Forecasting Crude Oil Returns using News Sentiment and Machine Learning *∗*

Tianyu Du *†*

Wednesday 15th April, 2020

# Contents

1. [Introduction](#_bookmark1) 3
2. [Literature Review](#_bookmark2) 3
3. [Data](#_bookmark3) 6
   1. [The West Texas Intermediate (WTI) Crude Oil Dataset](#_bookmark4) . . . . . . . . . . . . 6

[3.2 Crude Oil Returns](#_bookmark5) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 6

* 1. [Day of the Week Effect in Crude Oil Dataset](#_bookmark14) 12
     1. [Difference in Returns across the Week](#_bookmark15) 12
     2. [Kolmogorov-Smirnov test for Distributional Similarities](#_bookmark18) 14
  2. [News Sentiment Dataset](#_bookmark20) 15
     1. [Event Sentiment Scores](#_bookmark24) 18
     2. [Weighted Event Sentiment Scores](#_bookmark27) 20
     3. [News Arrival Time](#_bookmark32) 24
  3. [Classifying News Type](#_bookmark37) 27
  4. [Case Studies](#_bookmark42) 29

*∗*Compile Date: 22:41 Wednesday 15th April, 2020

*†*[tianyu.du@mail.utoronto.ca](mailto:tianyu.du@mail.utoronto.ca)

* + 1. [November 30, 2016: Postive Spike](#_bookmark43) 29
    2. [December 6, 2018: Negative Spike](#_bookmark44) 30
    3. [June 12-13 Positive Spike in Down Period](#_bookmark45) 31

### [Model](#_bookmark47) 32

* 1. [Framework](#_bookmark48) 32
  2. [Formal Framework](#_bookmark49) 34
     1. [Timestamps](#_bookmark51) 34
     2. [States of World](#_bookmark52) 34
     3. [Information Flow](#_bookmark54) 35
     4. [Characteristic Function](#_bookmark56) 37
     5. [Inter-temporal Dependency](#_bookmark61) 39
  3. [Empirical Model](#_bookmark62) 39

### [Experiments](#_bookmark67) 43

* 1. [Procedures](#_bookmark68) 43
     1. [Feature Constructions](#_bookmark69) 43
     2. [Rolling-Window Method](#_bookmark74) 46
     3. [Performance Metrics](#_bookmark76) 48
     4. [Model Selection and Randomized Cross Validation](#_bookmark77) 49
  2. [Baseline Models: The Naive Predictor and Moving Average Predictors](#_bookmark79) 51
  3. [Linear Models: Autoregressive Integrated Moving Average](#_bookmark81) 52
  4. [Support Vector Regression](#_bookmark84) 54
  5. [Random Forests](#_bookmark88) 56
  6. [Long Short-Term Memory Recurrent Neural Networks](#_bookmark92) 58
  7. [Taking the Day-of-the-Week Effect into Consideration](#_bookmark96) 60

### [Conclusions](#_bookmark101) 62

### [Appendix](#_bookmark122) 65

# Introduction

This is the introduction.

# Literature Review

Traditional time series methods explore endogenous patterns encoded in the series of returns and use lagged values of returns to forecast future returns. Mohammadi and Su examine the performance of autoregression integrated moving average-generalized autore- gressive conditional heteroskedasticity (ARIMA-GARCH) on eleven international crude oil market. The authors apply ARIMA-GARCH models to forecast the value and volatility of weekly crude oil returns in those markets and conclude that the return is characterized by a MA(1) process (Mohammadi and Su [2010).](#_bookmark116)

In addition to using historical returns as features, many are utilizing other technical in- dicators of the market as well. These methods aim to transform lagged values of return into meaningful predictors such as moving averages (MA) and moving average divergence con- vergence (MADC). Baker and Wurgler construct an investor sentiment index for the stock market using a collection of technical indicators: the closed-end fund discount, NYSE share turnover, numbers of IPOs, average first-day returns, and share of equity issues in total eq- uity and debt issues, dividend premium (BAKER and WURGLER [2006](#_bookmark102)). Moreover, a more recent study constructs market sentiment indices for both WTI and Brent oil future mar- kets, and these constructed indices have shown significant predictive power while controlling external variables such as stock indices and exchange rates (Deeney et al. [2015).](#_bookmark108)

More recently, besides exploring the predictive power of market indices, scholars are pay- ing more attention to alternative data sources such as news. The inceptions of most market partitioners are in fact shaped by what news they have heard, and these partitioners trade commodities, stocks, and other financial derivatives based on news they receive. In By- bee and others’ recent work, they analyze the full-text contents of over 800,000 articles on the Wall Street Journal over the past 30 years. They have demonstrated that text-based features from news articles can track economic activities accurately. Moreover, these text-

based features have additional predictive powers to traditional macroeconomic indicators for macroeconomic forecasting (Bybee et al. [2019).](#_bookmark106) Most works exploring and utilizing the predictive power of news sentiments are focusing on the stock market. Tetlock analyzes the interaction between articles in the “Abreast of the Market” column in the Wall Street Journal and the stock market. Using vector autoregression (VAR), Tetlock models the intertemporal correlation between the stock market and a measure of media pessimism constructed using principal component analysis (PCA). The author finds pessimistic signals in news media can a precursor of downward pressure on stock price and high trading volume (Tetlock [2007).](#_bookmark120) Mudinas, Zhang, and Levene demonstrate that news sentiments extracted from Financial Times and tweets Granger cause prices of several stocks in S&P500. Experiments in their paper suggest the prediction accuracies of support vector machines and recurrent neural net- works are improved by utilizing additional news sentiment features (2019). Hu and others designed a Hybrid Attention Network (HAN) to extract information and forecast price move- ments. Beyond accurately predicting trends in the stock market, trading algorithms based on the proposed predictive model demonstrate superior annualized return in the Chinese stock market compared with other algorithms (Hu et al. [2018).](#_bookmark113)

Instead of the stock market, Roache and Rossi analyze the impacts of macroeconomic announcements on daily prices of 12 commodity futures including oil, heating oil and nat- ural gas between 1997 and 2009. Their experiments show that commodities are in general insensitive to macroeconomic news and models based on news perform poorly on forecasting daily prices of commodities (Roache and Rossi [2010).](#_bookmark118) Brandt and Gao examine the poten- tially different impacts on crude oil markets from news about macroeconomic fundamentals and geopolitical events. News about geopolitical events shows strong short-run impacts on the crude oil market. In the long run, news about macroeconomic fundamentals acts as a significant predictor of crude oil returns. In contrast, news about geopolitical events only induces uncertainty and higher trading volume (Brandt and Gao [2019).](#_bookmark105)

The sentiment of one news article is highly subjective, assigning sentiment scores to ar- ticles manually will inevitably lead to biased sentiment scores. Instead of asking an expert in finance to evaluate the article after reading it, each article’s sentiment score should be completely based on a pre-defined scoring rule which is (mostly) independent of each individ-

ual article so that the score is as objective as possible. Modern natural language processing (NLP) techniques allow researchers to construct sentiment indices for a large volume of news articles without actually reading all texts. One simplest method of construct objective news sentiment is the dictionary-based method: researchers firstly build a dictionary mapping frequently observed words in financial news articles to their general sentiment. Loughran and McDonald propose a dictionary classifying words into six categories: negative, positive, uncertainty, litigious, modal and constraining (Loughran and McDonald [2011](#_bookmark115), Bodnaruk, Loughran, and McDonald [2015,](#_bookmark104) Loughran and McDonald [2016).](#_bookmark114) For instance, the word ‘bankrupt’ is classified as a word carrying negative sentiment in the Loughran-McDonald sentiment dictionary. Then, the algorithm divides the article into single words (the tok- enization step) and reduce each word to their lemma (the lemmatization step). For example, the word ‘bankruptcy’ would be replaced by ‘bankrupt’. Afterward, the algorithm counts occurrences of words in the article and calculates the sentiment score based on frequencies of words belonging to each category (e.g., positive and negative words). One potential draw- back of the dictionary-based method is that it can cover frequently used words only: the Loughran-McDonald covers around 86,000 vocabularies. Another issue is that the dictionary is domain-specific, the dictionary built for the stock market may not be optimal for the crude oil market. Models designed for one market cannot be easily transferred to another market without rebuilding the dictionary. Another method of constructing sentiment is based on word embedding techniques in NLP. The embedding algorithm maps each word to a high- dimensional vector, termed embedding vector, in the embedding space so that words with close meanings would have close embedding vectors. Pennington, Socher, and Manning in- troduce a Global Vectors for Word Representation (GloVe) algorithm to embed 2.2 millions of commonly used words to 300-dimensional embedding vectors (Pennington, Socher, and Manning [2014).](#_bookmark117) Embedding techniques cover a much wider range of vocabularies compared with dictionary-based methods so that models can be migrated easily.

# 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.

## 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 bench- mark 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.](#_bookmark103) 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.

## 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](#_bookmark6) 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

July 4

January 1

December 25

July 3

November 23

November 24

November 25

November 22

November 26

Number of Days with Missing Data

20

20

19

10

10

10

10

9

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 ∆ denotes the gap between date *t* and the previous trading day, so that *t −* ∆ 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 *rt*. So that the following relationship

quantifies the relationship between *pt−*∆ and *pt*:

*pt* = *ert*∆*pt−*∆ (3.1)

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,

∆:

*r* = ln(*pt*) *−* ln(*pt−*∆)

*t* ∆

(3.2)

Equivalently, *rt* measures the daily return over the ∆ day period. Because returns are closed to zero in most time, in order to avoid decimal issues, this paper converts all *rt*’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, ∆ = 3. If the previous Friday was a holiday without valid price data, *rt* will be ln(*p*Mon) *−* ln(*p*Prev Thu), and ∆ = 4.

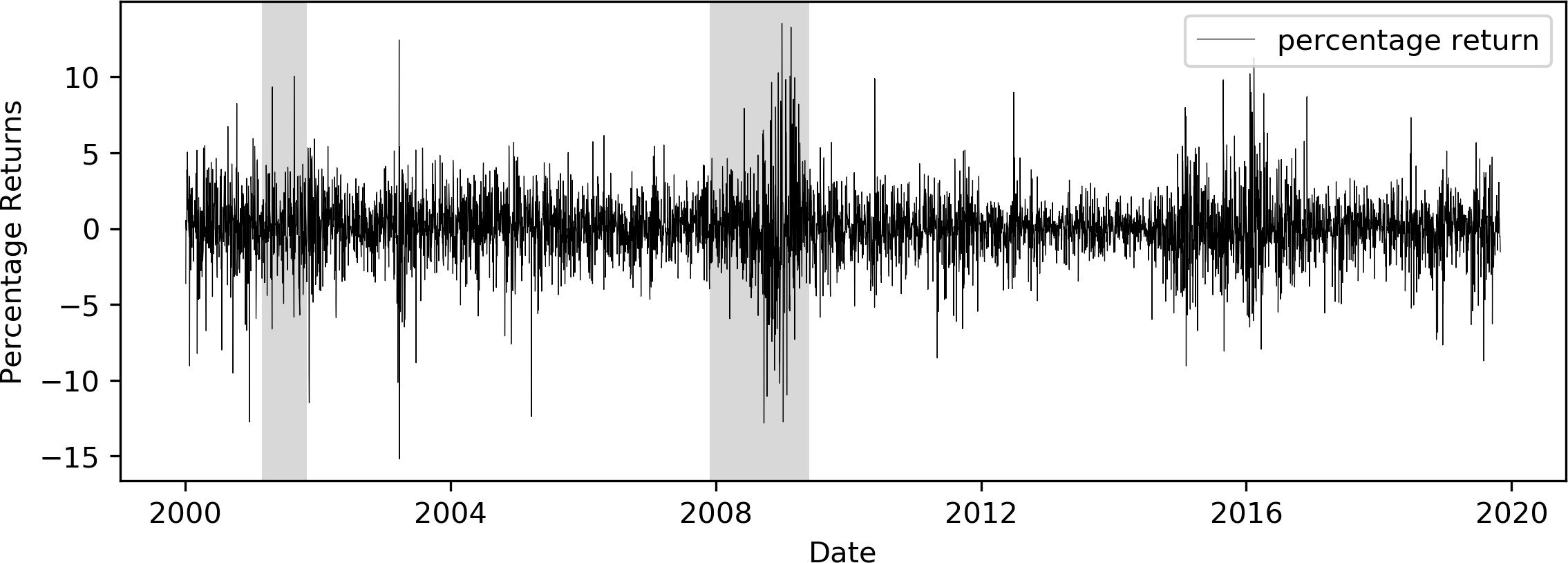
Table 2: Distribution of ∆ by Weekdays

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Day of the week | Num. Days. | Num. Trading Days | ∆=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](#_bookmark8) summaries the distribution of ∆ values. The ∆ 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 ∆ 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](#_bookmark9) 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

Figure 1: Crude Oil Returns

Add more

descrip- tions

on the chronol-

ogy.

Table [3](#_bookmark10) provides summary statistics for the percentage crude oil returns, in which nor- malized skewness and excess kurtosis are defined as

1 *n*

*i*=1

3

*x*

*m*ˆ3 :=

*n*

(*xi − x*¯)3

*σ*

1 *n*

(normalized skewness) (3.3)

*i*=1

4

*x*

*−* 3 (excess kurtosis) (3.4)

*m*ˆ4 :=

*n*

(*xi − x*¯)4

*σ*

1I 1 L*n*

where *σx* :=

*n*

(*xi − x*)2 (3.5)

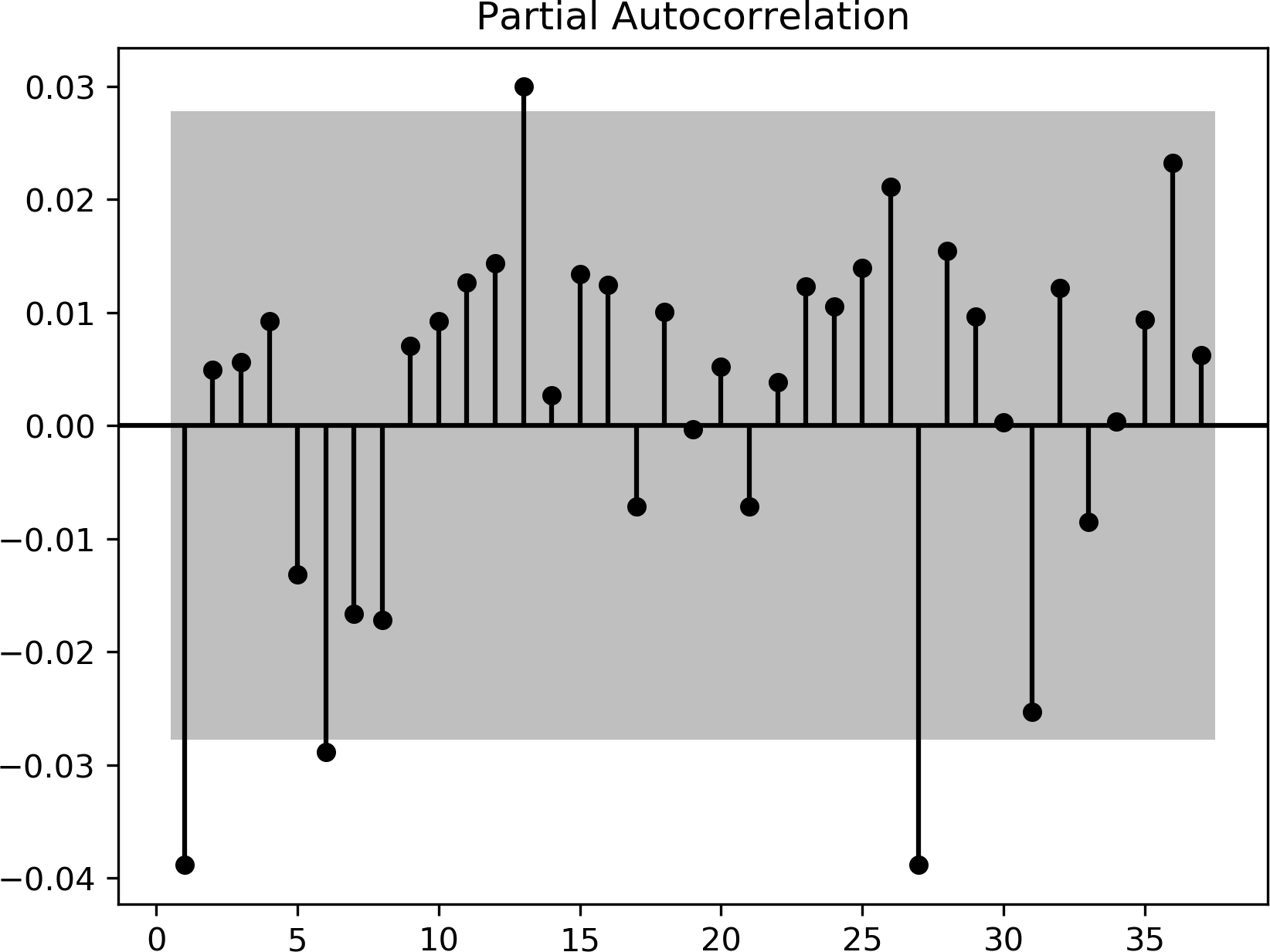
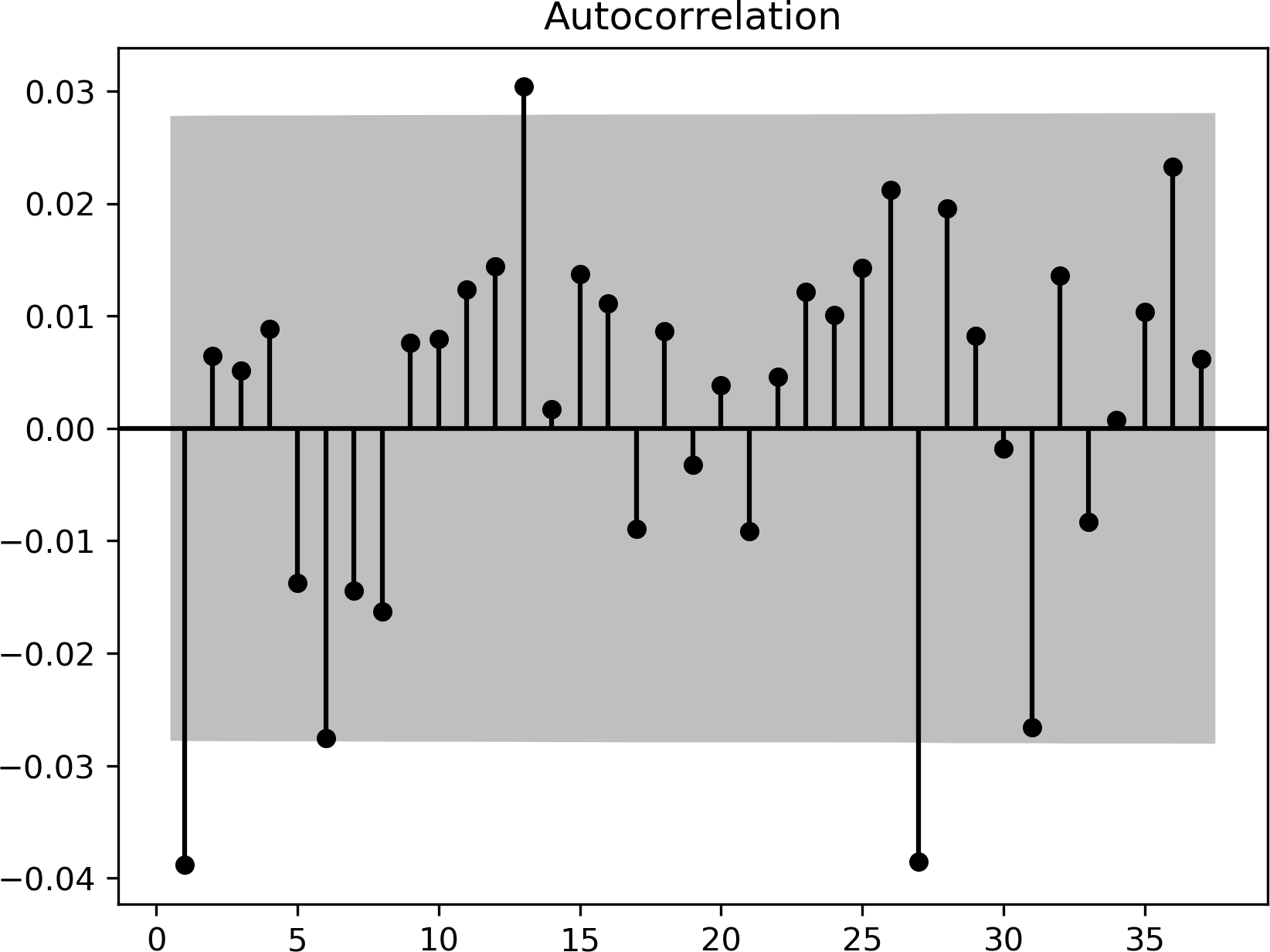
*i*=1

