$

One advantage of MSE metric is that it is differentiable with respect to each $\hat{r}_t$, this differentiability allows us to train models on this dataset using back-propagation algorithm (Hecht-Nielsen 1989). Even though not all predictive models in this paper are based on back-propagation or require differentiable objective functions, we use MSE to as the primary metric to select and evaluate models for consistency.

Unfortunately, the MSE is not naturally interpretable, and MSE changes when the unit of returns switches to percentage returns. We introduce anther two widely used error metrics, directional accuracy (DA) and mean absolute percentage error (MAPE).

$$DA := \frac{1}{T}\sum_{t=1}^{T}\mathbb{1}\{\text{sign}(r_t) = \text{sign}(\hat{r}_t)\} \tag{4.6}$$

$$MAPE := \frac{1}{T}\sum_{t=1}^{T}\left|\frac{r_t - \hat{r}_t}{r_t}\right| \tag{4.7}$$

The directional accuracy measures the frequency that the model predict the sign of return correctly. Both DA and MAPE are easily interpretable, but none of them is differentiable.

[5] (X_t, r_t) are split based on which day of the week t is. There are only 5 groups since r_t are always missing when t is weekend.

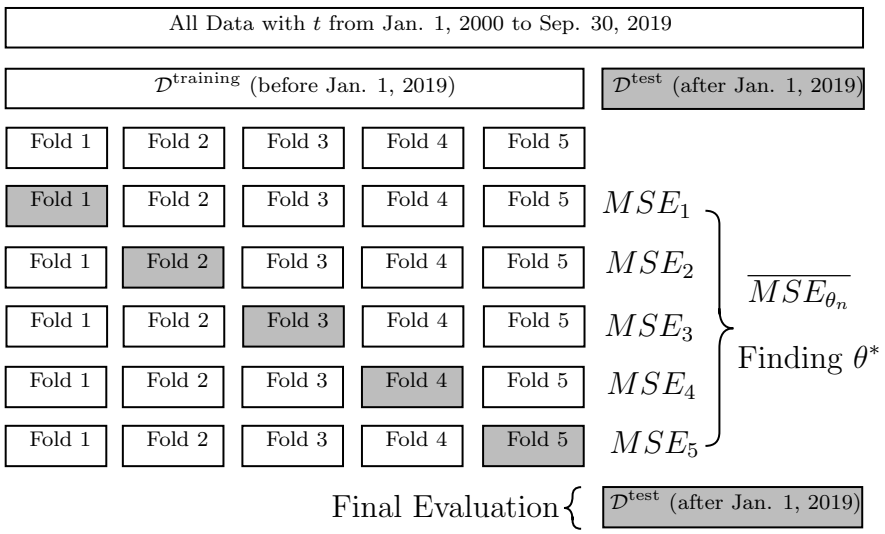
4.1.4 Model Selection and Randomized Cross Validation

After choosing a class of predictive models, one still needs to select a set of hyper-parameters (i.e., model configurations), $\theta \in \Theta$. In subsequent discussions, we use subscript $\mathcal{M}_\theta$ to denote a model with configuration θ from class $\mathcal{M}$. For example, suppose $\mathcal{M}$ is the class of all polynomial regressions, then one hyper-parameter is the maximum degree in the regression equation. In this case, the set of possible hyper-parameters, Θ, is all positive integers. One has to choose the optimal maximum degree θ^* from Θ so that $\mathcal{M}_\theta^*$ has the best (test-time) performance.

Choosing the optimal θ^* is crucial for building effective predictive algorithms, simply choosing a super complicated model would over-fit the training set and leads to poor test-time performance (Claeskens and Hjort 2008).

For each predictor class $\mathcal{M}$ and corresponding space of hyper-parameters Θ, we firstly determine the optimal hyper-parameter $\theta^* \in \Theta$ using a 5-fold randomized cross validation algorithm (5-fold RCV).

Figure 25: 5-fold Randomized Cross Validation



Firstly, N candidates of hyper-parameters $\{\theta_1, \theta_2, \cdots, \theta_N\}$ are sampled from a uniform distribution on Θ (i.e., the randomized part in RCV). Then, for each $\theta_n \in \{\theta_1, \theta_2, \cdots, \theta_N\}$, θ_n defines a predictive model $\mathcal{M}_{\theta_n}$ Figure 25 illustrates the cross validation procedure for one hyper-parameter set θ_n. The entire $\mathcal{D}^{\text{train}}$ are split into 5 equal consecutive subsets (called

folds): $\mathcal{D}_i^{\text{train}}$ for $i \in \{1,2,3,4,5\}$. Then, $\mathcal{M}_{\theta_n}$ is fitted on $\cup_{i\in\{1,2,3,4\}}\mathcal{D}_i^{\text{train}}$ (training set) and evaluated on $\mathcal{D}_5^{\text{train}}$ (validation set), let $\widehat{MSE}_5$ denote mean squared error of this model on $\mathcal{D}_5^{\text{train}}$. Afterwards, the same model is fitted again on $\cup_{i\in\{1,2,3,5\}}\mathcal{D}_i^{\text{train}}$, evaluated on $\mathcal{D}_4^{\text{train}}$ and leads to another error metric $\widehat{MSE}_4$. The same procedure can be repeated for five times with different validation set and creates five mean squared error metrics. Let $\overline{MSE}_{\theta_n}$ denote the average of $\widehat{MSE}_1$ to $\widehat{MSE}_5$ using hyper-parameter θ_n. Then, $\overline{MSE}_{\theta_n}$ constitutes an estimated test-time performance of model $\mathcal{M}_{\theta_n}$ on $\mathcal{D}^{\text{test}}$ when it is fitted on $\mathcal{D}^{\text{train}}$. Among the N candidates of hyper-parameters for model class $\mathcal{M}$, we choose θ_n with the smallest $\overline{MSE}_{\theta_n}$ to be the best-performing parameter, denoted as θ^*. This θ^* is not necessarily the truly best one among all θ in Θ, θ^* is only the best-performing configuration within N samples. However, since we sampled the N candidates uniformly from Θ, performance of the selected θ^* should be close to the truly optimal configuration especially when N is sufficiently large.

While using 5-fold RCV and N sampled θ, we need for fit $5N$ models from class $\mathcal{M}$ in total to determine the optimal configuration θ^* for this model class. Clearly, the larger N is the more likely for us to include the truly optimal configuration in our sampled configurations. Specifically, we choose N to be 500 in this paper.

4.2 Linear Models

4.2.1 Baseline Models: The Moving Average Predictor

In this section, we examine the predictive power of auto-regressive time series models. To better evaluate model performances, we firstly define several dummy models for benchmarking purpose. The easiest model is a naive predictor, $\mathcal{M}_{\text{naive}}$, predicting zero returns all the time. In addition, other dummy predictors, $\mathcal{M}_{\text{MA}(k)}$, are always predicting the average return of the last k trading days.

As mentioned before, this paper uses all data before December 31, 2019 as the training set and the rest as the test set. Table 14 shows those dummy models' performances in terms of mean-squared-error and directional accuracy.

Table 14: Performances of Benchmark Models

Model	Training MSE	Training DA	Testing MSE	Test DA
$\mathcal{M}_{\text{naive}}$	4.655	0.716%	4.057	0.538%
$\mathcal{M}_{\text{MA(5)}}$	5.612	50.274%	4.693	50.000%
$\mathcal{M}_{\text{MA(25)}}$	4.811	50.295%	4.248	50.000%
$\mathcal{M}_{\text{MA(50)}}$	4.725	49.536%	4.261	50.000%
$\mathcal{M}_{\text{MA(100)}}$	4.706	49.241%	4.226	44.624%
$\mathcal{M}_{\text{MA(300)}}$	4.676	47.977%	4.060	48.925%

Because returns are rarely zero in the dataset, the directional accuracy of $\mathcal{M}_{\text{naive}}$ model is nearly zero on both training and testing sets. As we expected, those benchmark models never achieve better performances than random guessing (50% accuracy). These benchmark results suggest that the crude oil market is efficient (i.e., unpredictable) with respect to historical returns and moving-average-based models. Then we are going to examine whether other more sophisticated models can attain significantly better test-time performances than these models in table 14.

4.2.2 Autoregressive Integrated Moving Average (ARIMA)

This paper searches over 72 configurations of ARIMA models, $\mathcal{M}_{\text{ARIMA}(p,d,q)}$, with (p, d, q) specified in table 15. This paper uses the Akaike's information criterion (AIC) of ARIMA models on the training set to determine the optimal configuration. Model with configurations $(5, 0, 5)$, $(4, 0, 3)$ and $(5, 0, 4)$ attain the three lowest AIC on training set. Table 16 summarizes the performances of the three optimal configurations identified.

Table 15: Scope of Hyper-parameters for ARIMA

Hyper-parameter	Scope
p	{0,1,2,3,4,5}
d	{0,1,2}
q	{0,1,2,3,4,5}

Table 16: Performances of Linear Models

Model	Testing MSE	Test DA
$\mathcal{M}_{\text{ARIMA(5,0,5)}}$	4.074	50.763 %
$\mathcal{M}_{\text{ARIMA(4,0,3)}}$	4.070	51.156 %
$\mathcal{M}_{\text{ARIMA(5,0,4)}}$	4.073	50.567 %

It turns out that ARIMA models perform poorly and merely out-performed benchmark models specified before. Therefore, we conclude that Since ARIMA models work univariate time series, we can only test the market's efficiency on the information set without news sentiment.

4.2.3 Vector Autoregressions (VAR)

This paper uses VAR models to incorporate the news sentiment dataset.

4.3 Support Vector Regressions (SVR)

Support vector machines (SVM) was firstly proposed by Boser, Guyon and Vapnik as a classification method used on hand-written digit recognition (1992). Over the past three decades, SVM has been believed to be the best off-the-shelf algorithm.