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)

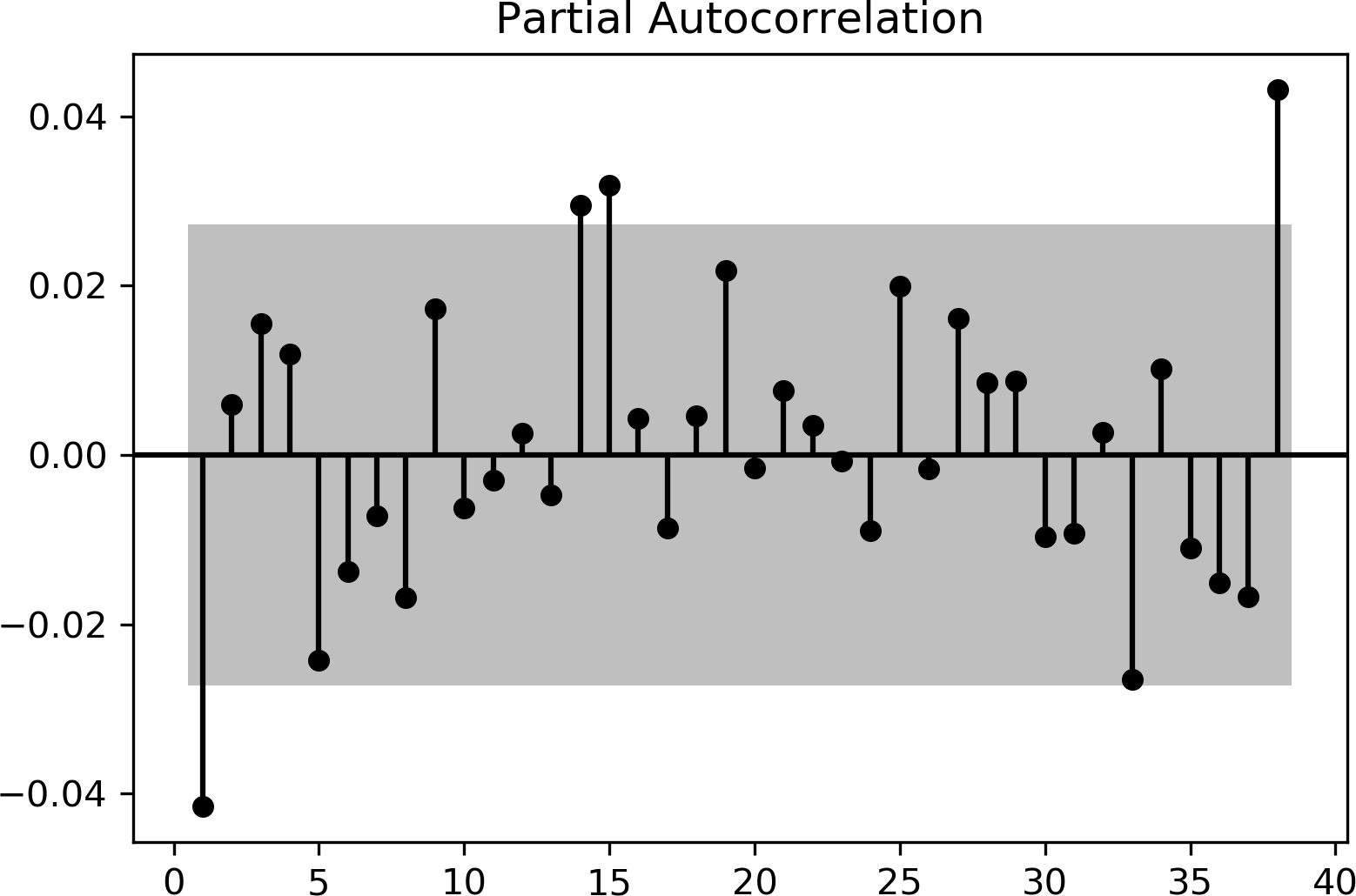
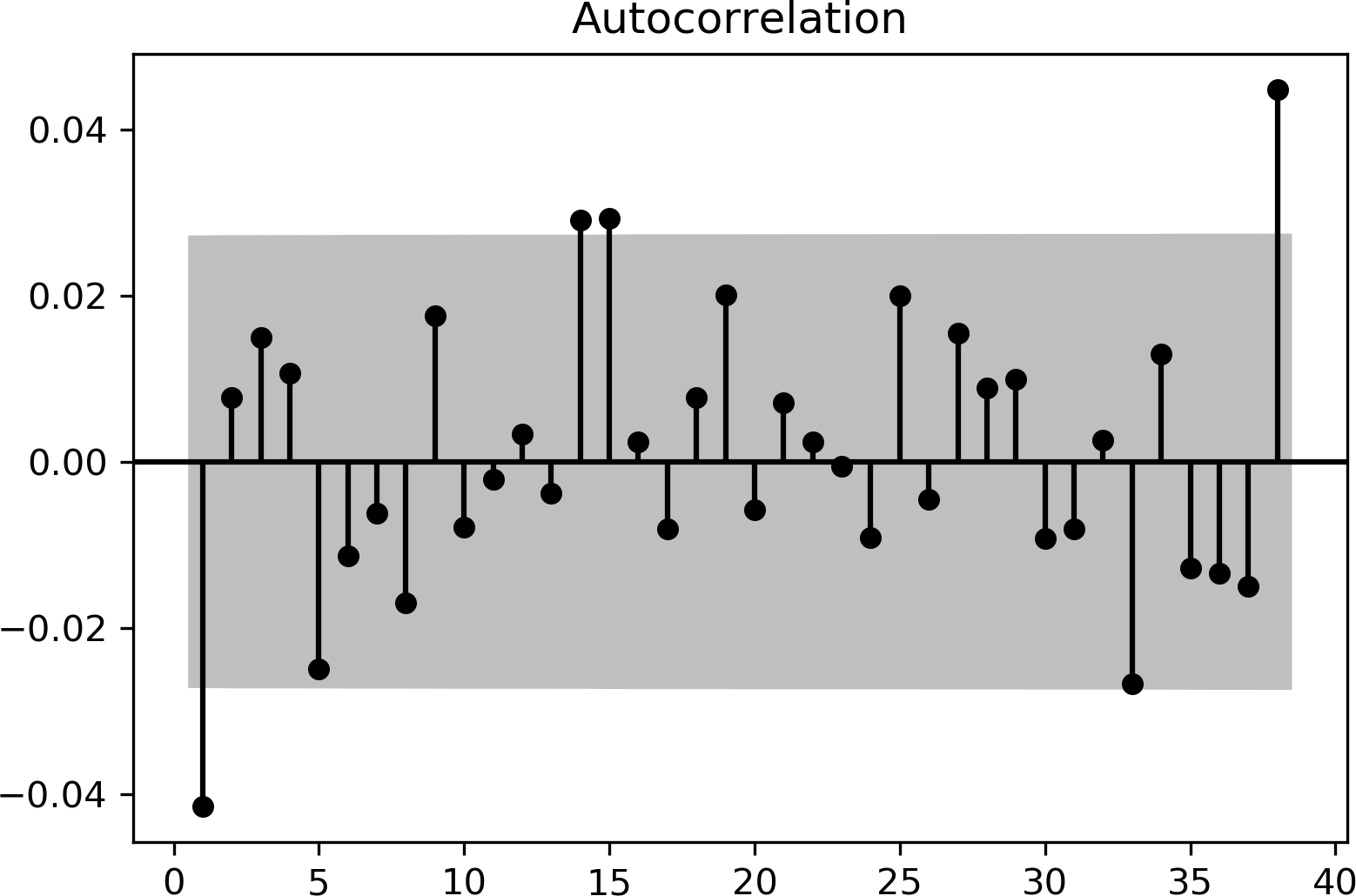


The autocorrelation function (ACF) and partial autocorrelation function (PACF) plots in figure [2](#_bookmark11) explore the inter-temporal correlation within the return series. Since only a few lags are statistically significant in the ACF and PACF plots, we do not expect linear time

series models are capable to achieve high performances in this return prediction task.

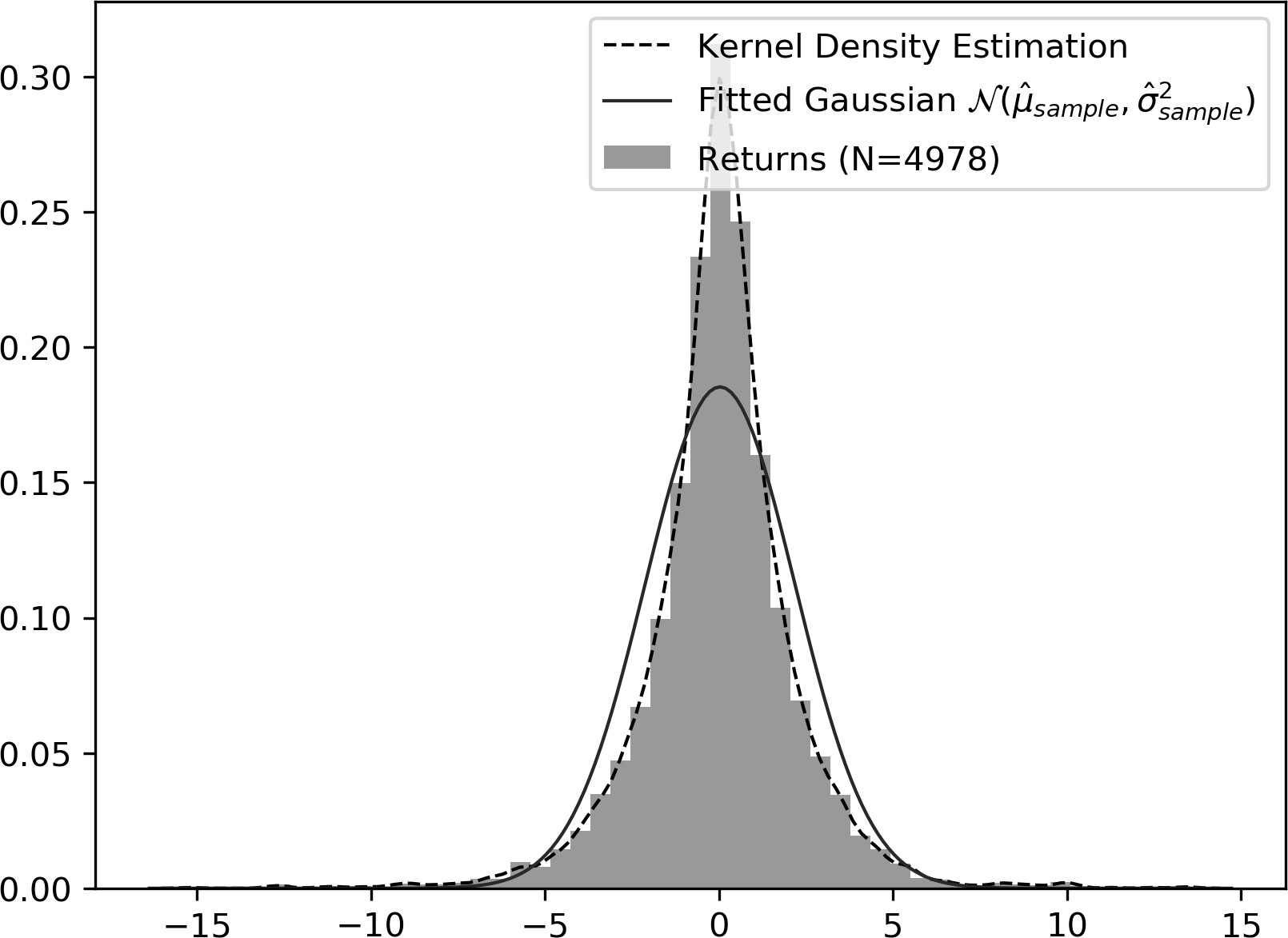
It is worth noticing that lag 1, 6, 13, 27 are significant in both ACF and PACF plots, which may indicate seasonalities with period of one week. However, regularity in missing data can lead to this observation as well. Hence, instead of dropping days with missing data, we fill up these missing data using random values from a Gaussian distribution parameterized by the mean and variance of the entire dataset. Figure [3](#_bookmark12) 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](#_bookmark13) 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.

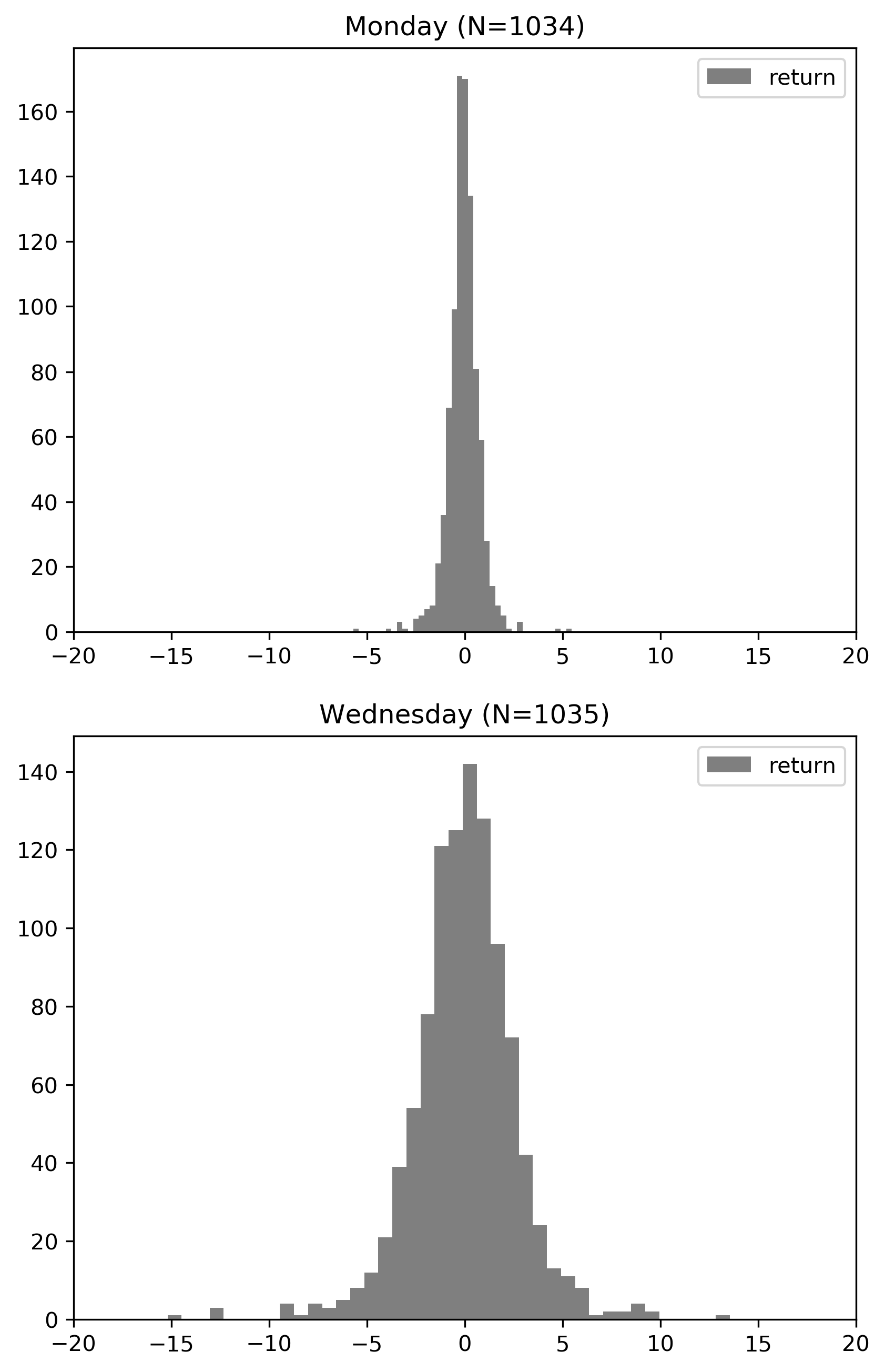
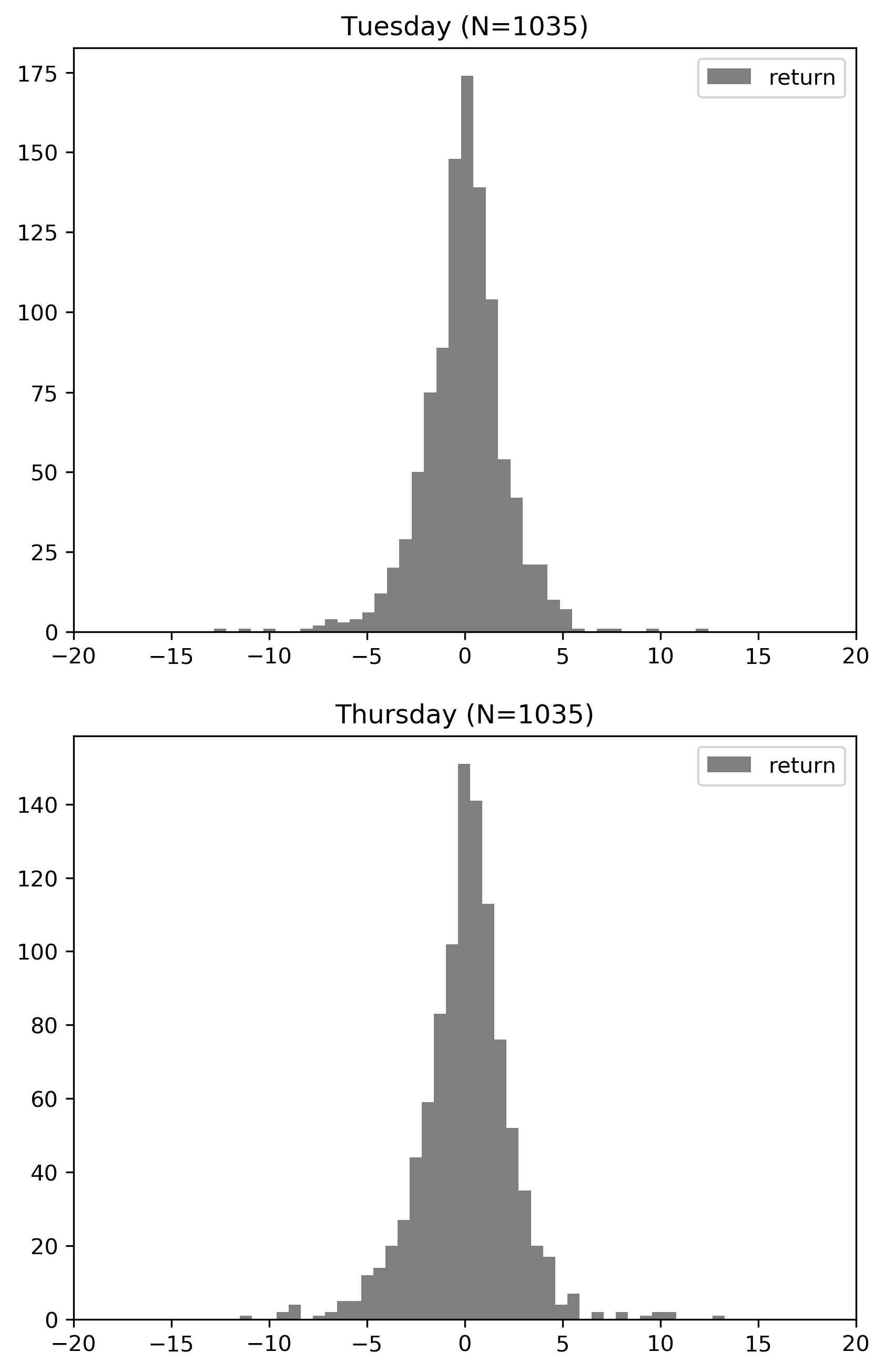
Figure 4: Distribution of Crude Oil Returns