The proposed characteristic functions generate 416 predictors in total for each one target r_t, so that the prediction task is in fact a high dimensional problem. Parametric methods such as linear regressions tend to over-fit the training set and perform poorly on the test set.

During the training time, SVM models only focus a few training samples termed *support vectors*, this allows SVM to be applied on high dimensional features without over-fitting.

Moreover, by using different *kernels*, SVMs are capable of transforming these raw features to an even higher dimensional space. For example, if one wishes to classify points in $\mathbb{R}^2$, instead of mapping each (x_1, x_2) to a class, a SVM with polynomial kernel of degree two will classify these points based on $(x_1, x_2, x_1^2, x_1x_2, x_2^2) \in \mathbb{R}^5$ instead. In this case, the original 2-dimensional input space is transformed into a 5-dimensional feature space by the kernel function. While a SVM is using the Radial Basis Function (RBF) kernel, the original input space is transformed into an infinite-dimensional feature space. This implicit feature engi-

neering enables SVM to explore more complex patterns in the dataset. Smola and Scholkorf provide a detailed review of training SVMs and theories behind kernel functions in their work (2004).

A few years after the SVM was proposed as a classifier, Drucker and others proposed an extension to original SVM called support vector regression machines (SVR) (Drucker et al. 1997) SVR performs well on high dimensional regression problems and engineers features implicitly using a kernel function as SVMs.

As mentioned in Smola and Scholkorf's work, performances of support vector machines are determined by several hyper-parameters listed in table 23. To choose the optimal configuration of SVR in this paper's prediction task, a RCV algorithm samples 500 configurations from the scope of hyper-parameters in table 23 and evaluates each configuration based on their MSE and DA on the validation set.

Table 17: Scope of Hyper-parameters for Support Vector Regression Machines

Hyper-parameter	Scope
Kernel Type	{Radial Basis Function (RBF) kernel}
γ	$\{10^{-10}, 10^{-9}, 10^{-8}, 10^{-7}, 10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 0.1, 1, 10\}$
Tolerance	$\{10^{-10}, 10^{-9}, 10^{-8}, 10^{-7}, 10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 0.1, 1, 10\}$
ε	$\{10^{-10}, 10^{-9}, 10^{-8}, 10^{-7}, 10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 0.1, 1, 10\}$
C	$\{10^{-10}, 10^{-9}, 10^{-8}, 10^{-7}, 10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 0.1, 1, 10\}$

Let $\mathcal{M}^{\text{criterion}}_{\text{model class}}$ denote model (configuration) selected using the specified criterion. For instance, $\mathcal{M}^{\text{MSE}}_{\text{SVR}}$ is the support vector regressor minimizes MSE on the validation set. Using two criterions, MSE and DA, there would be two optimal models selected from each model class. Then, the test-time performances of $\mathcal{M}^{\text{MSE}}_{\text{SVR}}$ and $\mathcal{M}^{\text{DA}}_{\text{SVR}}$ will be used to proxy the overall performance of SVR on the crude oil return prediction task.

Table 18 and table 19 presents the optimal model configurations under both criterions and their respective performances.

Table 18: Optimal Hyper-parameters for Support Vector Regression Machines

Model	Kernel	γ	Tolerance	ε	C
$\mathcal{M}^{\text{MSE}}_{\text{SVR}}$	RBF	10^{-6}	1	10^{-3}	10^{-9}
$\mathcal{M}^{\text{DA}}_{\text{SVR}}$	RBF	10^{-7}	10^{-5}	10^{-7}	10

Table 19: Performances of Support Vector Regression Machines

Model	Validation MSE	Validation DA	Testing MSE	Test DA
$\mathcal{M}_{\text{SVR}}^{\text{MSE}}$	4.655	51.433 %		%
$\mathcal{M}_{\text{SVR}}^{\text{DA}}$	4.830	51.665 %		%

4.4 Random Forests

Let p denote the number of independent variables for prediction, in our case, $p = 416$.

Table 20: Scope of Hyper-parameters for Random Forests

Hyper-parameter	Scope
n Number of Trees	$\{1, 2, 3, \cdots, 200\}$
f Max num. of features for each tree	$\{p, \log_2(p)\}$
d Max depth of each tree	$\{10, 14, 19, 24, \cdots, 100, 105, 110, \infty\}$
m_1 Min amount of samples required to split an internal node	$\{2, 5, 10\}$
m_2 Minimum number of samples required at each leaf node	$\{1, 2, 4\}$
What dataset is used to construct each tree	{bootstrapped samples, entire training dataset}

Table 21: Optimal Hyper-parameters for Random Forests

Model	n	f	d	m_1	m_2	Training Samples
$\mathcal{M}_{\text{RF}}^{\text{MSE}}$	96	$\log_2(p)$	10	10	2	Bootstrapped Samples
$\mathcal{M}_{\text{RF}}^{\text{DA}}$	41	p	14	5	4	Entire Training Set

Table 22: Performances of Random Forests

Model	Validation MSE	Validation DA	Testing MSE	Test DA
$\mathcal{M}_{\text{RF}}^{\text{MSE}}$	4.675	50.464 %		%
$\mathcal{M}_{\text{RF}}^{\text{DA}}$	5.716	51.960 %		%

4.5 Recurrent Neural Networks with Long-Short-Term-Memory Cells

Table 23: Scope of Hyper-parameters for Support Vector Regression Machines

Hyper-parameter	Scope
h Size of RNN hidden layer	$\{32, 64, 128, 256, 512, 1024\}$
ℓ Number of RNN hidden layers	$\{1, 2, 3\}$
p_{rnn} Dropout probability in RNN hidden layers	$\{0, 0.25, 0.5\}$
p_{fc} Dropout probability in the output layer	$\{0, 0.25, 0.5\}$
Epochs of training	$\{5, 6, 7, 8, \cdots, 18, 19, 20, 25, 30, 35, \cdots, 200\}$
B Batch size	$\{32, 128, 512\}$
α Learning rate	$\{10^{-5}, 3 \times 10^{-5}, 10^{-4}, 3 \times 10^{-4}, 10^{-3}, 3 \times 10^{-3}, 0.01, 0.03, 0.1, 0.3\}$

Table 24: Optimal Hyper-parameters for LSTM RNNs

Model	h	ℓ	p_{rnn}	p_{fc}	Epochs	B	α
$\mathcal{M}_{\text{LSTM}}^{\text{MSE}}$							
$\mathcal{M}_{\text{LSTM}}^{\text{DA}}$							

Table 25: Performances of LSTM RNNs

Model	Validation MSE	Validation DA	Testing MSE	Test DA
$\mathcal{M}_{\text{LSTM}}^{\text{MSE}}$		%		%
$\mathcal{M}_{\text{LSTM}}^{\text{DA}}$		%		%

4.6 Taking the Day-of-the-Week Effect into Consideration

In section 2.3, we have shown that the crude oil return experiences significant day-of-the-week effect. In particular, Mondays are more likely to experience negative returns compared with other days. Therefore, it is reasonable to conjecture that the underlying dynamics of returns on Mondays might be different from the dynamics of returns of other days. To exploit this fact, we select and train two separate models for returns on Mondays and returns on the rest of the week. The same procedure used before identifies the optimal configuration of

each model class for restricted datasets: dataset with Mondays only and dataset excluding all Mondays.

Table 26: Optimal Hyper-parameters for Random Forests on Restricted Datasets

Model	n	f	d	m_1	m_2	Training Samples
$\mathcal{M}^{\text{MSE}}_{\text{RF, Mondays}}$	115	$\log_2(p)$	38	10	4	Bootstrapped Samples
$\mathcal{M}^{\text{DA}}_{\text{RF, Mondays}}$	116	$\log_2(p)$	38	2	1	Entire Training Set
$\mathcal{M}^{\text{MSE}}_{\text{RF, Other Days}}$	88	$\log_2(p)$	10	5	4	Bootstrapped Samples
$\mathcal{M}^{\text{DA}}_{\text{RF, Other Days}}$	157	p	10	2	1	Entire Training Set

Table 27: Optimal Hyper-parameters for Support Vector Regressions on Restricted Datasets

Model	Kernel	γ	Tolerance	ε	C
$\mathcal{M}^{\text{MSE}}_{\text{SVR, Mondays}}$	RBF	10^{-10}	10^{-3}	10^{-4}	10
$\mathcal{M}^{\text{DA}}_{\text{SVR, Mondays}}$	RBF	10^{-6}	0.1	10^{-6}	1
$\mathcal{M}^{\text{MSE}}_{\text{SVR, Other Days}}$	RBF	0.1	0.1	10^{-4}	10
$\mathcal{M}^{\text{DA}}_{\text{SVR, Other Days}}$	RBF	10^{-9}	0.1	10^{-7}	1

Table 28: Performances of Models on Restricted Datasets

Model	Validation MSE	Validation DA	Testing MSE	Test DA
$\mathcal{M}^{\text{MSE}}_{\text{SVR, Mondays}}$	0.659	52.984 %		%
$\mathcal{M}^{\text{DA}}_{\text{SVR, Mondays}}$	0.681	54.887 %		%
$\mathcal{M}^{\text{MSE}}_{\text{RF, Mondays}}$	0.655	55.574 %		%
$\mathcal{M}^{\text{DA}}_{\text{RF, Mondays}}$	0.672	56.461 %		%
$\mathcal{M}^{\text{MSE}}_{\text{SVR, Other Days}}$	5.575	52.605 %		%
$\mathcal{M}^{\text{DA}}_{\text{SVR, Other Days}}$	5.583	52.631 %		%
$\mathcal{M}^{\text{MSE}}_{\text{RF, Other Days}}$	5.606	50.997 %		%
$\mathcal{M}^{\text{DA}}_{\text{RF, Other Days}}$	6.844	52.449 %		%

References

Baumeister, Christiane, and Lutz Kilian. 2016. "Forty Years of Oil Price Fluctuations: Why the Price of Oil May Still Surprise Us". *Journal of Economic Perspectives* 30 (1): 139–160. ISSN: 0895-3309. doi:10.1257/jep.30.1.139.