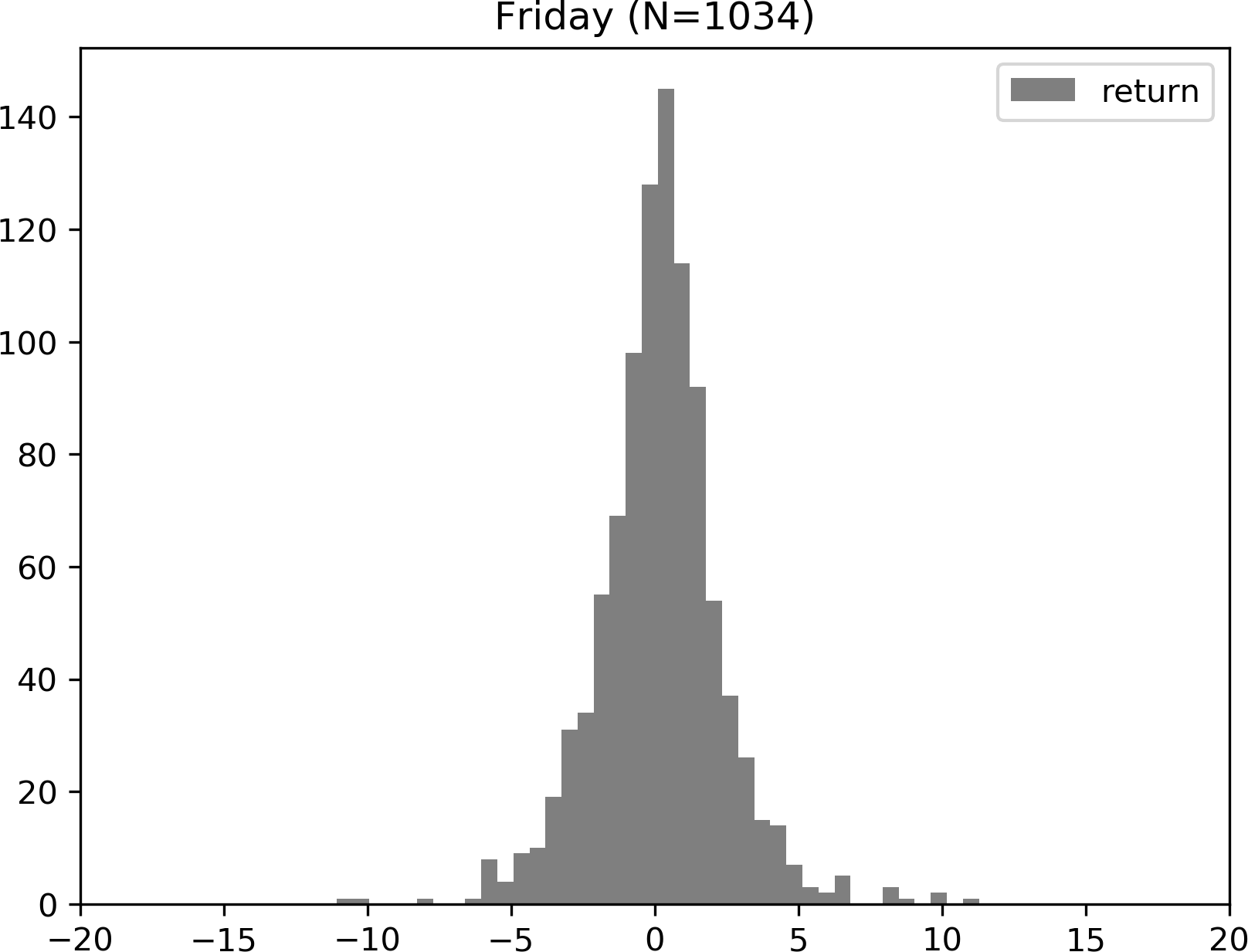


## Day of the Week Effect in Crude Oil Dataset

### 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).](#_bookmark111) Analysis in this paper unveils a similar daily seasonality presents in crude oil returns as well. Panels in figure [5](#_bookmark16) demonstrate the empirical distributions of returns on each day of the week. *N* s 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](#_bookmark17) below provide summary statistics for prices and returns on each day. [1](#_bookmark0) 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

1In table [4,](#_bookmark17) 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 *µ*0 = 0. Bold fonts indicate statistically significance at level *α* = 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 |  |  |  |

### 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](#_bookmark0) Firstly, the Kolmogorov–Smirnov test constructs the empirical CDFs *FMon,*927(*x*) and *FTue,*1018(*x*) from the dataset. Then, the Kolmogorov–Smirnov statistic measures the maximum discrepancy between two distribution functions, which is

*D* := sup *|FMon,*927(*x*) *− FTue,*1018(*x*)*| ∈* [0*,* 1] (3.6)

*x*

A smaller *D*-statistic implies stronger distributional similarity between two distributions. For instance, when *FMon,*927(*x*) and *FTue,*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,

2Different alternative hypotheses can be used in Kolmogorov–Smirnov test: i) *H*1 : *F* (*x*) *≥ G*(*x*), ii)

*H*1 : *F* (*x*) *G*(*x*), and iii) *H*1 : *F* (*x*) *G*(*x*). This paper is using the third (two-tailed) alternative hypothesis.

*≤*

there distributions are completely different, and *DX,Y* = 1. The test rejects *H*0 at a significance level of *α* if

2

2

*nm*

*D > −*1 ln *α n* + *m*

(3.7)

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 indi- cates the null hypothesis is rejected at a significance level of 0.01, which implies discrepancy in distributions.

Table [5](#_bookmark19) 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.

## 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](#_bookmark21) 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](#_bookmark22) below summarizes the average numbers of news on each day of the week, and table [8](#_bookmark23) 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

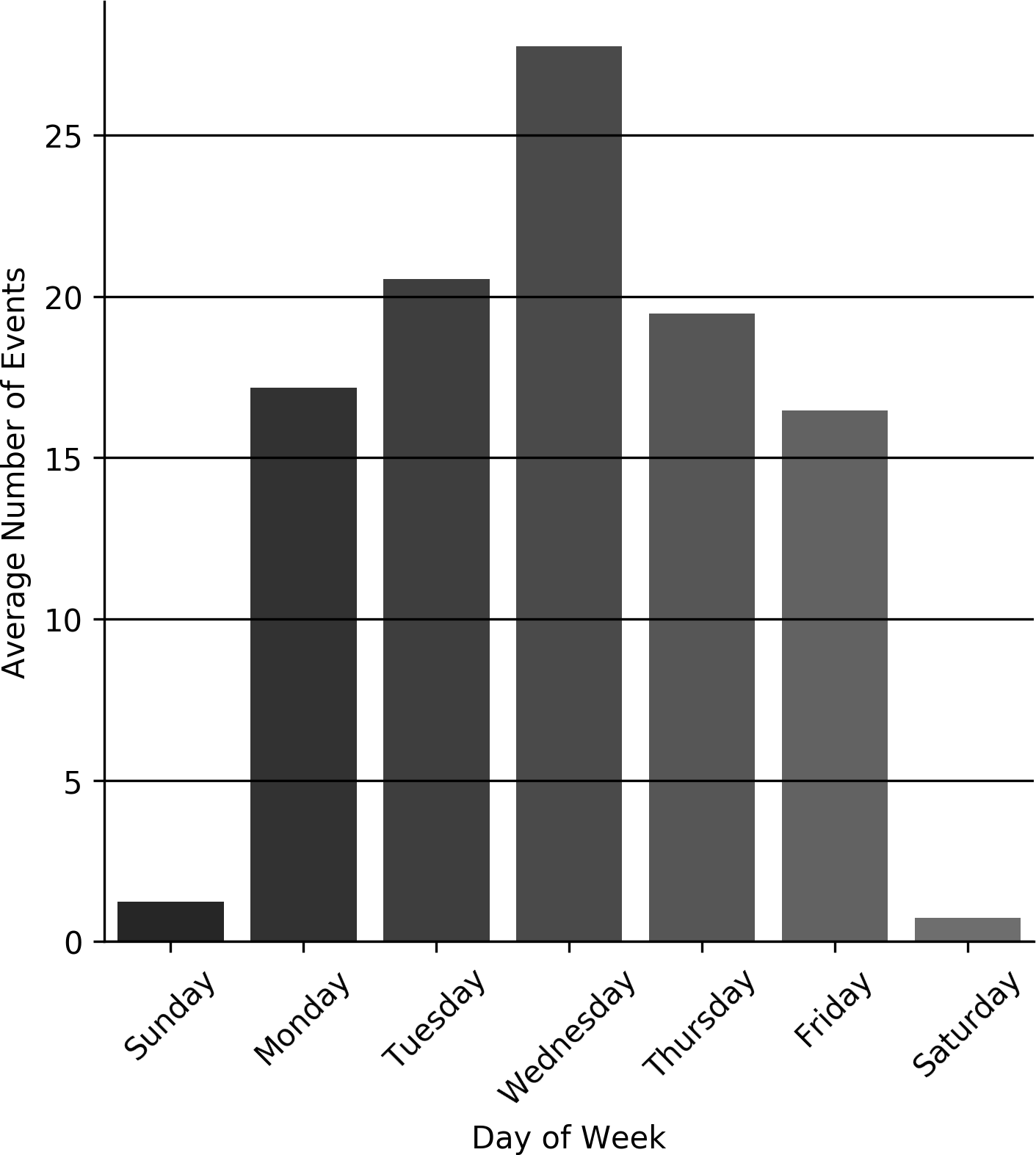


Table 8: Average Numbers of News on Each Day of Week in Each Year

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |

### 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](#_bookmark25) 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,](#_bookmark26) 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 lit-

erature here

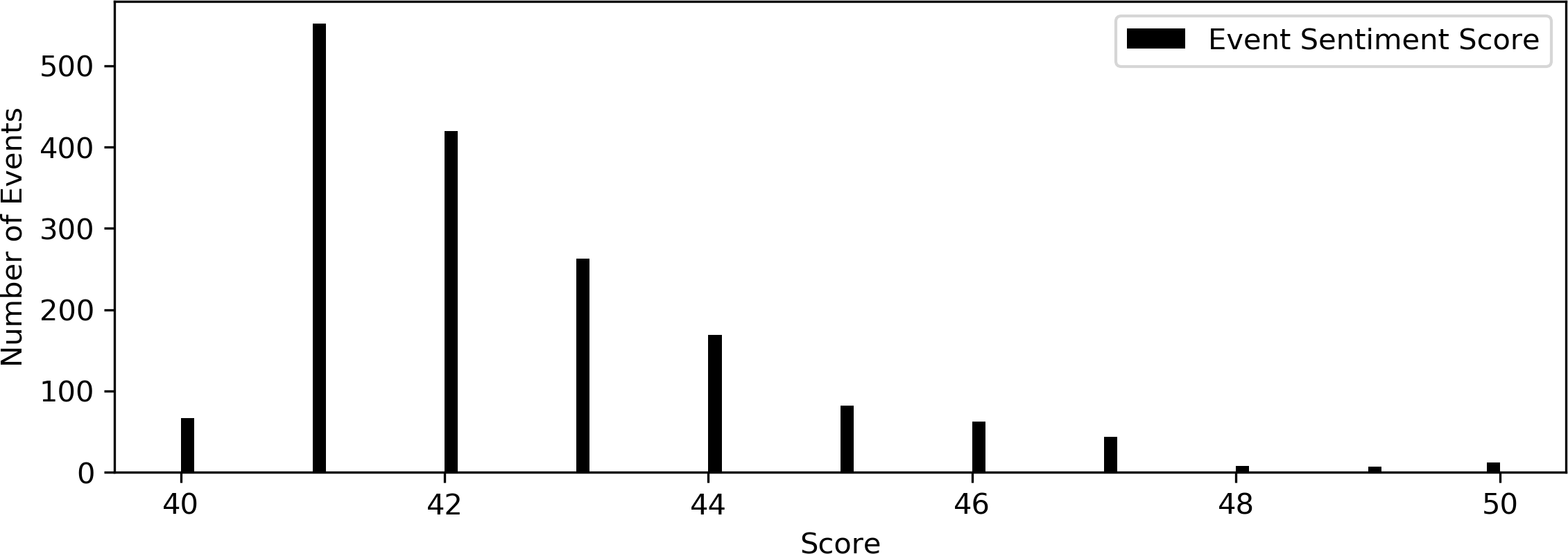
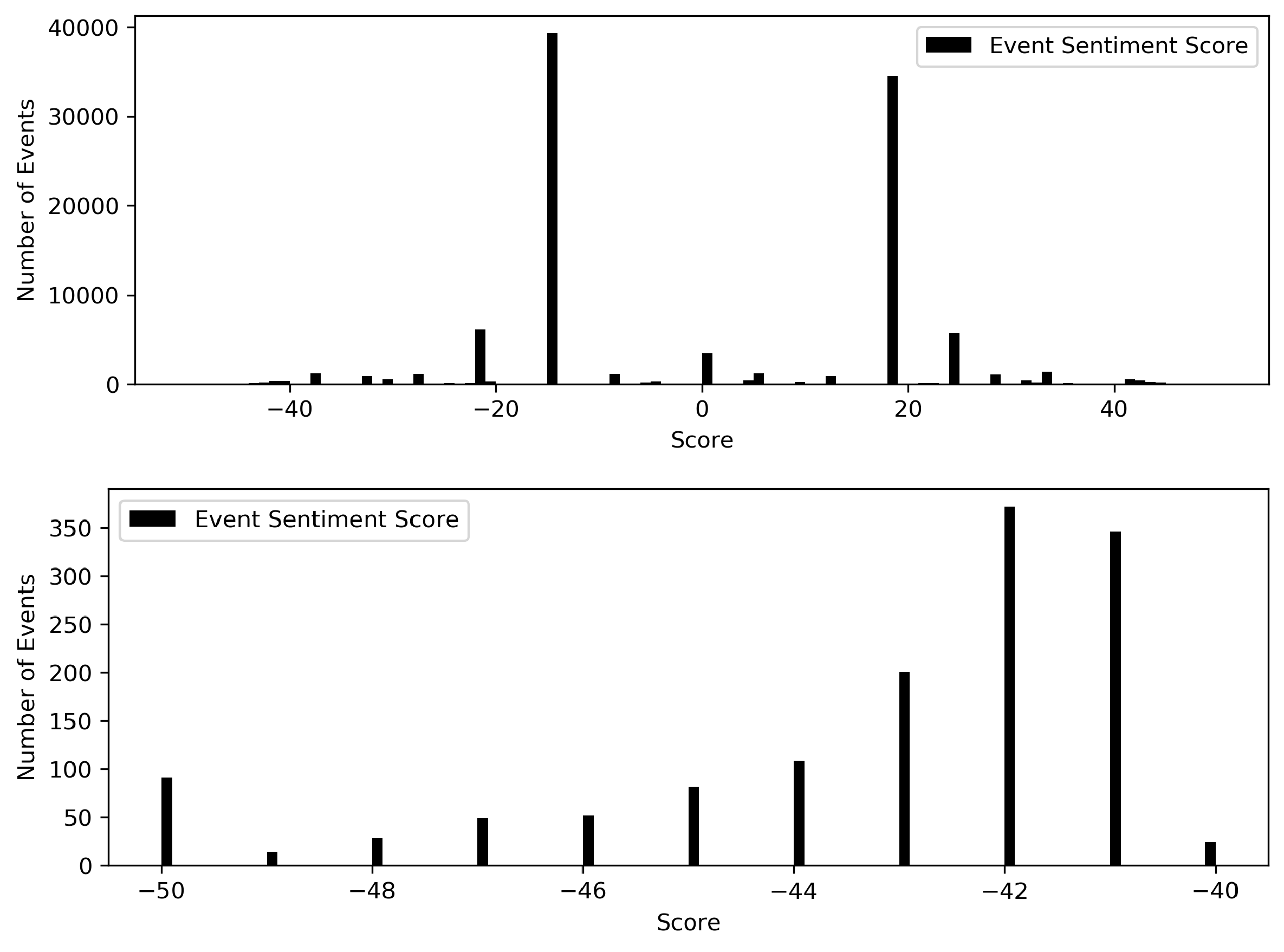
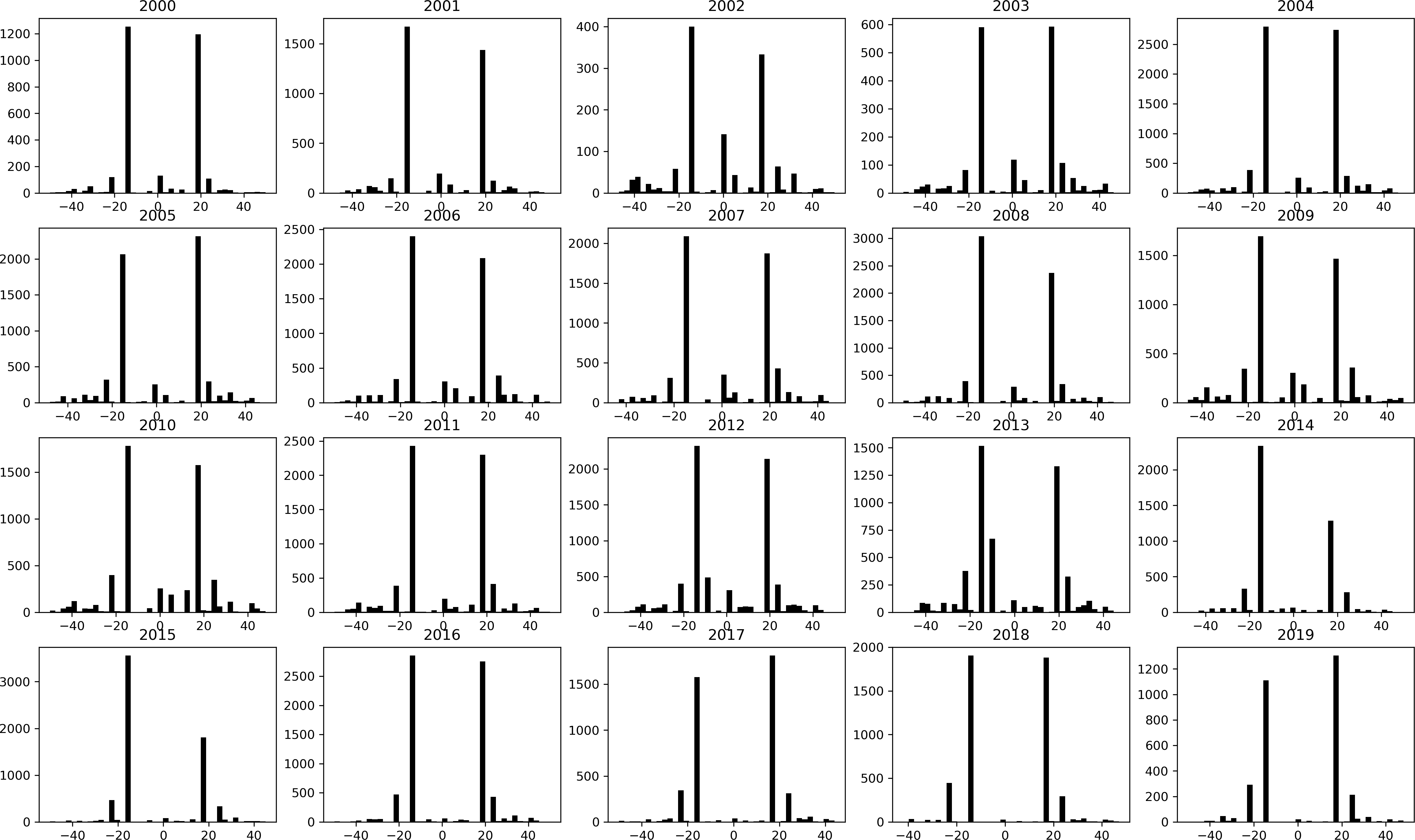
Figure 7: Distribution of Event Sentiment Scores (ENS)