Boser, Bernhard E, Isabelle M Guyon, and Vladimir N Vapnik. 1992. "A training algorithm for optimal margin classifiers". *Proceedings of the fifth annual workshop on Computational learning theory*: 144–152. doi:10.1145/130385.130401.

Claeskens, Gerda, and Nils Lid Hjort. 2008. *Model Selection and Model Averaging*. Cambridge Books. Cambridge University Press. ISBN: 9780521852258. https://ideas.repec.org/b/cup/cbooks/9780521852258.html.

Drucker et al. 1997. "Support vector regression machines". *Advances in neural information processing systems*: 155–161.

Friedman, Jerome H. 1997. "On Bias, Variance, 0/1—Loss, and the Curse-of-Dimensionality". *Data Mining and Knowledge Discovery* 1 (1): 55–77. ISSN: 1384-5810. doi:10.1023/a:1009778005914.

Gibbons, Michael R, and Patrick Hess. 1981. "Day of the Week Effects and Asset Returns". *The Journal of Business* 54 (4): 579. ISSN: 0021-9398. doi:10.1086/296147.

Hecht-Nielsen. 1989. "Theory of the backpropagation neural network". *International 1989 Joint Conference on Neural Networks*: 593–605 vol.1. doi:10.1109/ijcnn.1989.118638.

Hodges, J L. 1958. "The significance probability of the smirnov two-sample test". *Arkiv för Matematik* 3 (5): 469–486. ISSN: 0004-2080. doi:10.1007/bf02589501.

Smales, Lee A. 2014. "News sentiment in the gold futures market". *Journal of Banking & Finance* 49:275–286. ISSN: 0378-4266. doi:10.1016/j.jbankfin.2014.09.006.

Smirnov, Nikolai. 1939. "On the estimation of the discrepancy between empirical curves of distribution for two independent samples". *Bulletin Moscow University* 2:3–16.

Smola, Alex J., and Bernhard Schölkopf. 2004. "A tutorial on support vector regression". *Statistics and Computing* 14 (3): 199–222. ISSN: 0960-3174. doi:10.1023/b:stco.0000035301.49549.88.

Ushakov, Nikolai G. 1999. *Selected Topics in Characteristic Functions*. ISBN: 9783110935981.

5 Appendix

Figure 26:

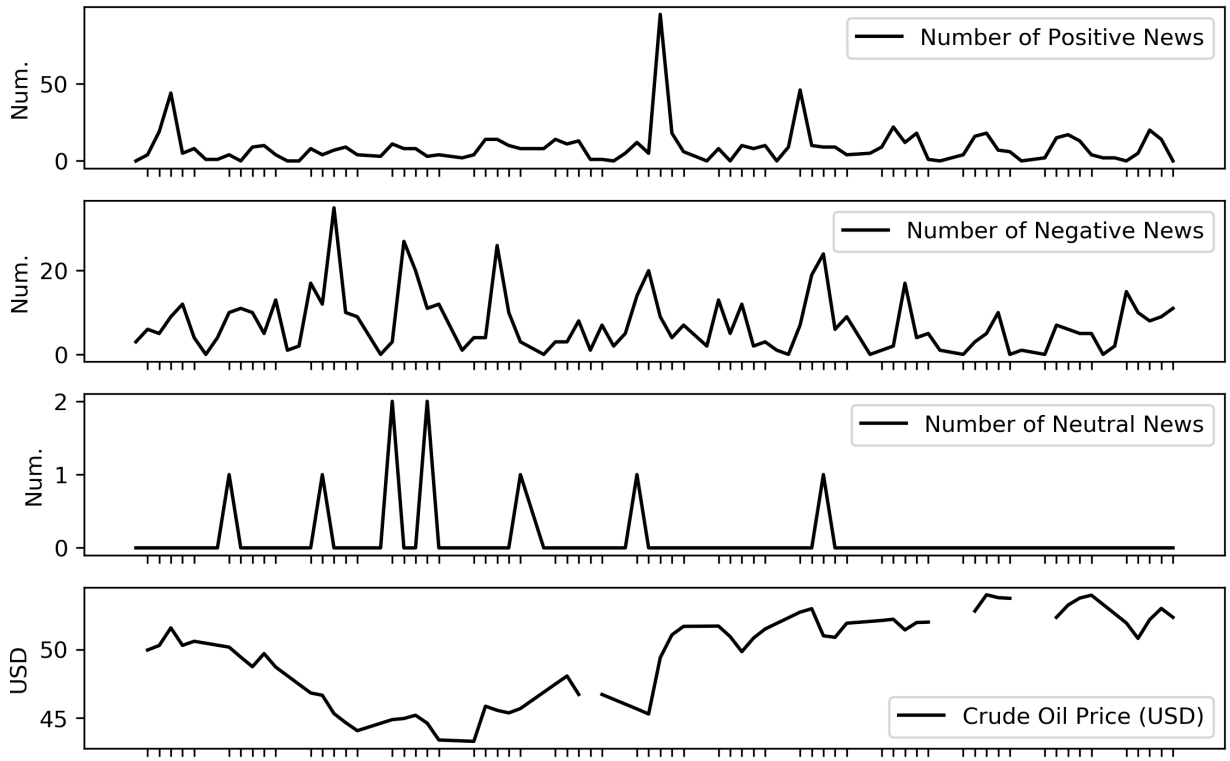


Figure 27:

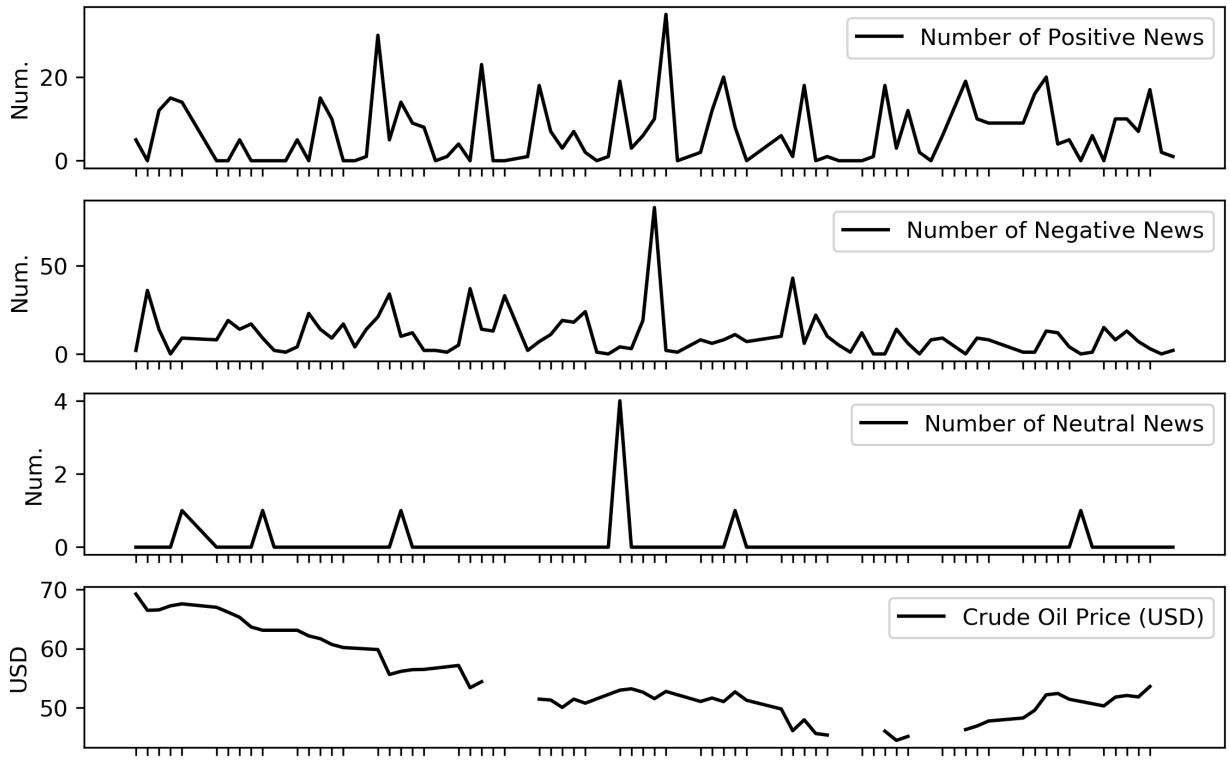

Number of Positive News
Num.
20
0
Number of Negative News
Num.
50
0
Number of Neutral News
Num.
4
2
0
Crude Oil Price (USD)
USD
70
60
50

Figure 28:

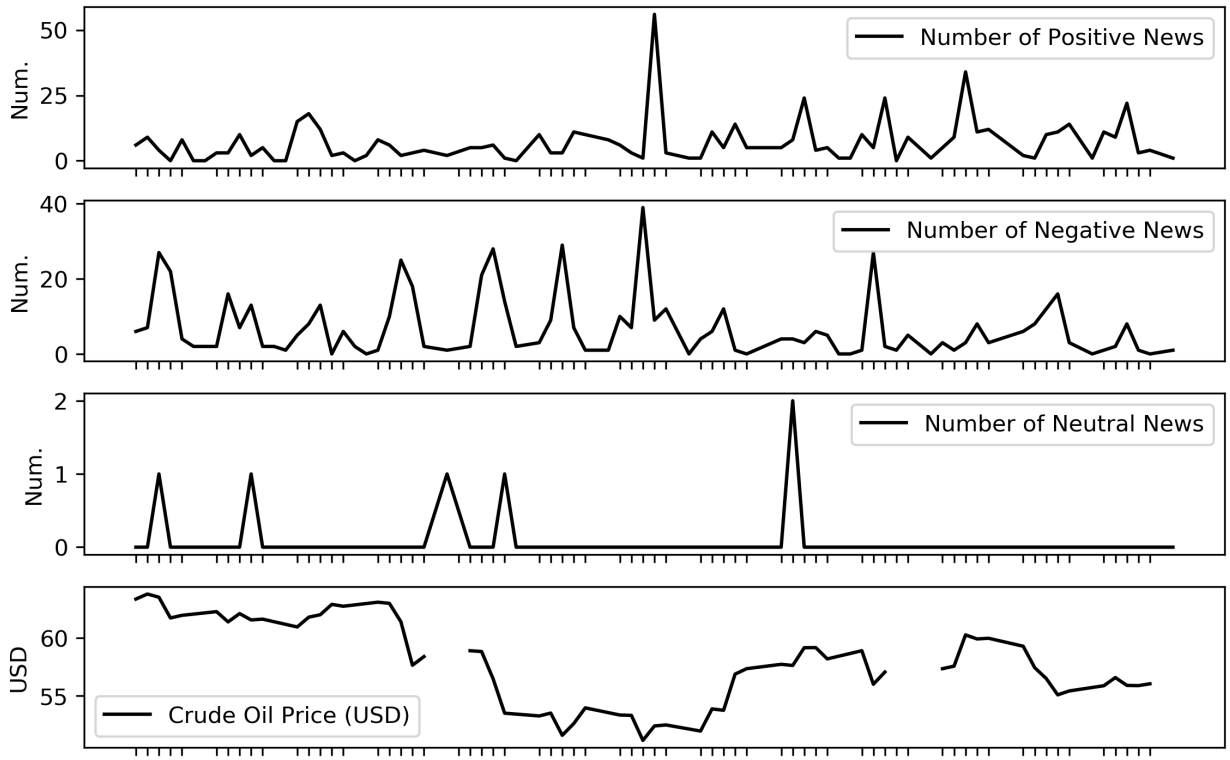

Num.
50
25
0
Number of Positive News
Num.
40
20
0
Number of Negative News
Num.
2
1
0
Number of Neutral News
USD
60
55
Crude Oil Price (USD)

Table 29: All Categories of Positive News

Category	Number of Positive news
commodity-price-gain	22,893
commodity-futures-gain	11,648
supply-decrease-commodity	5,845
imports-up	2,705
commodity-buy-target	1,171
demand-increase-commodity	1,070
exports-down	1,020
spill-commodity	787
commodity-offer-target	429
demand-guidance-increase-commodity	375
price-target-upgrade	332
exports-guidance-down	217
technical-view-bullish	193
supply-guidance-decrease-commodity	186
imports-guidance-up	122
relative-strength-index-oversold	85
embargo	80
piracy	57
pipeline-bombing-attack	32
force-majeure-commodity	26
tanker-accident-commodity	17
export-tax-guidance-decrease	11
pipeline-accident-commodity	11
platform-accident-commodity	11
import-tax-guidance-decrease	8
drilling-suspended-commodity	8
facility-close-output	6
import-tax-decrease	6
hijacking-target-commodity	4
export-tax-decrease	3
market-guidance-up-commodity	2
refinery-accident-commodity	2
facility-accident-commodity	2
technical-price-level-support-bullish	1
pipeline-bombing-threat	1

Table 30: All Categories of Negative News

Category	Number of negative news
commodity-price-loss	26,475
commodity-futures-loss	12,818
supply-increase-commodity	6,629
imports-down	2,017
exports-up	1,308
resource-discovery-commodity	1,179
technical-view-bearish	1,172
demand-decrease-commodity	650
demand-guidance-decrease-commodity	341
commodity-sell-target	303
supply-guidance-increase-commodity	268
price-target-downgrade	261
exports-guidance-up	208
technical-price-level-resistance-bearish	150
force-majeure-lifted-commodity	85
imports-guidance-down	75
export-tax-increase	29
drilling-commodity	27
export-tax-guidance-increase	24
facility-upgrade-output	21
import-tax-increase	18
relative-strength-index-overbought	16
embargo-lifted	12
import-tax-guidance-increase	9
facility-open-output	5
facility-accident-contained-commodity	4
import-tax	3
export-tax	3
facility-sale-output	3
hijacking-released-commodity	1
tax-break-ended	1