Figure [8](#_bookmark26) 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

### 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

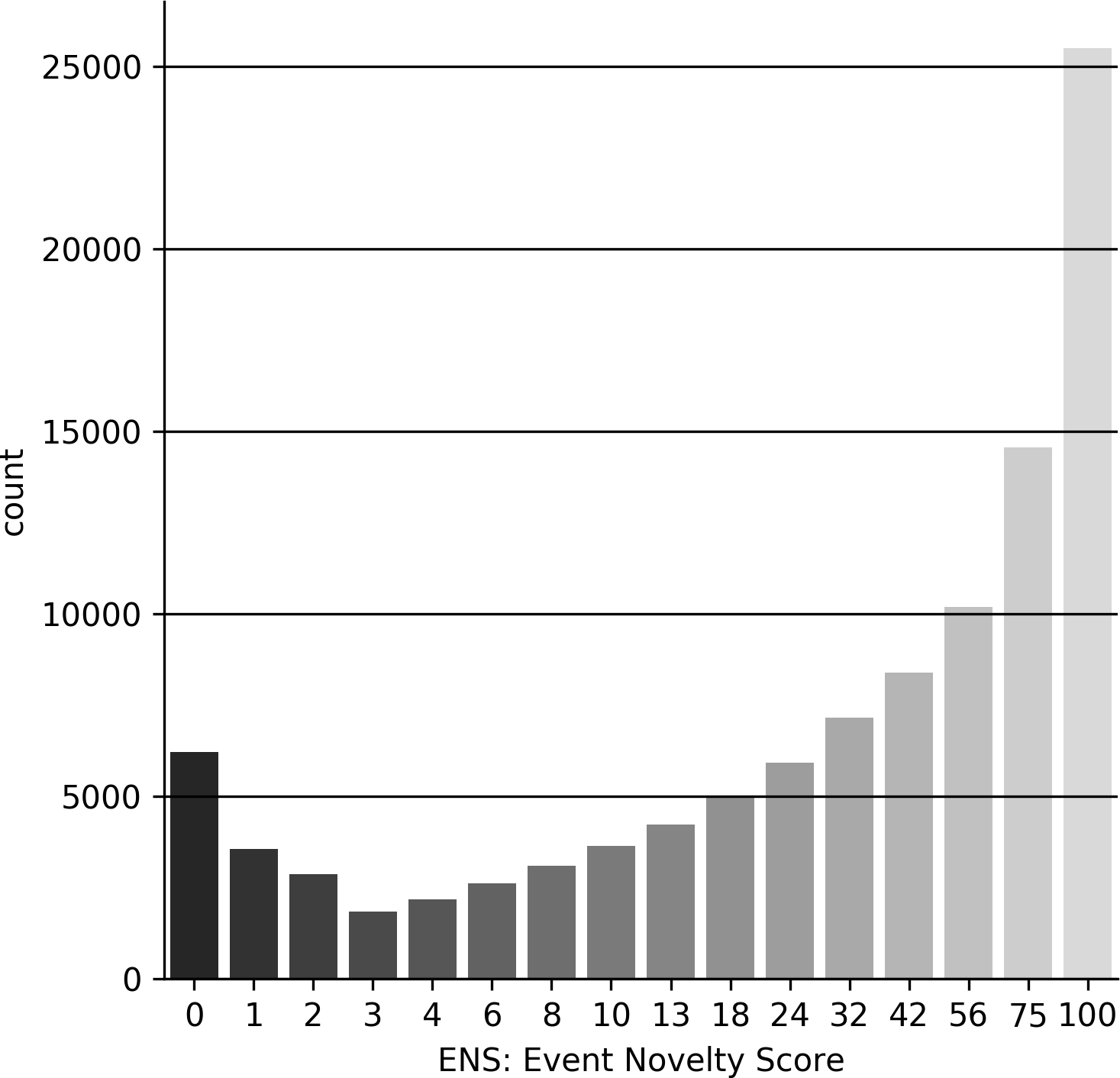
a.m. January 10. After 30 minutes (11:30 a.m., January 10) one news source reports this export cut, and this news article is one entry in the dataset. To determine the ENS of this news article, the algorithm looks into the 24-hour period prior to the news arrival, that is, from 10.30 a.m. January 9 to 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

ENS*i* = 100 *×* 0*.*75*k* (3.8)

Figure [9](#_bookmark28) 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.

WESS := ESS *×* ENS

100

(3.9)

We divide the product of ESS and ENS in equation [(3.9)](#_bookmark29) 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](#_bookmark30) illustrate the overall distributions of WESS as well as the two tails of it. It turns out that the clustering pattern in figure [7](#_bookmark25) 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)

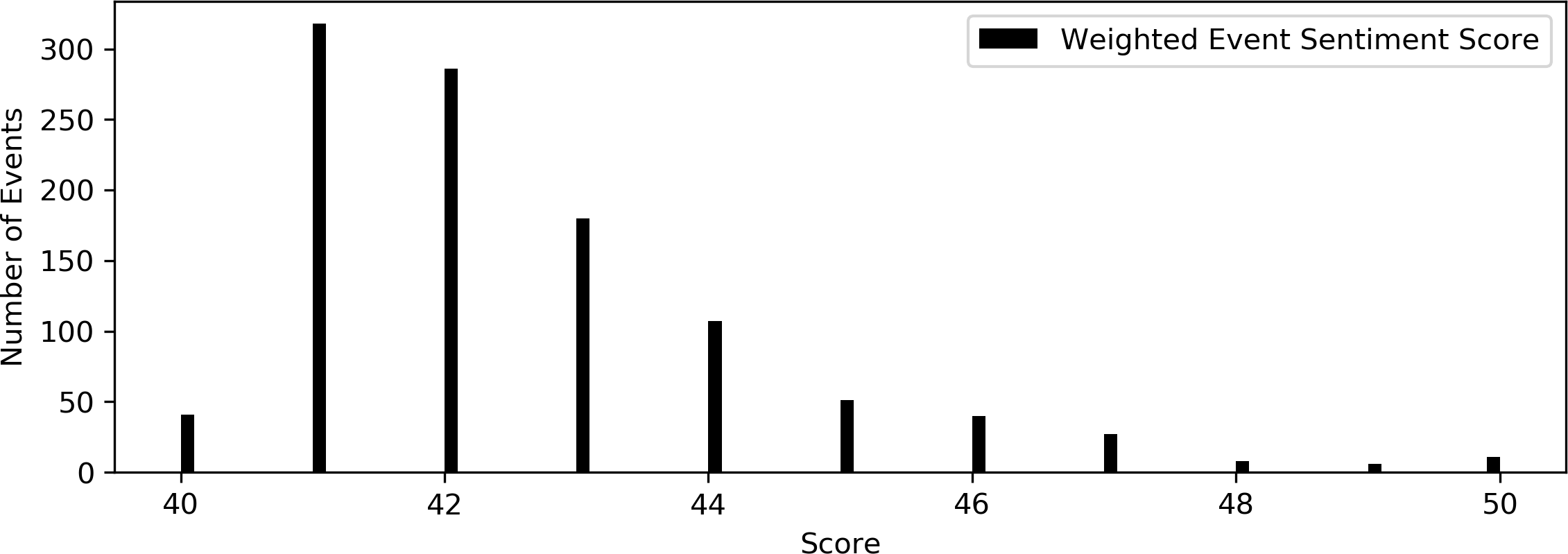
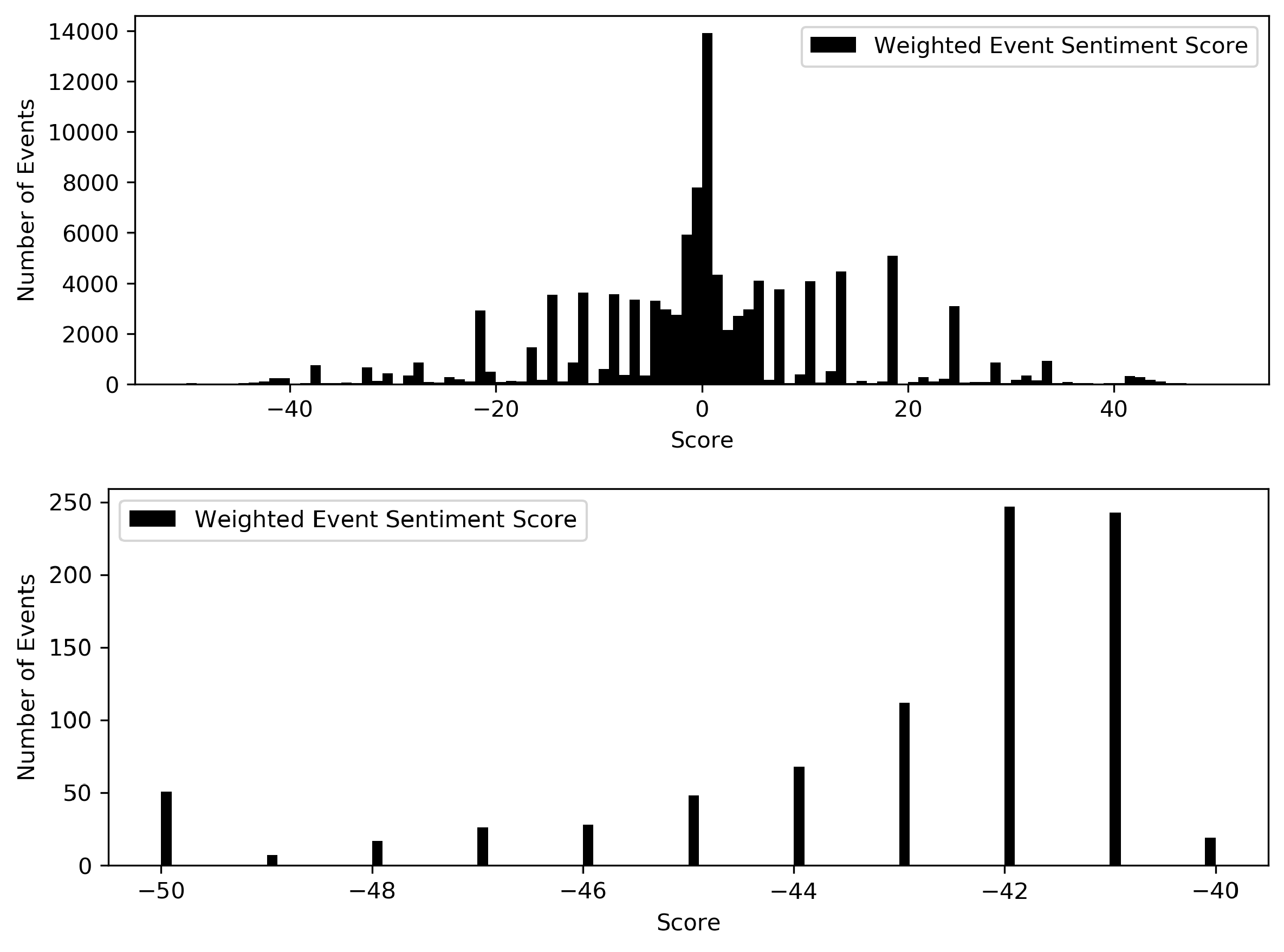
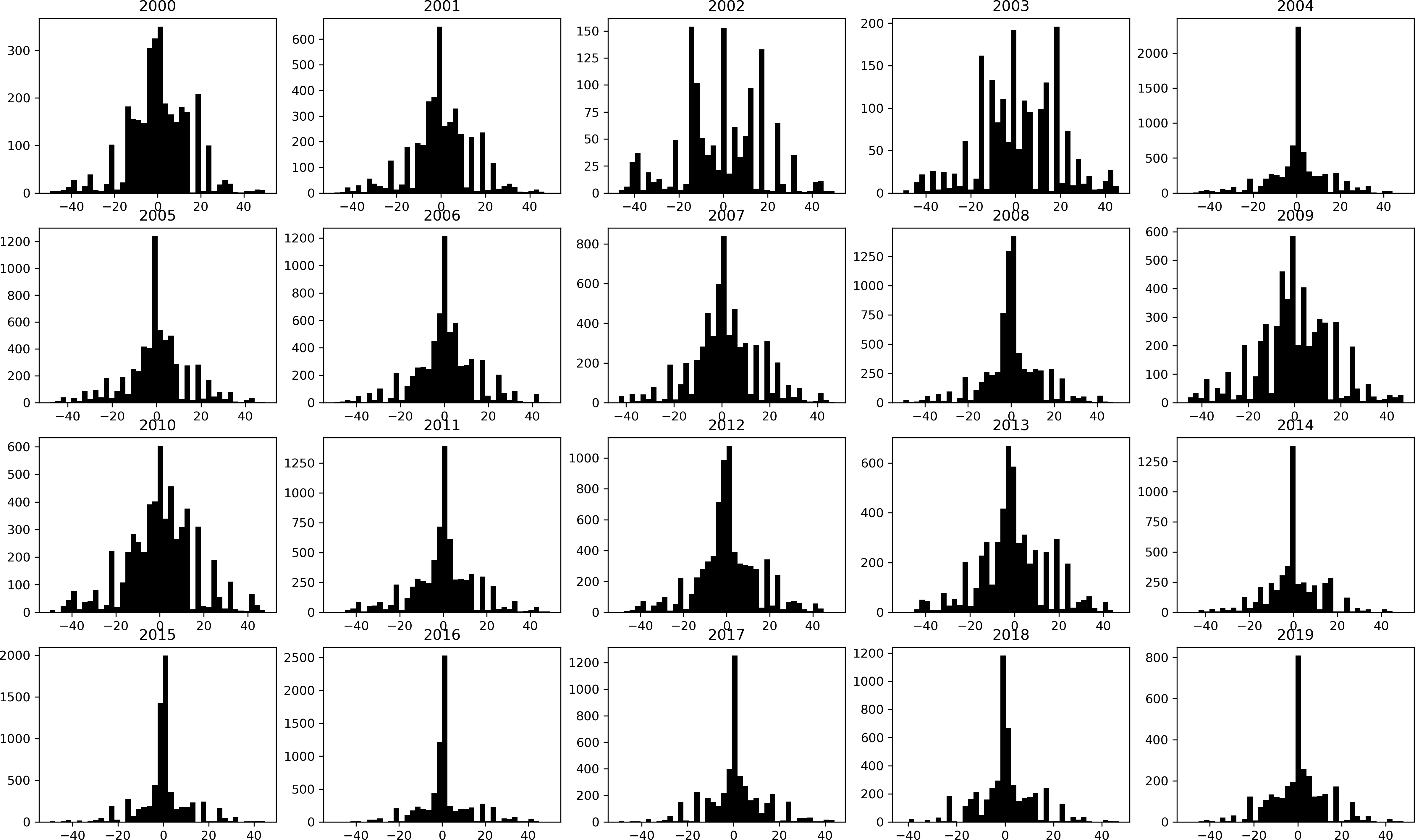


Figure [11](#_bookmark31) plots the yearly distributions of WESS. The yearly distributions suggest that there are more events with negative sentiments in 2001 as well as in 2008-2009, such obser- vation matches the US recession records in figure [1.](#_bookmark9)

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



### 3.4.3 News Arrival Time

Add cita-

tions 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](#_bookmark33) 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

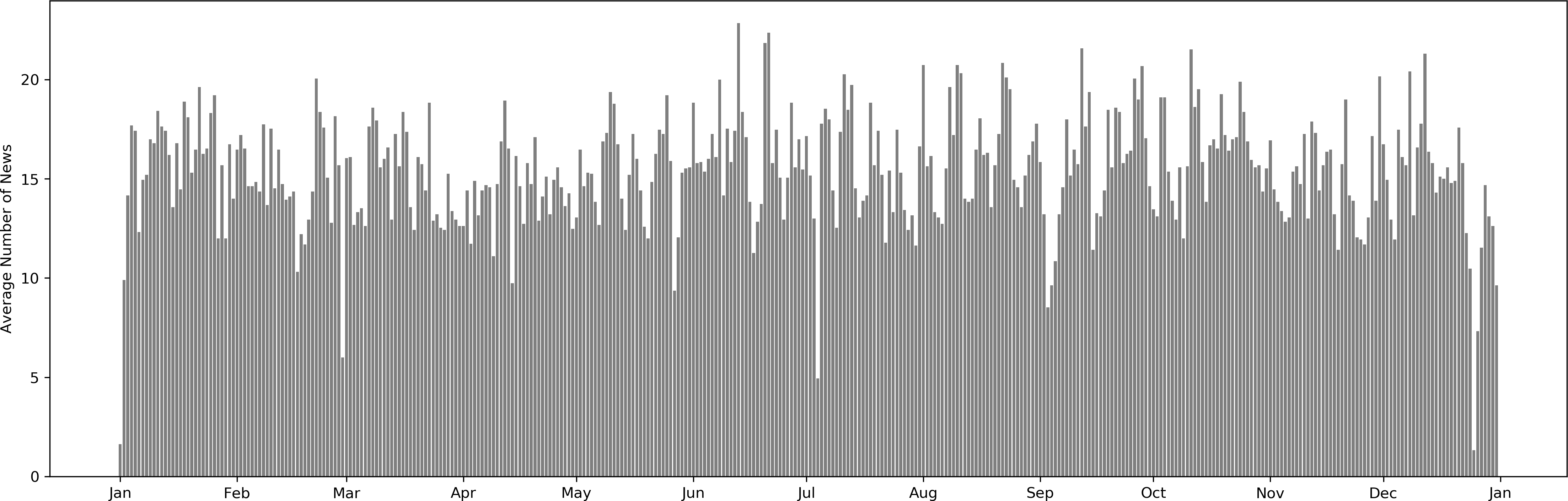
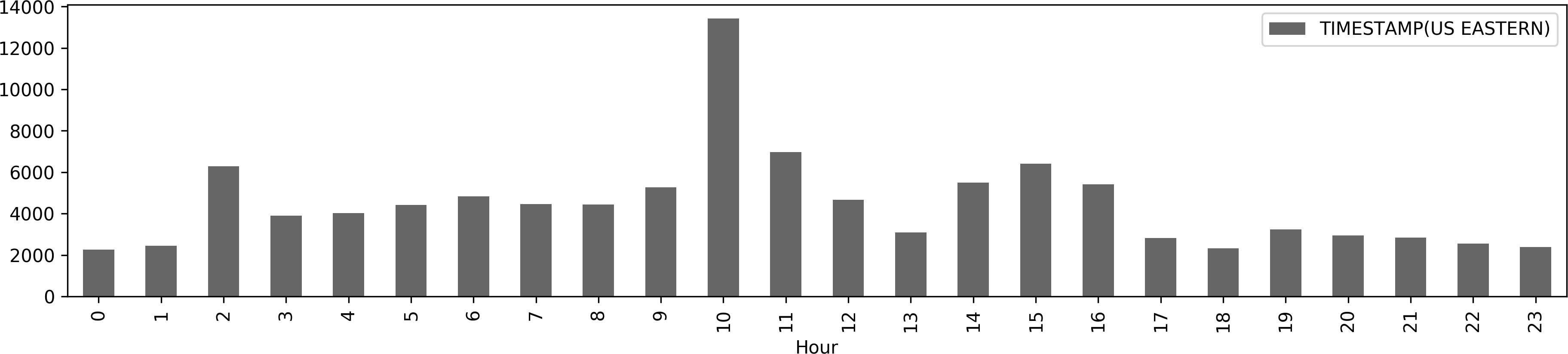


Figure [13](#_bookmark34) has a look at the distribution of news arrival within 24 hours. It is worth noticing that, in the RPNA dataset, the original timestamps recording when news arrives are using Universal Time Coordinated (UTC). To incorporate the crude oil dataset, we convert raw timestamps to Eastern Standard Time (EST) timezone[3](#_bookmark0), 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

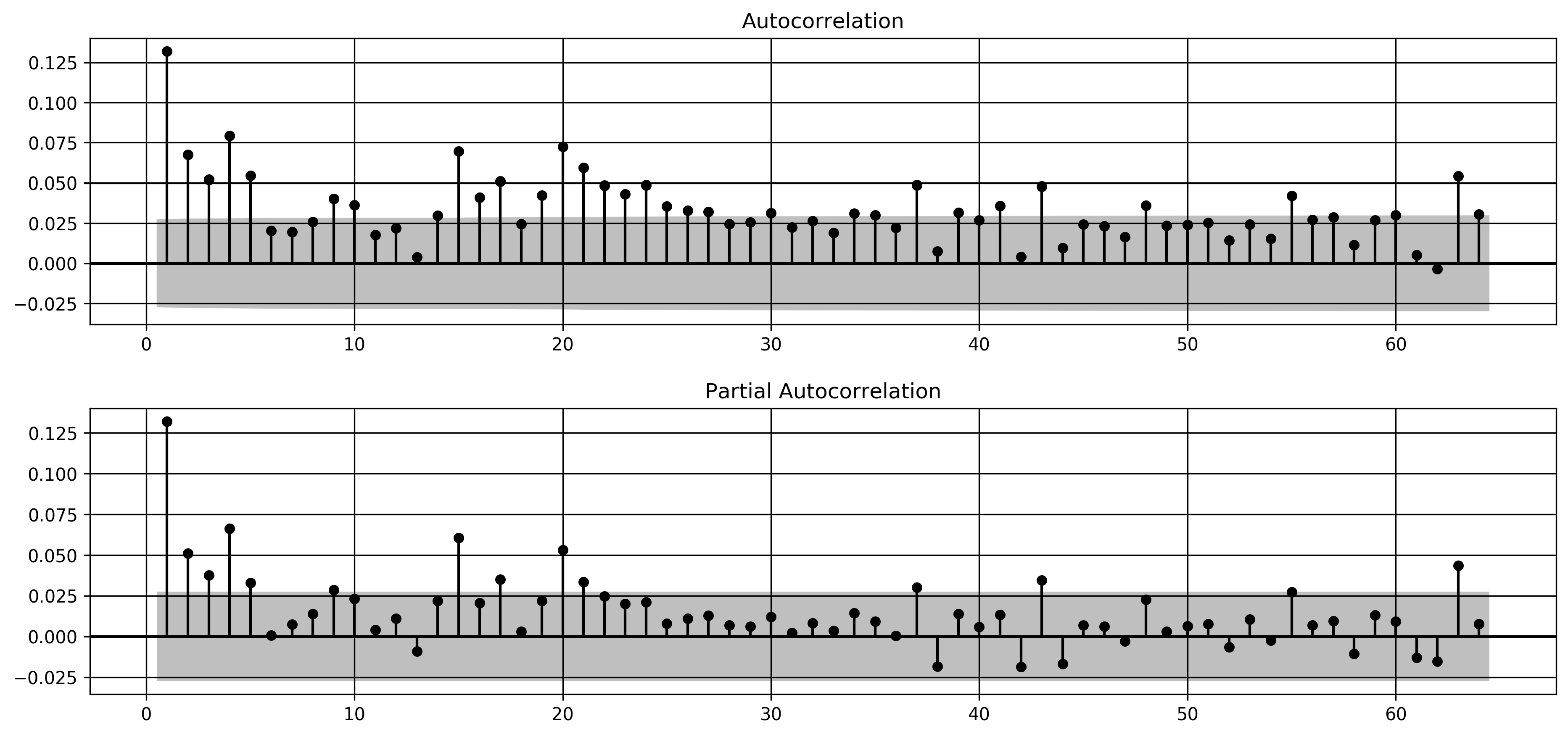


In order to have a closer look at the intertemporal correlation of event sentiment, this

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

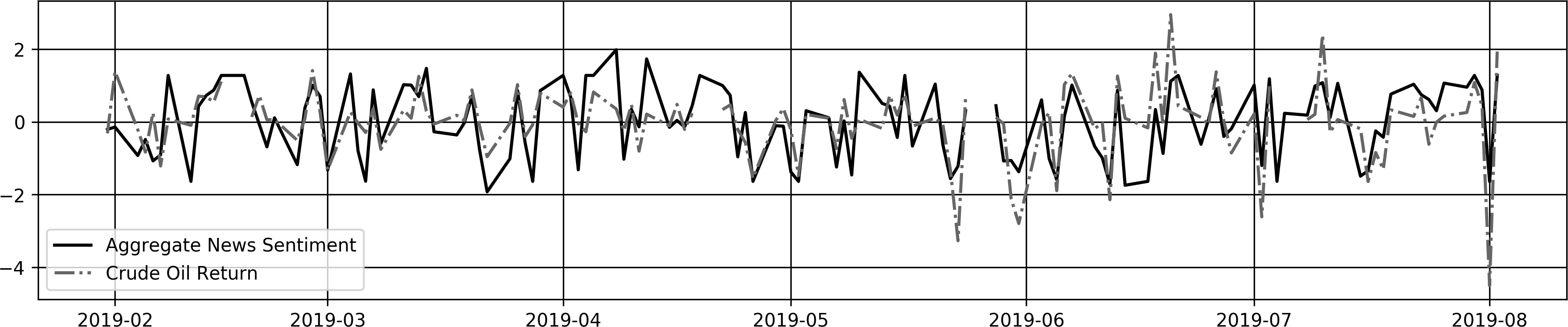
paper firstly compute the mean event sentiment score of all events within each day, denoted as ESS and WESS respectively, in the span of 20 years. The ACF and PACF plots of the daily average ESS in figure [14](#_bookmark35) suggest the intertemporal correlation here is much more salient than the series of returns, which has only a few significant lags.

Figure 14: ACF and PACF of ESS



Moreover, there exists significant correlation between the price movement series and news sentiment. Figure [15](#_bookmark36) 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 ESS and Return in 2019



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

crude oil returns.

## 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](#_bookmark25) and [10),](#_bookmark30) 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 ≥* 0. Afterward, the algorithm classifies all news based on their (weighted) event sentiment scores. News with score (W)ESS *∈* [*−*50*, −r*) are negative news, and all news with (W)ESS *∈* (*r,* 50] are positive news, and, news in [*−r, r*] are neutral news. Figure [16](#_bookmark38) and figure [17](#_bookmark39) 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

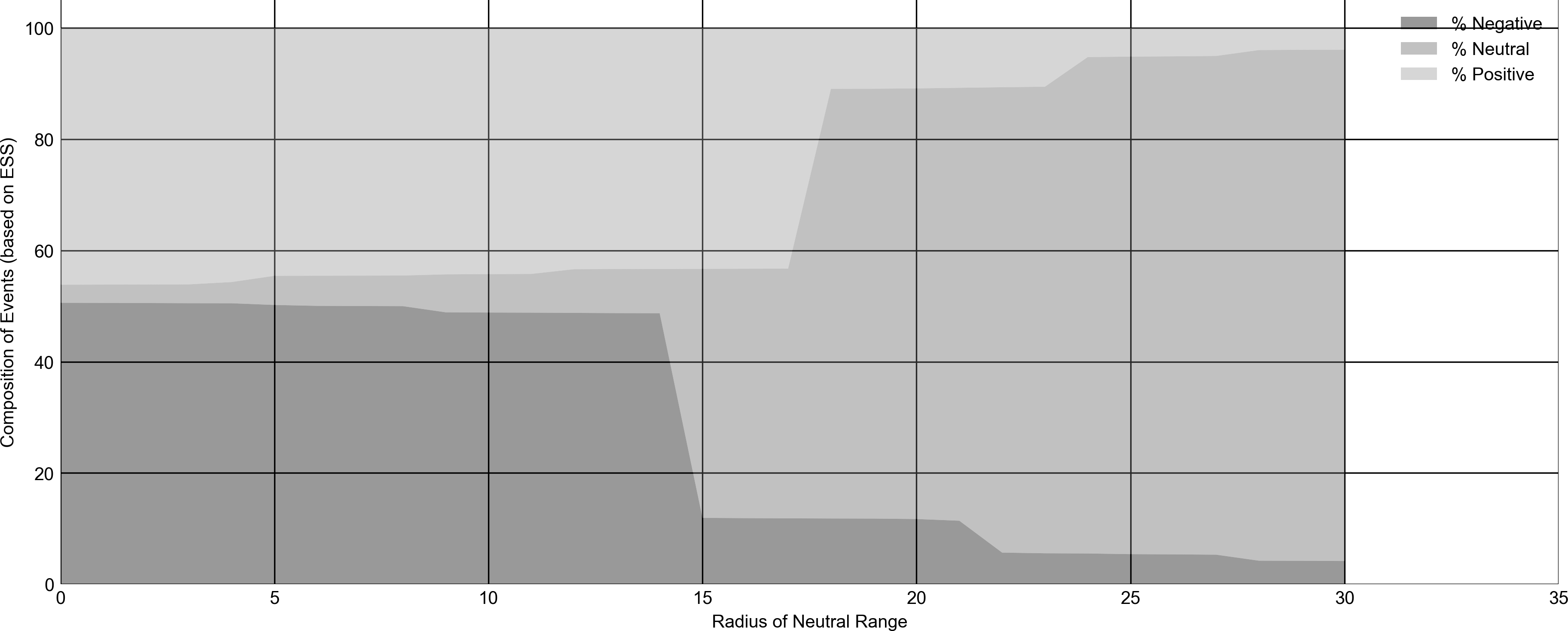
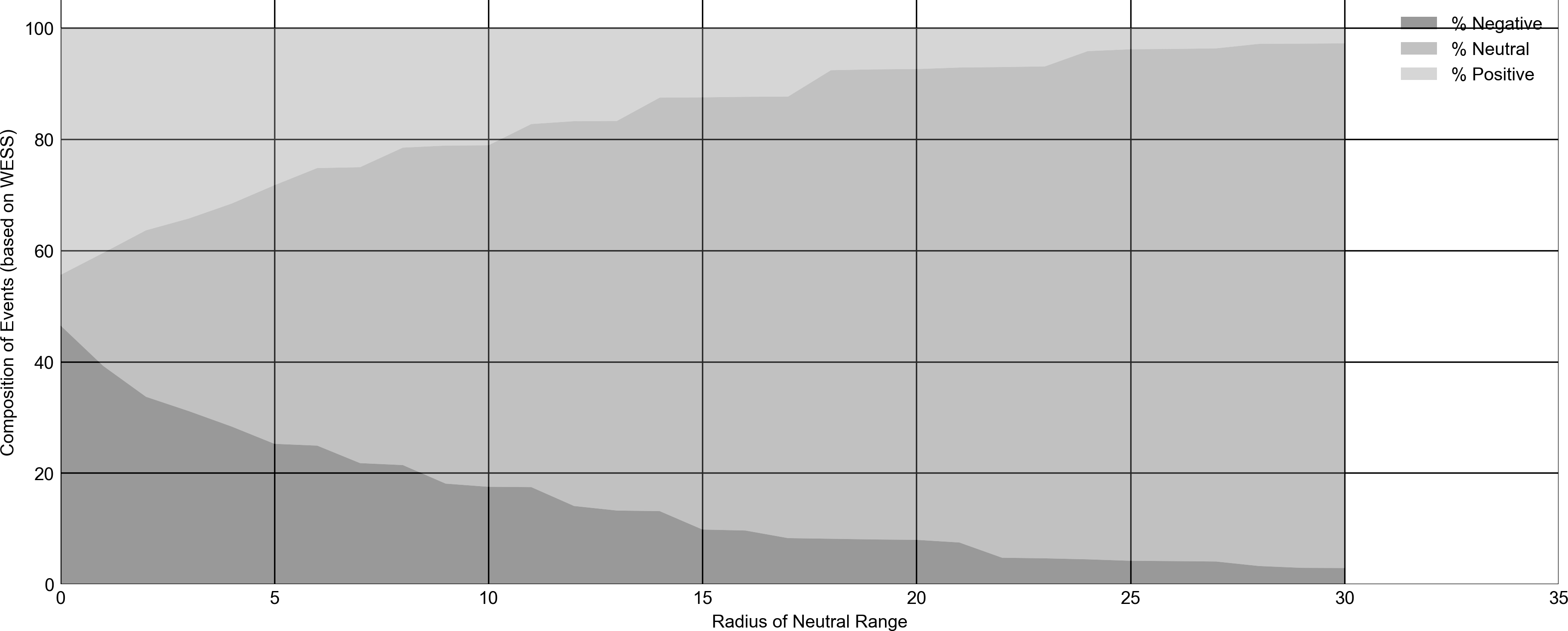


Figure 17: Composition of News Type based on WESS



In figure [16,](#_bookmark38) 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.](#_bookmark25) Table [9](#_bookmark40) and table [10](#_bookmark41) 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](#_bookmark40) (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 [32](#_bookmark123) and [33](#_bookmark124) 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

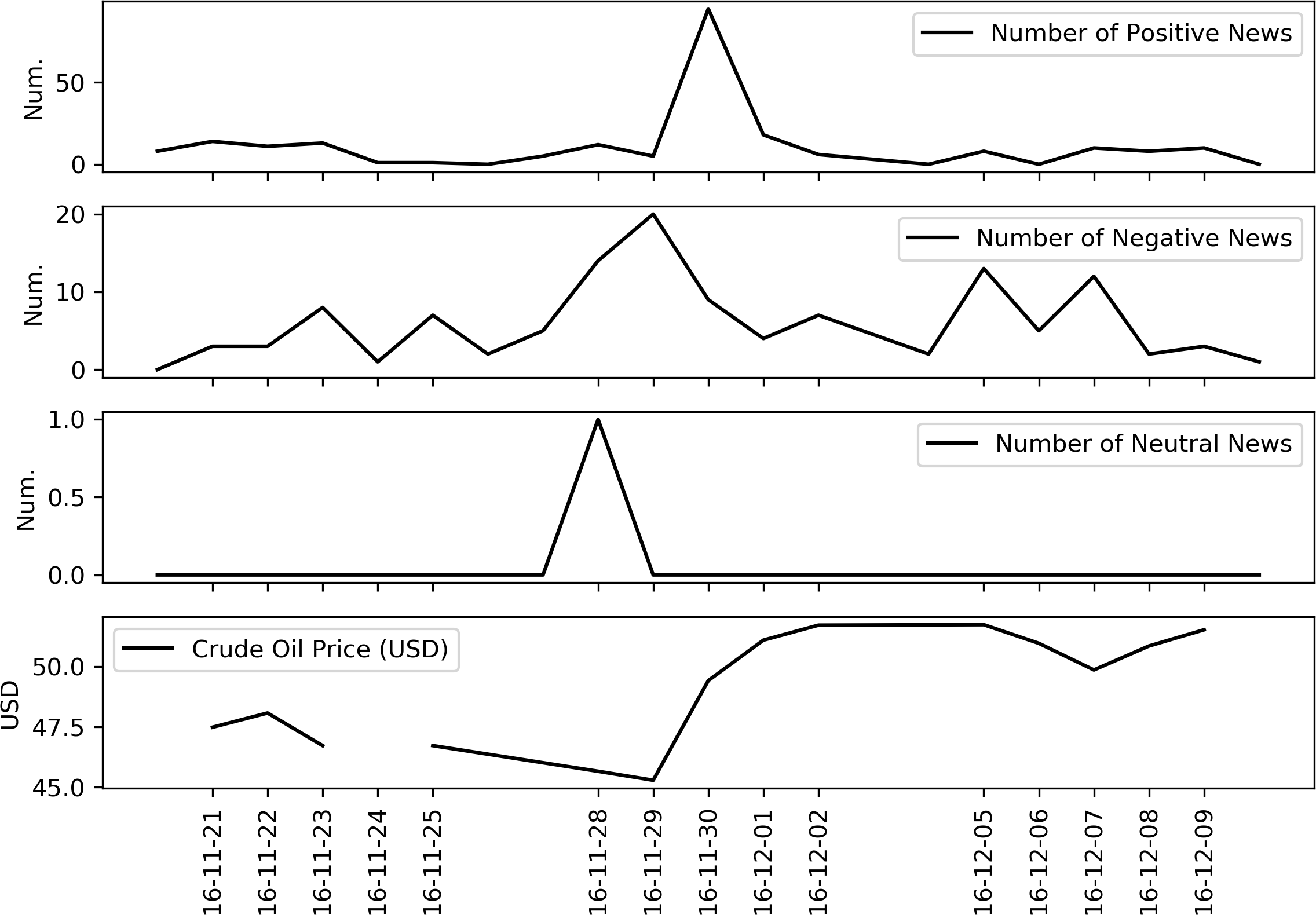
|  |  |  |  |
| --- | --- | --- | --- |
| *r* | Num Negative | Num Neutral | Num Positive |
| 0 | 46.54% (100.00%) | 9.06% (100.00%) | 44.40% (100.00%) |
| 0*.*3 | 42.85% (92.08%) | 13.99% (154.43%) | 43.16% (97.19%) |
| 1 | 39.25% (84.33%) | 20.32% (224.32%) | 40.43% (91.06%) |
| 3 | 31.12% (66.87%) | 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%) |