Table 31: Filtering using Event Sentiment Score

r	Num Negative	Num Neutral	Num Positive
0	50.59% (100.00%)	3.25% (100.00%)	46.15% (100.00%)
1	50.57% (99.96%)	3.29% (101.24%)	46.13% (99.96%)
2	50.55% (99.91%)	3.33% (102.33%)	46.12% (99.93%)
3	50.52% (99.85%)	3.39% (104.08%)	46.09% (99.87%)
4	50.51% (99.83%)	3.82% (117.33%)	45.68% (98.96%)
5	50.20% (99.23%)	5.24% (161.14%)	44.55% (96.54%)
6	50.04% (98.91%)	5.42% (166.77%)	44.54% (96.49%)
7	50.01% (98.85%)	5.47% (168.07%)	44.52% (96.46%)
8	49.99% (98.81%)	5.51% (169.27%)	44.50% (96.42%)
9	48.88% (96.62%)	6.82% (209.80%)	44.29% (95.97%)
10	48.84% (96.53%)	6.91% (212.45%)	44.25% (95.88%)
11	48.82% (96.49%)	6.97% (214.20%)	44.21% (95.80%)
12	48.78% (96.41%)	7.86% (241.51%)	43.37% (93.96%)
13	48.74% (96.33%)	7.92% (243.55%)	43.34% (93.90%)
14	48.72% (96.29%)	7.96% (244.64%)	43.33% (93.87%)
15	11.93% (23.58%)	44.76% (1376.20%)	43.31% (93.83%)
16	11.88% (23.49%)	44.83% (1378.41%)	43.28% (93.78%)
17	11.85% (23.42%)	44.89% (1379.97%)	43.27% (93.74%)
18	11.82% (23.37%)	77.24% (2374.59%)	10.94% (23.71%)
19	11.80% (23.33%)	77.27% (2375.51%)	10.93% (23.68%)
20	11.73% (23.18%)	77.41% (2379.79%)	10.87% (23.55%)
21	11.42% (22.57%)	77.82% (2392.47%)	10.76% (23.32%)
22	5.69% (11.25%)	83.65% (2571.83%)	10.66% (23.09%)
23	5.57% (11.00%)	83.86% (2578.38%)	10.57% (22.90%)
24	5.53% (10.94%)	89.23% (2743.37%)	5.24% (11.34%)
25	5.41% (10.70%)	89.43% (2749.47%)	5.16% (11.17%)
26	5.37% (10.62%)	89.52% (2752.20%)	5.11% (11.07%)
27	5.32% (10.51%)	89.65% (2756.25%)	5.03% (10.91%)
28	4.23% (8.37%)	91.79% (2822.05%)	3.98% (8.62%)
29	4.21% (8.33%)	91.86% (2824.12%)	3.93% (8.51%)
30	4.18% (8.27%)	91.90% (2825.38%)	3.92% (8.49%)

Table 32: Filtering using Weighted Event Sentiment Score

r	Num Negative	Num Neutral	Num Positive
1	46.54% (100.00%)	9.06% (100.00%)	44.40% (100.00%)
1	39.25% (84.33%)	20.32% (224.32%)	40.43% (91.06%)
2	33.69% (72.40%)	29.93% (330.42%)	36.37% (81.92%)
3	31.12% (66.87%)	34.62% (382.22%)	34.25% (77.14%)
4	28.35% (60.92%)	40.08% (442.46%)	31.57% (71.10%)
5	25.24% (54.24%)	46.49% (513.20%)	28.27% (63.66%)
6	24.92% (53.54%)	49.89% (550.79%)	25.19% (56.73%)
7	21.78% (46.79%)	53.19% (587.19%)	25.03% (56.38%)
8	21.42% (46.04%)	57.06% (629.93%)	21.51% (48.45%)
9	18.09% (38.87%)	60.76% (670.80%)	21.15% (47.62%)
10	17.51% (37.64%)	61.39% (677.67%)	21.10% (47.52%)
11	17.46% (37.53%)	65.26% (720.44%)	17.28% (38.91%)
12	14.07% (30.23%)	69.17% (763.62%)	16.76% (37.75%)
13	13.25% (28.47%)	70.03% (773.05%)	16.72% (37.66%)
14	13.15% (28.25%)	74.32% (820.49%)	12.53% (28.22%)
15	9.83% (21.13%)	77.68% (857.58%)	12.48% (28.11%)
16	9.66% (20.76%)	77.96% (860.57%)	12.38% (27.88%)
17	8.29% (17.82%)	79.37% (876.19%)	12.34% (27.79%)
18	8.18% (17.57%)	84.22% (929.75%)	7.60% (17.12%)
19	8.06% (17.31%)	84.49% (932.75%)	7.45% (16.78%)
20	7.98% (17.15%)	84.63% (934.20%)	7.39% (16.65%)
21	7.51% (16.14%)	85.37% (942.46%)	7.12% (16.03%)
22	4.77% (10.24%)	88.20% (973.67%)	7.03% (15.84%)
23	4.66% (10.01%)	88.42% (976.11%)	6.92% (15.58%)
24	4.48% (9.63%)	91.35% (1008.46%)	4.17% (9.39%)
25	4.22% (9.06%)	91.95% (1015.06%)	3.83% (8.63%)
26	4.16% (8.95%)	92.06% (1016.33%)	3.77% (8.50%)
27	4.09% (8.79%)	92.24% (1018.25%)	3.67% (8.27%)
28	3.28% (7.04%)	93.86% (1036.13%)	2.86% (6.45%)
29	2.95% (6.34%)	94.23% (1040.22%)	2.82% (6.35%)
30	2.92% (6.27%)	94.31% (1041.09%)	2.77% (6.25%)