## Case Studies

### 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.

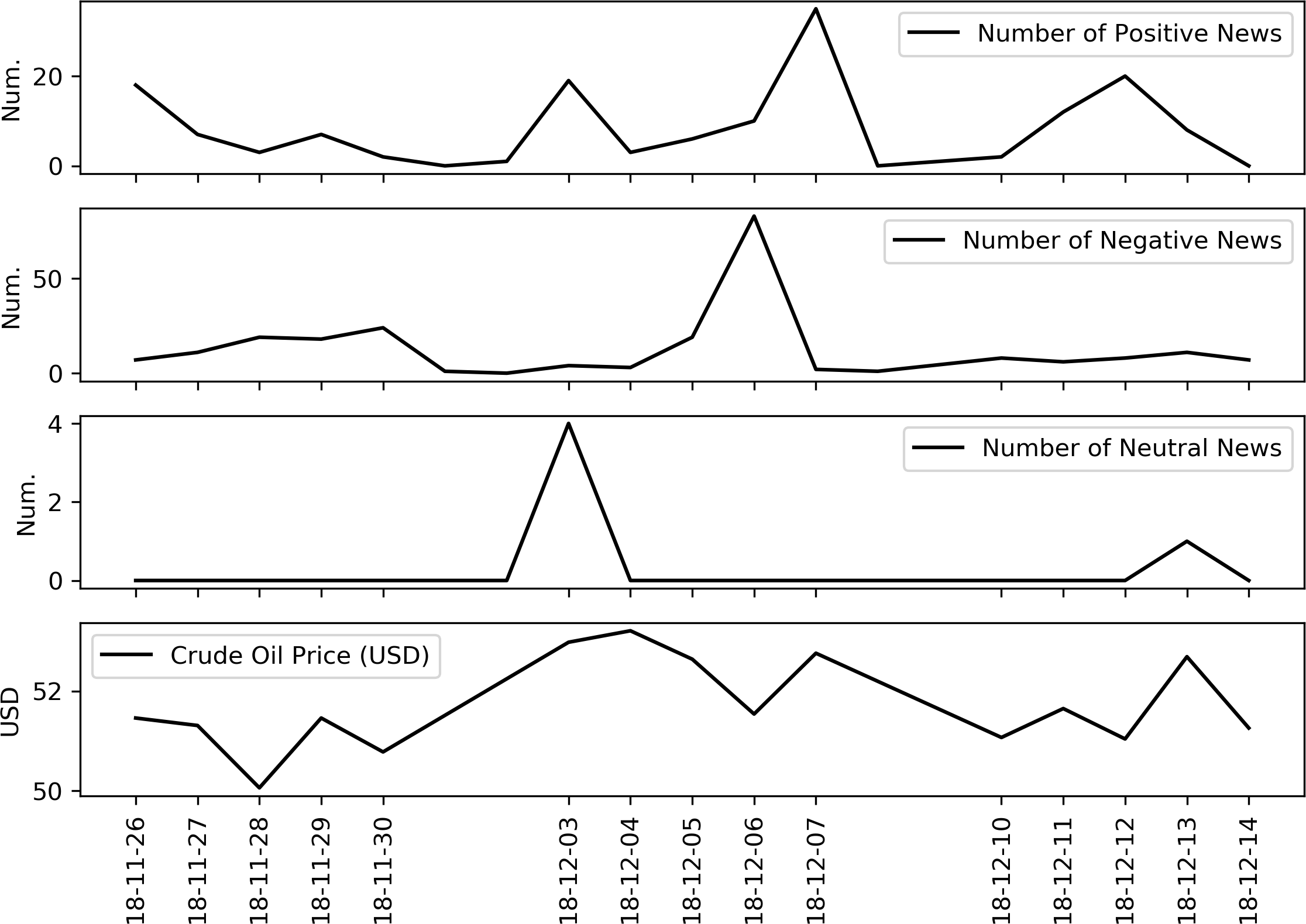
Figure 18: Crude Oil Price and Number of Events within 10 Days



### 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



### 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

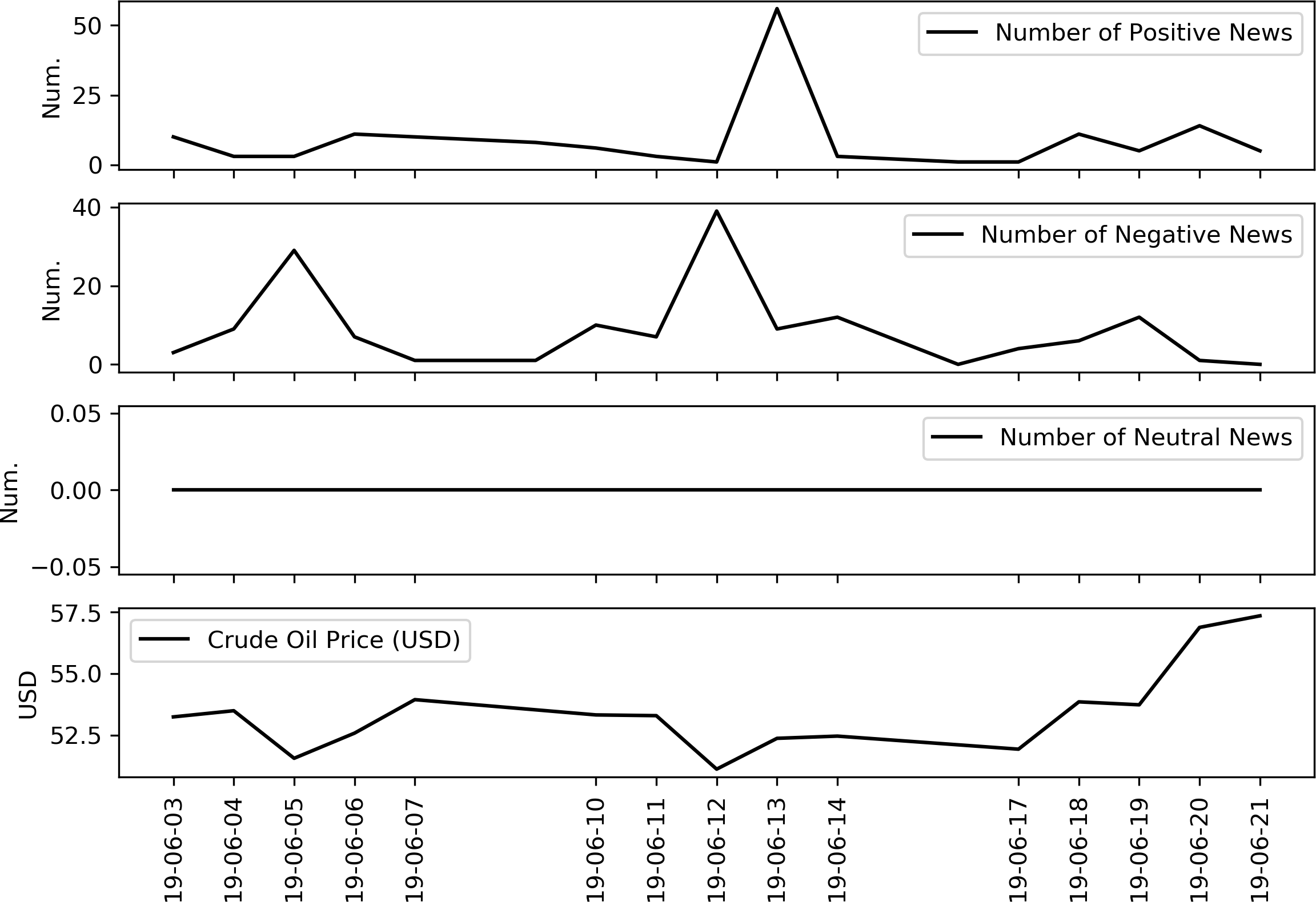
The crude oil price had been decreasing since April 2019. The tension between the US source

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,](#_bookmark46) 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



# Model

## Framework

Figure [21](#_bookmark50) illustrates the framework of our model as a directed acyclic graph (DAG). In a DAG, an arrow *X → Y* indicates *p*(*X, Y* ) = *p*(*Y |X*)*p*(*X*). Intuitively, one may interpret *X → Y* as *X* is causing *Y* .

On each day *t*, the real state of world is denoted as *ωt*, 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 *ωt* 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 *ωt*.

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, **x***t*. Ideally, **x***t* should provide a finer summary, especially sentiments, of the IF*t* as well as *ωt*. Sometime we refer to **x** as the quantified information flow and IF as the abstract information flow.

The last component is the realized return *rt*. The true state *ωt* is determining the actual realized return *rt* on day *t*. Moreover, since traders are often reacting to news reports, so the quantified information flow **x***t* is affecting *rt* as well.

## Formal Framework

Figure 21: The Framework

*· · ·*

*ω*0

*Q*

*ω*1

*Q*

*ρ*

*ρ*

IF0

IF1

*ϕ*

*ϕ*

**x**0

*G*

*P*

**x**1

*P*

*E*

*R*

*E*

*R*

*r*0

*r*1

*Q*

*ωt*

*Q*

*ρ*

IF*t*

*G*

*P*

*ϕ*

**x***t*

*P*

*R*

*E*

*R*

*rt*

*· · ·*

*· · ·*

*G*

*· · ·*

*· · ·*

*· · ·*

*· · · · · ·*

### Timestamps

In the following discussion, this paper uses non-negative real numbers, *t ∈* 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 *tth* 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 + 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

### States of World

Figure [21](#_bookmark50) illustrates the framework of our model as a directed acyclic graph (DAG). In a DAG, an arrow *X → Y* indicates *p*(*X, Y* ) = *p*(*Y |X*)*p*(*X*). Intuitively, one may interpret *X → Y* as *X* is causing *Y* .

On each day *t ∈* Z+, the real state of world is denoted as *ωt ∈* Ω, 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 *ωt* explicitly for this prediction task.

The dynamics of *{ωt}* is a stochastic process governed a transition probability *Q*. The process evolves following equation [(4.1):](#_bookmark53)

*ωt ∼ Q*(*ωt|ωt−*1*, ωt−*2*, · · · , ω*0) (4.1)

### 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 *ωt* 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 *ρ*(*ωt*). The functional form suggests the information flow received by an individual depends on both the state of world *ωt* 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

*ρ*2(*ω*) *⊆ ρ*1(*ω*) *∀ω ∈* Ω (4.2)

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 *ωt*.

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 ∈ {*1*,* 2*, · · · , N}*. For example, news article *n* is the *nth* news in the dataset. Let *τn* denote the time when the *nth* 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,

IF*t* = *{n* : *τn ∈* [*t, t* + 1)*}* (4.3)

Figure [22](#_bookmark55) summarizes the workflow of the prediction task in a minimal setting: predicting *rt*+1 using information on the day *t* only. Each of *t*, *t* + 1 and *t* + 2 denotes the beginning of the day, *pt* and *pt*+1 indicate the time when the closing price is computed. Firstly, the return on day *t* + 1 is computed from *pt*+1 and *pt* using equation [(3.2).](#_bookmark7) Then, some predictive models are used to predict *rt*+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 *rt*+1 using IF*t*, the model is not peaking into the future.

Figure 22: Workflow of the Prediction Task



*rt*+1

IF*t*

IF*t*+1

[

*t*

*pt*

*t* + 1

)[

*pt*+1 *t* )

+ 2

time

 calculating return

 predicting return

### 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 *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 */i* days in the past, *M* is a map from information in the past */i* days to a prediction:

IF*t−* +1

IF*t−* +2

IF*t*

*M* 

*rt−* +1

 *,* 

*rt−* +2

 *, · · · ,* 

*rt*

 = *r*ˆ*t*+1 (4.4)

Then the model *M* is evaluated based on how close *rt*+1 and

*r*ˆ*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 *T* be an arbitrary time period, such as the whole day *t*:

*T* := [*t, t* + 1) (4.5)

We define a characteristic function *ϕ* [4](#_bookmark0) as a mapping from a time period *T* to a real-valued vector *ϕ*(*T* ) *∈* R*d*, where *d* denotes the number of features constructed. Ideally, *ϕ*(*T* ) should provide a quantitative summary from various aspects of news articles in IF*T* .

For example, one valid characteristic function *ϕ* can construct the following two features:

1) the the number of news articles (events) in period *T* and 2) the average event sentiment

4In the context of probability, the characteristic function of a distribution fully describes the distribution, refer to (Ushakov [1999)](#_bookmark121) for a review of this topic. Here we define the characteristic function to be the function mapping a collection of news to a vector **x** of summary statistic of these news.

score of these news:

 *|{n* : *τn ∈ T }|*

 

Number of News on Day *t* 

*ϕ*(*T* ) = 1



*|{n*:*τn∈T }|*

*n s*L*.t.τn∈T*

ESS*n*

 = 

Average ESS Score of News on Day *t*

 *∈* R2 (4.6)

Note that in subsequent sections, the characteristic function actually used has far more features than the example in equation [(4.6).](#_bookmark58)

With a chosen characteristic function *ϕ*, one can summarize the information flow on each day *t* using a real-valued vector **x***t*:

**x***t* := *ϕ*([*t, t* + 1)) (4.7)

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

Using characteristic functions, the predictive model in equation [(4.4)](#_bookmark57) can be equivalently expressed as

*M* *ϕ*([*t − /i* + 1*, t − /i* + 2)) *,* *ϕ*([*t − /i* + 2*, t − /i* + 3)) *, · · · ,* *ϕ*([*t, t* + 1)) (4.8)

*rt−* +1

*rt−* +2

*rt*

= *M* (**x***t−* +1*, rt−* +1*,* **x***t−* +2*, rt−* +2*, · · · ,* **x***t, rt*) = *r*ˆ*t*+1 (4.9)

**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 *T I* = [*t − /i* + 1*, t* + 1). Then, we can construct a summary of all news arrive in the past */i* days by applying *ϕ* on *T I*:

**x***t−* +1:*t* = *ϕ*(*T I*) (4.10)

Therefore, the following alternative formulation is equally valid,

*M* (*ϕ*([*t − /i* + 1*, t* + 1))*, rt−* +1*, rr−* +2*, · · · , rt*) (4.11)

= *M* (**x***t−* +1:*t, rt−* +1*, rr−* +2*, · · · , rt*) = *r*ˆ*t*+1 (4.12)

In the following parts of this paper, we call equation [(4.9)](#_bookmark59) the **summarizing-first** for- mulation since it apply *ϕ* on news arrived in each day first, then feeds the collection of summaries to a predictive model. And, we call [(4.12)](#_bookmark60) 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 **x** of news arrive in the period of consideration. However, no matter how large */i* is, gathering- first approach only returns one copy of sentiment. In contrast, summarizing-first approach returns */i* 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.

### Inter-temporal Dependency

The state of world is changing from *ωt* to *ωt*+1 between two consecutive trading days, and *ωt*+1 depends on *ωt*. Moreover, **x***t* affects **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 *rt* impacts **x***t*+1 as well.

## Empirical Model

The empirical models to be estimated in this paper are focusing on the dynamics of sentiments *{***x***t}t* and returns *{rt}t*.

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

Specifically, in figure [21,](#_bookmark50) each pair of **x***t* and **x***t*+1 are correlation through the chain consisting of **x***t*, *ωt*, *ωt*+1, and **x***t*+1. In order to model the dynamic from **x***t* to **x***t*+1, we have to construct models estimating *P* , *E*, and *R* in the graph first.

Figure [23](#_bookmark63) illustrates the idea of forecasting returns via sentiments. Suppose we wish to predict returns on day *t*, *rt*, using only information at time *t−/i* (i.e., both *rt−* and **x***t−* ) for some integer */i >* 1. The ACF and PACF of return series have only a few significant lags, so that the arrow from *rt−* to *rt* is blocked by this weak correlation. This prevents one from forecasting *rt* using *rt−* 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 Sentiment*t−* . Secondly, the correlation between sentiment and return enables the model to estimate *rt* from the prediction of Sentiment*t*. The composite of two steps above provides an indirect approach of forecasting *rt* using information at time *t − /i*.

Figure 23: Framework

blocked

*rt*

Sentiment*t−*

*rt−*

Literature have been using hidden Markov models (HMMs) for time series forecasting. mentioned before, **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, *rt*. Therefore, the proposed framework extends the hidden Markov framework by allowing the directed edge from *rt* to **x***t*+1, the edge *R*, to explicitly model the impact of historical price movements on future news sentiment.

As Literature

here

Sentiment*t*

We model *{***x***t}t* as a stochastic process whose dynamics is governed by the **transition**

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

**x***t* = **x***A* + **x***B* + *εt* (4.13)

*t t*

where **x***A ∼ P* (**x***A|***x***t−*1) (4.14)

*t*

*t*

**x***B ∼ R*(**x***A|rt−*1) (4.15)

*t*

*t*

The transition probability *P* models the impact of past news sentiments on the future news

sentiment, and **x***A* is the portion of sentiment **x***t* solely determined by the inter-temporal

*t*

correlation among **x**. In addition, another reporting probability *R* models the impact of

past returns on future news sentiment. Hence, **x***B* is the part of **x***t* responses to past price

*t*

movement. For example, **x***B*

*t*

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*:

**x***t ∼ PR*(**x***t|***x***t−*1*, rt−*1) (4.16)

Expanding equation [(4.16)](#_bookmark64) recursively shows that **x***t* is in fact impacted by all historical values of **x***t* and *rt*. Therefore, the entire history of *{***x***t−*1*,* **x***t−*2*, · · · ,* **x**0*}* and *{rt−*1*, rt−*2*, · · · , r*0*}* contribute to the distribution of **x***t*

**x***t ∼ F* (**x***t|***x***t−*1*,* **x***t−*2*, · · · ,* **x**0*, rt−*1*, rt−*2*, · · · , r*0) (4.17)

The **order** of a Markov model determines the length of its memory, a Markov model has order */i* if the distribution of an arbitrary random variable *Yt* only depends on the past */i* values, that is, for every *t > /i*,

*P* (*Yt|Yt−*1*, Yt−*2*, · · · , Y*0) = *P* (*Yt|Yt−*1*, Yt−*2*, · · · , Yt−* ) (4.18)

In most cases, the impact of observations in the distant past, say **x***t−*1*,*000, on the current observation **x***t* is negligible. Therefore, for simplicity, we assume the chain in equation [(4.17)](#_bookmark65)

is assumed to have a finite order */i*. Hence,

**x***t ∼ F* (**x***t|***x***t−*1*,* **x***t−*2*, · · · ,* **x***t− , rt−*1*, rt−*2*, · · · , rt−* ) (4.19) As mentioned before, the actual return *rt* is determined by multiple factors. Firstly,

all those events happening on day *t*, that is, *ωt* affects *rt*. Moreover, traders are often time response to news published, and traders’ behaviours affects the market and therefore *rt*. The distribution of *rt* follows an **emission probability** distribution *E*. In particular, *E* maps the current state of world *ωt* and current news sentiment **x***t* to a distribution of realized returns *rt*. Hence, the distribution of *rt* depends on both the true state of world and the news sentiment.

*rt ∼ E*(*r|ωt,* **x***t, rt−*1*, rt−*2*, · · · , rt−* ) (4.20)

For generality, even not shown in figure [21,](#_bookmark50) we assume the distribution of *rt* depends on his- torical returns too. Therefore, the distribution in equation [(4.20)](#_bookmark66) 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 */i*, so that historical return values before *r − /i* are discarded.

Note that the series of *{ωt}* is latent, the model has only observations on *{***x***t}* and *{rt}*.

For each day *t*, we can now construct two predictors of return, *rt*,

*r*ˆraw = E[*rt|rt−*1*, rt−*2*, · · · , rt−* ] (4.21)

*t*

= *M*raw(*rt−*1*, rt−*2*, · · · , rt−* ) (4.22)

*r*ˆsenti = E[*rt|***x***t−*1*,* **x***t−*2*, · · · ,* **x***t− , rt−*1*, rt−*2*, · · · , rt−* ] (4.23)

*t*

= *M*senti(**x***t−*1*,* **x***t−*2*, · · · ,* **x***t− , rt−*1*, rt−*2*, · · · , rt−* ) (4.24)

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

Using this framework, we may formulate the original question of interest as whether

*r*ˆsenti

*t*

is a significantly better prediction of *rt*+1 than *r*ˆraw, in terms of prediction accuracy.

Recall the workflow in figure [22,](#_bookmark55) since the closing price *pt* is realized near the end of day and the return *rt* can be computed at the same time. Based on figure [13,](#_bookmark34) most information within IF*t* will arrive before *rt* is realized, if the market is efficient, a great portion of IF*t* should have already been reflected in *rt*. In this case, IF*t*, and therefore **x***t*, will not provide meaningful information to the prediction of *rt*. 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 *r*ˆsenti outperformed *r*ˆraw a lot, then we can claim that the news sentiment series

*t*

*t t*

is capable of improving return predictions.

In experiment section of this paper, this paper uses different types of statistical learning models to estimate *r*ˆraw and *r*ˆsenti, then compare their performances.

*t t*

# Experiments

## Procedures

The empirical model in section [4.3](#_bookmark62) gives a brief description on how to assess the predictive power of models with and without sentiment dataset. We need to choose a specific char- acteristic function *ϕ* and predictive model *M* in order to generate quantitive metrics and compare performances.

### 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 */i ∈* Z+ denote the length of model’s memory. That is, while predicting *rt*, the model can only use information from day *t − /i* to day *t −* 1. This paper chooses */i* = 31 so that the model uses information within a whole month to predict one return.

Firstly, for each day from day *t − /i* to day *t −* 1, the characteristic function computes all variables in table [11.](#_bookmark70) 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

ESS MEAN ESS TOTAL NUM EVENTS

Variable

Average ESS Sum of ESS

Number of Events

Code Name Variable

WESS MEAN Average WESS

WESS TOTAL Sum of WESS

Moreover, following the methodology in section [3.5,](#_bookmark37) 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

NUM POSITIVE ESS NUM NEGATIVE ESS

Variable

# news s.t. ESS *> r*

# news s.t. ESS *< −r*

Code Name

NUM POSITIVE WESS NUM NEGATIVE WESS

Variable

# news s.t. WESS *> r*

# news s.t. WESS *< −r*

NUM NEUTRAL ESS # news s.t. ESS *∈* [*−r, r*] NUM NEUTRAL WESS # news s.t. WESS *∈* [*−r, r*]

Table [11](#_bookmark70) and table [12](#_bookmark71) together provide the daily summary for the sentiment dataset from day *t − /i* to day *t −* 1, and concatenating them gives 11*/i* features in total. We denote the characteristic function computing daily summary as *ϕ*daily, for a given information flow on day *t*, *ϕ*daily(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).](#_bookmark119) 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 11*/i* to 22*/i*), and can potentially lead to the curse of dimensionality especially when */i* is large (Friedman [1997).](#_bookmark110) 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 consider- ation, IF[*t− ,t*) (i.e., all news from day *t − /i* to day *t −* 1), instead of creating */i* copies of daily summaries for */i* days, the number of features constructed no longer depends on */i*. 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 *ϕ*aggregate denote the second characteristic function extracting features from IF[*t− ,t*). Table [13](#_bookmark72) enumerates 8 types of summary statistics used. Firstly, *ϕ*aggregate computes the the 8 statistics in table [13](#_bookmark72) for ESS and WESS of all news in IF[*t− ,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](#_bookmark72) 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 [(5.1)](#_bookmark73)

so that their signs are preserved.

ESS2 := sign(ESS) *×* ESS2 (5.1) WESS2 := sign(WESS) *×* WESS2

Afterwards, we split IF into the positive group, IF+

[*t− ,t*)

[*t− ,t*)

, and the negative group, IF*−*[*t ,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 IF[*t− ,t*) based on ESS and WESS always gives the same outcome. Then, *ϕ*aggregate summarizes

*−*

*−*

the number of news, the average ESS and the average WESS for each of IF+

[*t− ,t*)

and IF*−*[*t ,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 IF++

[*t− ,t*)

(extremely positive news) to be the subset of IF[*t− ,t*)

with ESS strictly greater than 18, and IF*−−*

[*t*

*− ,t*)

(extremely negative news) to be these news

with ESS strictly less than -15. Afterwards, *ϕ*aggregate summarizes the number of extremely

positive (negative), the average ESS and the average WESS of news in IF++

[*t− ,t*)

and IF*−*[*t − ,t*)

(6 features in total).

*−*

Overall, *ϕ*aggregate constructs 44 features to summarize each information flow IF[*t− ,t*). Altogether with the 11*/i* features from *ϕ*daily, 11*/i* + 44 features are used to predict the return *rt*. For example, there would be 385 features if one chose */i* to be 31. Given there are around 5,000 daily returns to train the model, using approximate 400 independent variable is reasonable.

### Rolling-Window Method

The second step of the pipeline is to construct a batch of feature-target pairs (called a sample), (*Xt, rt*), so that we can evaluate model *M* based on how close *M*(*Xt*) and *rt* are. Let */i* = 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.](#_bookmark75)

Figure 24: Using Rolling Window to Construct (*Xt, rt*) pairs

rolling windo*r*w *t*

*Xt t*



*ϕ*daily *ϕ*aggregate

*day*

*t − fi, t*

*− fi,* + 1

*t −* 1*t t* + 1





features target

*T*

 rolling window *t* + 1 (*Xt*+1*, rt*+1)

 rolling window *t* + 2 (*Xt*+2*, rt*+2)

 rolling window *t* + 3 (*Xt*+3*, rt*+3)

. . .



rolling window *T* (*XT , rT* )

For each day *t* with valid return *rt* (those days with missing returns are discarded), the set of features *Xt* consists of 31 daily summary from *ϕ*daily, one aggregate summary from *ϕ*aggregate and 31 lagged values of returns.

*X* =31 := {*ϕ*daily(IF*t−*31)*, · · · , ϕ*daily(IF*t−*1)*, ϕ*aggregate(IF[*t−*31*,t*))*, rt−*31*, · · · , rt−*1} (5.2)

*t*

Therefore, *X*=31 consists of 416 real-valued features used to predict *rt*. Afterwards, the

*t*

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 (*Xt*+1*, rt*+1).

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

*D*train := *{*(*Xt, rt*) : *t ≤* December 31, 2018*}* (5.3)

*D*test := *{*(*Xt, rt*) : *t ≥* January 1, 2019*}* (5.4)

After assessing models’ performances on the test set, *D*train and *D*test are further split into 5

subsets [5](#_bookmark0) to explore the effectiveness of models on each day of the week.

### Performance Metrics

In order to quantify the performance of a predictive model, we have to specify a perfor- mance metric measuring the proximity between predictions and actual values. Let *r*ˆ*t* denote the predicted value of *rt*, the performance metric should reflect the proximity between pre- dicted 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*, · · · , rT }* is defined as

2

*T*

L1

*MSE* := (*rt*

*T*

*t*=1

*− r*ˆ*t*) (5.5)

One advantage of MSE metric is that it is differentiable with respect to each *r*ˆ*t*, this dif- ferentiability allows us to train models on this dataset using back-propagation algorithm (Hecht-Nielsen [1989).](#_bookmark112) 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 pri- mary 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* := 1 L 1*{*sign(*r* ) = sign(*r*ˆ )*}* (5.6)

*t*=1

*T*

*T*

1

*T*

*t*

*t*

1

*t*

*t*

1 L 1 *r − r*ˆ 1

*t*=1

*r*

*t*

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.

*MAPE* :=

*T*

(5.7)

5(*Xt, rt*) are split based on which day of the week *t* is. There are only 5 groups since *rt* are always missing when *t* is weekend.

### 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), *θ ∈* Θ. In subsequent discussions, we use subscript *Mθ* to denote a model with configuration *θ* from class *M*. For example, suppose *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 *M∗θ* 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).](#_bookmark107)

For each predictor class *M* and corresponding space of hyper-parameters Θ, we firstly determine the optimal hyper-parameter *θ∗ ∈* Θ using a 5-fold randomized cross validation algorithm (5-fold RCV).

Figure 25: 5-fold Randomized Cross Validation

All Data with *t* from Jan. 1, 2000 to Sep. 30, 2019

(before Jan. 1, 2019)

training

*D*

test

*D* (after Jan. 1, 2019)

Fold 1

Fold 2

Fold 3

Fold 4

Fold 5

Fold 1

Fold 2

Fold 3

Fold 4

Fold 5

Fold 1

Fold 2

Fold 3

Fold 4

Fold 5

Fold 1

Fold 2

Fold 3

Fold 4

Fold 5

Fold 1

Fold 2

Fold 3

Fold 4

Fold 5

Fold 1

Fold 2

Fold 3

Fold 4

Fold 5

Final Evaluation

test

*D* (after Jan. 1, 2019)

*MSE*1 *MSE*2 *MSE*3 *MSE*4 *MSE*5

*MSEθn*

Finding *θ∗*

Firstly, *N* candidates of hyper-parameters *{θ*1*, θ*2*, · · · , θN }* are sampled from a uniform distribution on Θ (i.e., the randomized part in RCV). Then, for each *θn ∈ {θ*1*, θ*2*, · · · , θN }*,

*θn* defines a predictive model *Mθn* Figure [25](#_bookmark78) illustrates the cross validation procedure for one hyper-parameter set *θn*. The entire *D*train are split into 5 equal consecutive subsets (called

folds): *D*train for *i ∈ {*1*,* 2*,* 3*,* 4*,* 5*}*. Then, *Mθ* is fitted on *∪i∈{*1*,*2*,*3*,*4*}D*train (training set) and

*i*

*i*

*n*

evaluated on *D*train (validation set), let *M*-*SE*5 denote mean squared error of this model on

5

*D*train. Afterwards, the same model is fitted again on *∪i∈{*1*,*2*,*3*,*5*}D*train, evaluated on *D*train

5

*i*

4

and leads to another error metric *M*-*SE*4. The same procedure can be repeated for five times with different validation set and creates five mean squared error metrics. Let *MSEθn* denote the average of *M*-*SE*1 to *M*-*SE*5 using hyper-parameter *θn*. Then, *MSEθ* constitutes an estimated test-time performance of model *Mθ* on *D*test when it is fitted on *D*train. Among the *N* candidates of hyper-parameters for model class *M*, we choose *θn* with the smallest *MSEθn* 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.

*n*

*n*

While using 5-fold RCV and *N* sampled *θ*, we need for fit 5*N* models from class *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.

Since this paper works with two performance metrics: mean-squared-error (MSE) and directional accuracy (DA), there are at least two criterions to identify the optimal model.

**MSE-Optimal** The first selection criterion identifies the optimal model from a given model class based on models’ validation MSEs. Given a class of models, one model is the MSE- optimal of this model class if

* + - * this model achieves at least a DA of 50% on the validation set;
      * and, it achieves the lowest MSE on the validation set among all models satisfy the first condition.

If there are more than one models with the same lowest validation MSE and DA greater than 50%, we will choose the one with the best DA to be the optimal model.

**DA-Optimal** The second criterion selects the best model from candidates based on their validation DAs. Given a class of models, one model is the DA-optimal of this model class if

* + - * this model achieves the highest DA on the validation set.

If there are multiple models with the same lowest accuracy, the model with the lowest validation MSE is chosen to be the DA-optimal.

MSE

Let *M*

model class

DA

model class

and *M*

denote the MSE-optimal and DA-optimal models from a

particular model class. After identifying optimal models, the performances of *M*MSE

model class

DA

and *M*

model class

on the test set will be proxies of the performance of the specified model class

on the prediction task. Note that these optimal models can be identify completely without the test set, therefore, evaluating them on the test set mimics real-world environment: some- one builds and trains a model at time *t* using all information available up to time *t*, then the model is implemented on company’s server and used in production after day *t*. What traders really care about is the predictive model’s performance after day *t*, called test time performance, since the test time performance is directly related to the profitability of any trading algorithms built on this predictive model. Therefore, one model’s performance on the test set is a fair proxy for the business value of this model in real-world.

## Baseline Models: The Naive Predictor and Moving Average Predictors

To better evaluate model performances, we firstly define several dummy models for bench- marking purpose. The simplest model is a naive predictor, *M*naive, predicting zero returns all the time. In addition, we define other moving-average predictors denoted as *M*MA(*k*), where *k* is a positive integer representing the scope of this model. To predict *rt*, *M*MA(*k*) looks into the past *k* days and predicts the return to be the average return over the last *k* trading days.

0

1

2

*r*ˆ = *M*

*t*

MA(*k*)

(*r , r , r , · · · , r*

1 *t−*1

) = *r*

L

*t−*1

*k*

*t*

(5.8)

*τ* =*t−k*

As mentioned before, this paper uses all data before December 31, 2018 as the training set and data after January 1, 2019 as the test set. Table [14](#_bookmark80) shows performances of benchmark models in terms of mean-squared-error (MSE) and directional accuracy (DA).

Table 14: Performances of Benchmark Models

Model

*M*naive *M*MA(5) *M*MA(25) *M*MA(50) *M*MA(100) *M*MA(300)

Training MSE

4.655

5.612

4.811

4.725

4.706

4.676

Training DA Testing MSE Testing DA

0.716% 4.057 0.538%

50.274% 4.693 50.000%

50.295% 4.248 50.000%

49.536% 4.261 50.000%

49.241% 4.226 44.624%

47.977% 4.060 48.925%

Because returns are rarely exactly zero in the dataset, the directional accuracy of *M*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 results suggest that the crude oil market is efficient (i.e., unpredictable) with respect to

* the information set containing historical returns,
* and naive predictors as well as moving-average predictors.

Then we are going to examine whether other more sophisticated models can attain signifi- cantly better test time performances than those above-mentioned benchmark models.

## Linear Models: Autoregressive Integrated Moving Average

One classical model used for time series forecasting is the autoregressive integrated moving average (ARIMA) model.

The autoregressive moving average (ARMA) models the return at time step *t*, *rt*, as a function the series itself and a series of error terms. In particular, an ARMA(*p, q*) process identifies *rt* as a linear combination of *p* lagged variation of *rt* and *q* noise terms:

*Xt − φ*1*Xt−*1 *− · · · − φpXt−p* = *εt* + *θ*1*εt−*1 + *· · ·* + *θqεt−q* (5.9)

More concisely, let *L* denote the lag operator, an ARMA(*p*,*q*) process can be written with the more compact notion

Φ*p*(*L*)*rt* = Θ*q*(*L*)*εt* (5.10)

where Φ*p*(*·*) and Θ*q*(*·*) are polynomials up to degree *p* and *q* respectively.

One crucial assumption on *{rt}* is that it has to be stationary. To handle non-stationary processes, one can apply the differencing operator, 1 *− L*, iteratively until the differenced series becomes stationary. If the stationarity is achieved after *d* iterations of differencing, the original series is said to be integrated of order *d* and can be modelled using an ARIMA(*p, d, q*) model:

Φ*p*(*L*)(1 *− L*)*dXt* = Θ*q*(*L*)*εt* (5.11)

ARIMA models the inter-temporal dependencies naturally, however, ARIMA can only handle linear relationships. In later sections, non-linear models such as support vector machines will be introduced.

This paper searches over 72 configurations of ARIMA models, *M*ARIMA(*p,d,q*), with differ- ent combinations of (*p, d, q*) specified in table [15.](#_bookmark82) This paper uses the Akaike’s information criterion (AIC) of ARIMA models on the training set to identify the optimal configuration. ARIMA(5,0,5), ARIMA(4,0,3) and ARIMA(5,0,4) are the three models attain the three low- est AIC values on training set. Table [16](#_bookmark83) summarizes the performances of these three optimal models identified.

Table 15: Scope of Orders for ARIMA

Parameter Scope

*p {*0,1,2,3,4,5*}*

*d*

*q {*0,1,2,3,4,5*}*

*{*0,1,2*}*

Table 16: Performances of Linear Models

Model

*M*ARIMA(5*,*0*,*5)

*M*ARIMA(4*,*0*,*3)

*M*ARIMA(5*,*0*,*4)

Testing MSE

4.074

4.070

4.073

Testing DA

50.763 %

51.156 %

50.567 %

It turns out that ARIMA models perform poorly and merely out-performed benchmark models. Therefore, we may extend our previous conclusion: the crude oil market is efficient with respect to

* the information set containing historical returns,
* and naive predictors, moving-average predictors and ARIMA models.

Since ARIMA models work univariate time series, we can only test the market’s efficiency on the information set without news sentiment.

## Support Vector Regression

The previous section shows that the crude oil market is efficient with respect to linear models. In contrast, this section is devoted to non-linear models and answers whether the market is efficient against non-linear models.

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

As mentioned in section [5.1.1,](#_bookmark69) characteristic functions generate 416 predictors in total for each one target *rt*, so that the prediction task is in fact a high dimensional problem. SVM models only focus a few training samples termed *support vectors*, therefore, SVMs often produce promising results on high dimensional problems.

Moreover, by using different *kernel functions*, SVMs are capable of transforming these raw features to an even higher dimensional space. For example, if one wishes to classify points in R2, other algorithms like logistic regressions would classify the point based on (*x*1*, x*) directly and maps (*x*1*, x*2) to class labels. Instead, a SVM with polynomial kernel of degree two will classify the point based all combinations of (*x*1*, x*2) up to degree two, that is, (*x*1*, x*2*, x*2*, x*1*x*2*, x*2) *∈* R5. In this case, the original 2-dimensional input space is

1

2

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 engineering 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).](#_bookmark109) As the name suggests, SVR is designed for regression problems. It has been shown that SVRs perform reasonably well on high dimensional regression problems by focusing on a few support vectors and engineering features implicitly using kernel functions.

As mentioned in Smola and Scholkorf’s work, performances of support vector machines are determined by several hyper-parameters listed in table [17.](#_bookmark85) To choose the optimal config- uration of SVR in this paper’s prediction task, a RCV algorithm samples 500 configurations from the scope of hyper-parameters in table [17](#_bookmark85) 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

Kernel Type

*γ* Tolerance *ε*

*C*

Scope

*{*Radial Basis Function (RBF) kernel*}*

*{*10*−*10*,* 10*−*9*,* 10*−*8*,* 10*−*7*,* 10*−*6*,* 10*−*5*,* 10*−*4*,* 10*−*3*,* 10*−*2*,* 0*.*1*,* 1*,* 10*}*

*{*10*−*10*,* 10*−*9*,* 10*−*8*,* 10*−*7*,* 10*−*6*,* 10*−*5*,* 10*−*4*,* 10*−*3*,* 10*−*2*,* 0*.*1*,* 1*,* 10*}*

*{*10*−*10*,* 10*−*9*,* 10*−*8*,* 10*−*7*,* 10*−*6*,* 10*−*5*,* 10*−*4*,* 10*−*3*,* 10*−*2*,* 0*.*1*,* 1*,* 10*}*

*{*10*−*10*,* 10*−*9*,* 10*−*8*,* 10*−*7*,* 10*−*6*,* 10*−*5*,* 10*−*4*,* 10*−*3*,* 10*−*2*,* 0*.*1*,* 1*,* 10*}*

Table [18](#_bookmark86) and table [19](#_bookmark87) presents the optimal models under both criterions and their respec- tive performances.

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

Model

MSE

Kernel *γ*

Tolerance *ε*

*M*

*M*

SVR DA SVR

RBF

RBF

10*−*6

10*−*7

1

10*−*3

10*−*5 10*−*7

*C*

10*−*9

10

Table 19: Performances of Support Vector Regression Machines

*M*

*M*

Model

MSE

SVR DA SVR

Validation MSE Validation DA Testing MSE Testing DA

4.655 51.433 % 4.055 54.301 %

4.830 51.665 % 4.399 49.462 %

The MSE-optimal SVR, *M*MSE, out-performs the naive predictor and all linear models in terms of both test-time MSE and DA. In contrast, *M*MSE only attains unsatisfactory results

DA

SVR

compared with our benchmarks.

Given *M*MSE’s superior performance, we can conclude the crude oil market is inefficient with respect to

SVR

* (information set) the information set consists of both historical returns and news sen- timents,
* (predictive models) support vector regression machines;
* (searching technology) randomized cross validation and select the optimal model based on validation MSE.

## Random Forests

Another class of non-linear methods used widely is the random forest. Breiman proposed an ensemble model based on traditional tree methods called random forest (2001). A forest as an ensemble of independently trained trees reduces the variance of prediction and achieves a better performance compared with one single tree predictor. Let *p* denote the number of independent variables for prediction (*p* = 416 here). Training each tree in the forest using all *p* features can cause over-fitting problems and lead to poor test time performance. Therefore, each independent tree predictor in the forest is trained only using a random subset of *p* features, which constitutes the randomness of a random forest. This randomness on feature selection helps random forests to generalize better and achieves even lower loss on the test set. Typically, each tree in the forest is only trained on log2(*p*) features.

Table [20](#_bookmark89) enumerates key hyper-parameters of a random forest predictor and correspond-

ing scopes our cross validation procedure searches over.

Table 20: Scope of Hyper-parameters for Random Forests

Hyper-parameter

*n* Number of trees

*f* Max num. of features for each tree

*d* Max depth of each tree

*m*1 Min amount of samples required to split an internal node

*m*2 Minimum number of samples required

at each leaf node What dataset is used

to construct each tree

Scope

*{*1*,* 2*,* 3*, · · · ,* 200*}*

*{p,* log2(*p*)*}*

*{*10*,* 14*,* 19*,* 24*, · · · ,* 100*,* 105*,* 110*, ∞}*

*{*2*,* 5*,* 10*}*

*{*1*,* 2*,* 4*}*

*{*bootstrapped samples, entire training dataset*}*

Table [20](#_bookmark89) summarizes configurations of *M*MSE and *M*DA identified.

RF

RF

Table 21: Optimal Hyper-parameters for Random Forests

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *n* | *f* | *d* | *m*1 | *m*2 | Training Samples |
| 96 | log2(*p*) | 10 | 10 | 2 | Bootstrapped Samples |
| 41 | *p* | 14 | 5 | 4 | Entire Training Set |

Model

MSE RF DA RF

*M*

*M*

After identifying optimal models, these models are evaluated using the test set, and table [21](#_bookmark90) reports the performances. Since we search over a sufficiently large scope of hyper- parameters for random forests, those test-time performances in table [21](#_bookmark90) should be close to the best a random forest can do on this return prediction problem. It turns out that the performances of random forests only achieve an unsatisfactory result in terms of testing MSE. Both optimal models perform worse than a naive predictor, which attains a testing MSE of 4.057.

Table 22: Performances of Random Forests

Model Validation MSE Validation DA Testing MSE Testing DA

*M*

MSE RF DA RF

In terms of the d guessing

*M*

on both validation and test sets. However, a market is predictable only if a model can consistently achieve better-than-guessing results. The statistics in table [22](#_bookmark91) are not sufficient

|  |  |  |  |
| --- | --- | --- | --- |
| 4.675 | 50.464 % | 4.148 | 48.387 % |
| 5.716 | 51.960 % | 4.753 | 53.226 % |
| irectional accuracy, *M*DA achieves better results than random  RF | | | |

to conclude the market is predictable. Therefore, we conclude the crude oil market to be efficient with respect to

* an information set containing both historical returns and news sentiments,
* and random forest models.

## Long Short-Term Memory Recurrent Neural Networks

An ARIMA model captures the intertemporal correlation among independent variables explicitly. However, ARIMA models are not capable of modelling complex non-linearities. The superior performance of SVRs indicates that considering non-linear interactions among independent variables can improve model performance significantly.

Unfortunately, even though SVRs and random forests are capable to model complex, they squeeze all independent variables and disregard the orders among independent variables. Hence, SVRs and random forests are not able to utilize information from the sequential structures (orders) of independent variables.

In current literature, neural networks have been used widely to capture non-linearity among independent variables. One special type of neural works termed recurrent neural networks (RNN) is designed to model both non-linearities and inter-temporal correlations. However, conventional RNNs suffer from vanishing and exploding gradient problems and becomes impotent on longer time series. In section [3.2,](#_bookmark5) the ACF and PACF plots have identified possible seasonality in crude oil returns. Modelling seasonality requires the RNN to pay attention to inter-temporal dependencies over a longer period of time.

Hochreiter and Schmidhuber proposed the long short-term memory (LSTM) cell for RNNs, this novel architecture allows RNNs to focus on inter-temporal dependencies over both short and long periods (1997). All RNNs trained and evaluated in this paper are based on this LSTM architecture.

Table [23](#_bookmark93) summarizes the scope of hyper-parameters for LSTM RNNs.

Table 23: Scope of Hyper-parameters for LSTM RNNS

Hyper-parameter

Epochs of training

*h* Size of RNN hidden layer

*f* Number of RNN hidden layers

*p*rnn Dropout probability in RNN hidden layers

*p*fc Dropout probability

in the output layer

*B* Batch size

*α* Learning rate

Scope

*{*5*,* 6*,* 7*,* 8*, · · · ,* 18*,* 19*,* 20*,* 25*,* 30*,* 35*, · · · ,* 200*}*

*{*32*,* 64*,* 128*,* 256*,* 512*,* 1024*}*

*{*1*,* 2*,* 3*}*

*{*0*,* 0*.*25*,* 0*.*5*}*

*{*0*,* 0*.*25*,* 0*.*5*}*

*{*32*,* 128*,* 512*}*

*{*10*−*5*,* 3 *×* 10*−*5*,* 10*−*4*,* 3 *×* 10*−*4*,* 10*−*3*,* 3 *×* 10*−*3*,* 0*.*01*,* 0*.*03*,* 0*.*1*,* 0*.*3*}*

Table [24](#_bookmark94) and table [25](#_bookmark95) present two optimal models and their test time performances. Be- cause LSTM RNNs are capable to capture all of non-linearities, inter-temporal dependencies over short time periods (inter-day transitions) and inter-temporal dependencies over longer

and *M*

time periods (seasonalities), both *M*MSE

LSTM

DA LSTM

out-perform other models examined

before.

Table 24: Optimal Hyper-parameters for LSTM RNNs

Model

MSE

*M*

*M*

LSTM DA LSTM

Epochs *h f*

40 32 2

12 128 2

*p*rnn

0.0

0.0

*p*fc

0.5

0.25

*B α*

512

512

10*−*5

10*−*5

Table 25: Performances of LSTM RNNs

Model

MSE

*M*

*M*

LSTM DA LSTM

Validation MSE Validation DA Testing MSE Testing DA

4.192 51.609 % 4.043 54.012 %

4.888 54.480 % 4.041 54.011 %

Statistics in table [25](#_bookmark95) suggest that LSTM RNNs consistently attain non-trivial perfor- mances on both validation and test sets. Therefore, crude oil returns are in fact predictable using LSTM RNNs and the market is concluded to be inefficient with respect to

* An information set consists of both historical returns and news sentiment,
* LSTM RNN models,
* and model selection techniques based on either MSE or DA on the validation set.

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

In section [3.3,](#_bookmark14) we have shown that crude oil returns experience 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 utilize this fact, we select and train two separate models for returns on Mondays and returns on the rest of the week.

The same procedure identifies the optimal configuration of each model class for two separate datasets:

* a dataset with Mondays only consisting of 889 training samples and 34 testing samples,
* and another dataset excluding all Mondays consisting of 3,857 training samples and 152 tests samples.

Optimal models identified for both datasets are reported in table [26,](#_bookmark97) table [27](#_bookmark98) and table [28.](#_bookmark99)

The notation for optimal models is similar, for example, *M*MSE represents the MSE- optimal random forest for predicting returns on Mondays and *M*MSE indicates the

RF, Mondays

SVR, Other Days

MSE-optimal SVR for predicting returns on Tuesdays, Wednesdays, Thursdays and Fridays.

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

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *n* | *f* | *d* | *m*1 | *m*2 | Training Samples |
| 115 | log2(*p*) | 38 | 10 | 4 | Bootstrapped Samples |
| 116 | log2(*p*) | 38 | 2 | 1 | Entire Training Set |
| 88 | log2(*p*) | 10 | 5 | 4 | Bootstrapped Samples |
| 157 | *p* | 10 | 2 | 1 | Entire Training Set |

Model

MSE

*M*

RF, Mondays DA

*M*

RF, Mondays

MSE

RF, Other Days DA

*M*

*M*

RF, Other Days

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

Model

MSE

*M*

*M*

*M M*

SVR, Mondays DA

SVR, Mondays

MSE

SVR, Other Days DA

SVR, Other Days

Kernel

RBF RBF

*γ*

10*−*10

10*−*6

0.1

10*−*9

Tolerance

10*−*3

0.1

*ε*

10*−*4

10*−*6

10*−*4

10*−*7

*C*

10

1

RBF

RBF

0.1

0.1

10

1

Table 28: Optimal Hyper-parameters for LSTM RNNs on Restricted Dataset

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Epochs | *h* | *f* | *p*rnn | *p*fc | *B* | *α* |
| 45 | 128 | 2 | 0.5 | 0.0 | 128 | 10*−*4 |
| 75 | 1024 | 3 | 0.0 | 0.25 | 128 | 0.01 |
| 14 | 512 | 3 | 0.0 | 0.5 | 32 | 0.001 |
| 12 | 512 | 3 | 0.0 | 0.0 | 32 | 0.001 |

Model

*M*

*M*

MSE

LSTM, Mondays DA

LSTM, Mondays

MSE

*M*

LSTM, Other Days DA

*M*

LSTM, Other Days

Performances of models are reported in table [29,](#_bookmark100) the best performances are in bold font. It turns out that optimal models for both datasets are LSTM RNNs. In particular,

MSE

*M*

LSTM, Mondays

achieves a superior performance in terms of the directional accuracy in the

test set. However, all three types of models used are data-intensive but there are only 889 training samples of Mondays. Therefore, none of them delivers a significantly better result compared with models previously trained using the entire dataset.

Table 29: Performances of Models on Restricted Datasets

|  |  |  |  |
| --- | --- | --- | --- |
| Validation MSE | Validation DA | Testing MSE | Testing DA |
| 0.659 | 52.984 % | 0.943 | 41.176 % |
| 0.681 | 54.887 % | 1.042 | 44.118 % |
| 0.655 | 55.574 % | 0.919 | 47.059 % |
| 0.672 | 56.461 % | **0.939** | 55.882 % |
| **0.484** | 51.479 % | 0.971 | **57.143** % |
| 0.673 | **56.819** % | 1.080 | 42.857 % |
| 5.575 | 52.605 % | 4.762 | 53.289 % |
| 5.583 | 52.631 % | 4.776 | 53.289 % |
| 5.606 | 50.997 % | 4.799 | 46.711 % |
| 6.844 | 52.449 % | 5.499 | 51.974 % |
| **4.947** | 51.255 % | **4.749** | **53.290** % |
| 5.761 | **54.836** % | 4.760 | **53.290** % |

Model

*M*MSE

SVR, Mondays DA

*M*

SVR, Mondays MSE

*M*

RF, Mondays DA

*M*

RF, Mondays MSE

*M*

LSTM, Mondays DA

*M*

LSTM, Mondays

MSE

SVR, Other Days DA

*M*

*M*

SVR, Other Days MSE

*M*

RF, Other Days DA

*M*

RF, Other Days MSE

*M*

LSTM, Other Days DA

*M*

LSTM, Other Days

# Conclusions

**References**

BAKER, MALCOLM, and JEFFREY WURGLER. 2006. “Investor Sentiment and the Cross- Section of Stock Returns”. *The Journal of Finance* 61 (4): 1645–1680. issn: 0022-1082. doi:[10.1111/j.1540-6261.2006.00885.x](http://dx.doi.org/10.1111/j.1540-6261.2006.00885.x).

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](http://dx.doi.org/10.1257/jep.30.1.139).

Bodnaruk, Andriy, Tim Loughran, and Bill McDonald. 2015. “Using 10-K Text to Gauge Financial Constraints”. *Journal of Financial and Quantitative Analysis*.

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](http://dx.doi.org/10.1145/130385.130401).

Brandt, Michael W, and Lin Gao. 2019. “Macro fundamentals or geopolitical events? A textual analysis of news events for crude oil”. *Journal of Empirical Finance* 51 (J. Finance 59 3 2004): 64–94. issn: 0927-5398. doi:[10.1016/j.jempfin.2019.01.007](http://dx.doi.org/10.1016/j.jempfin.2019.01.007).

Breiman, Leo. 2001. “Random Forests”. *Machine Learning* 45 (1): 5–32. issn: 0885-6125. doi:10.1023/a:1010933404324.

Bybee, Leland, et al. 2019. “The Structure of Economic News”. *SSRN Electronic Journal*. doi:[10.2139/ssrn.3446225](http://dx.doi.org/10.2139/ssrn.3446225).

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

Deeney, Peter, et al. 2015. “Sentiment in oil markets”. *International Review of Financial Analysis* 39:179–185. issn: 1057-5219. doi:[10.1016/j.irfa.2015.01.005](http://dx.doi.org/10.1016/j.irfa.2015.01.005).

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](http://dx.doi.org/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](http://dx.doi.org/10.1109/ijcnn.1989.118638).

Hochreiter, Sepp, and Ju¨rgen Schmidhuber. 1997. “Long Short-Term Memory”. *Neural Com- putation* 9 (8): 1735–1780. issn: 0899-7667. doi:[10.1162/neco.1997.9.8.1735](http://dx.doi.org/10.1162/neco.1997.9.8.1735).

Hodges, J L. 1958. “The significance probability of the smirnov two-sample test”. *Arkiv f¨or Matematik* 3 (5): 469–486. issn: 0004-2080. doi:[10.1007/bf02589501](http://dx.doi.org/10.1007/bf02589501).

Hu, Ziniu, et al. 2018. “Listening to Chaotic Whispers”: 261–269. doi:[10.1145/3159652.](http://dx.doi.org/10.1145/3159652.3159690)

[3159690](http://dx.doi.org/10.1145/3159652.3159690). eprint: 1712.02136.

Loughran, Tim, and Bill McDonald. 2016. “Textual Analysis in Accounting and Finance: A Survey”. *Journal of Accounting Research* 54 (4): 1187–1230. issn: 0021-8456. doi:[10.](http://dx.doi.org/10.1111/1475-679x.12123) [1111/1475-679x.12123](http://dx.doi.org/10.1111/1475-679x.12123).

— . 2011. “When Is a Liability Not a Liability? Textual Analysis, Dictionaries, and 10- Ks”. *The Journal of Finance* 66 (1): 35–65. issn: 1540-6261. doi:[10 . 1111 / j . 1540 - 6261.2010.01625.x](http://dx.doi.org/10.1111/j.1540-6261.2010.01625.x).

Mohammadi, Hassan, and Lixian Su. 2010. “International evidence on crude oil price dy- namics: Applications of ARIMA-GARCH models”. *Energy Economics* 32 (5): 1001–1008. issn: 0140-9883. doi:[10.1016/j.eneco.2010.04.009](http://dx.doi.org/10.1016/j.eneco.2010.04.009).

Mudinas, Andrius, Dell Zhang, and Mark Levene. 2019. “Market Trend Prediction using Sentiment Analysis: Lessons Learned and Paths Forward”. eprint: 1903.05440.

Pennington, Jeffrey, Richard Socher, and Christopher Manning. 2014. “Glove: Global Vectors for Word Representation”. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*: 1532–1543. doi:[10.3115/v1/d14-1162](http://dx.doi.org/10.3115/v1/d14-1162).

Roache, Shaun K, and Marco Rossi. 2010. “The effects of economic news on commodity prices”. *The Quarterly Review of Economics and Finance* 50 (3): 377–385. issn: 1062-

9769. doi:[10.1016/j.qref.2010.02.007](http://dx.doi.org/10.1016/j.qref.2010.02.007).

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](http://dx.doi.org/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 Sch¨olkopf. 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.

Tetlock, Paul C. 2007. “Giving Content to Investor Sentiment: The Role of Media in the Stock Market”. *The Journal of Finance* 62 (3): 1139–1168. issn: 0022-1082. doi:[10.1111/](http://dx.doi.org/10.1111/j.1540-6261.2007.01232.x) [j.1540-6261.2007.01232.x](http://dx.doi.org/10.1111/j.1540-6261.2007.01232.x).

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

# Appendix

Figure 26:

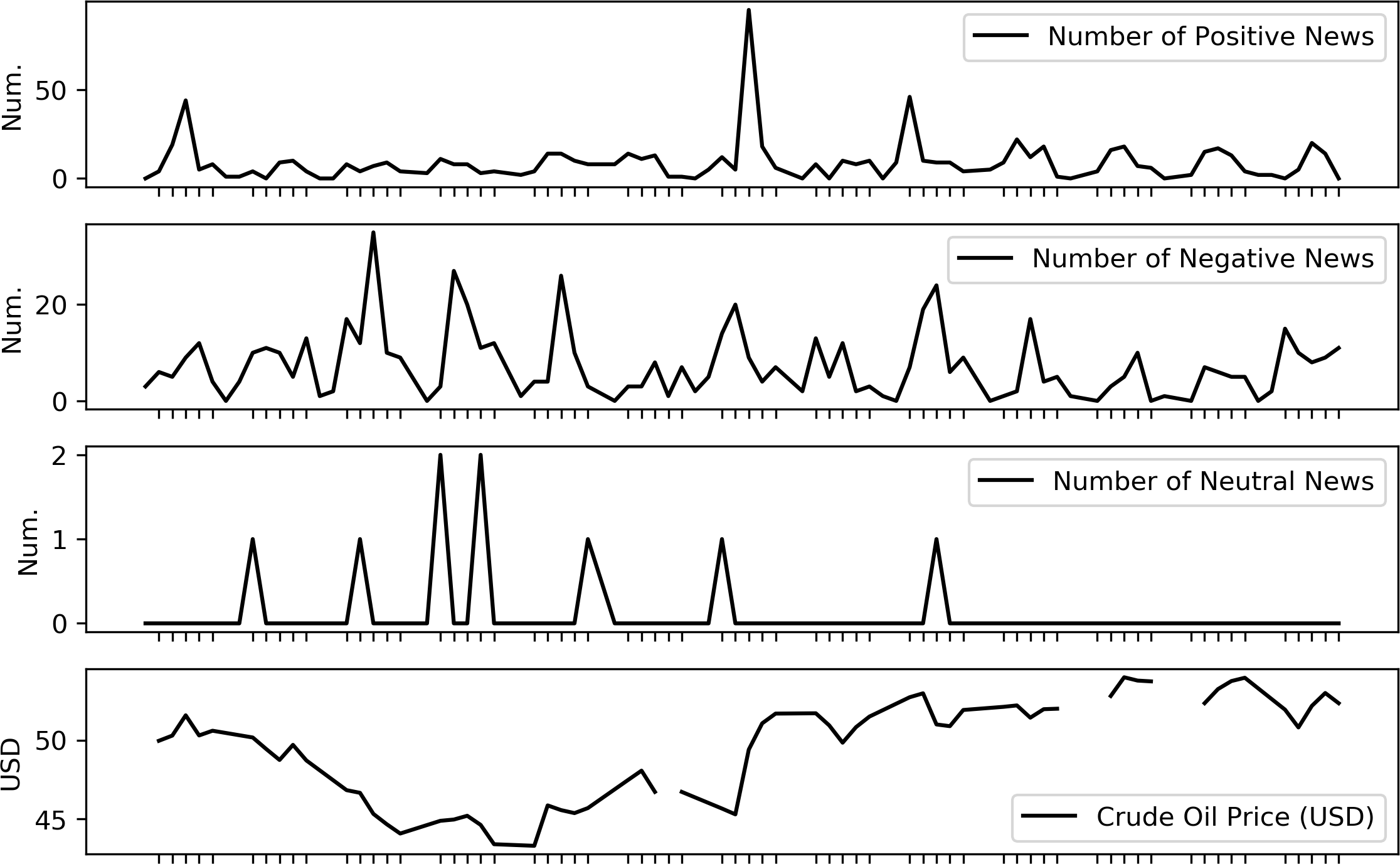


Figure 27:

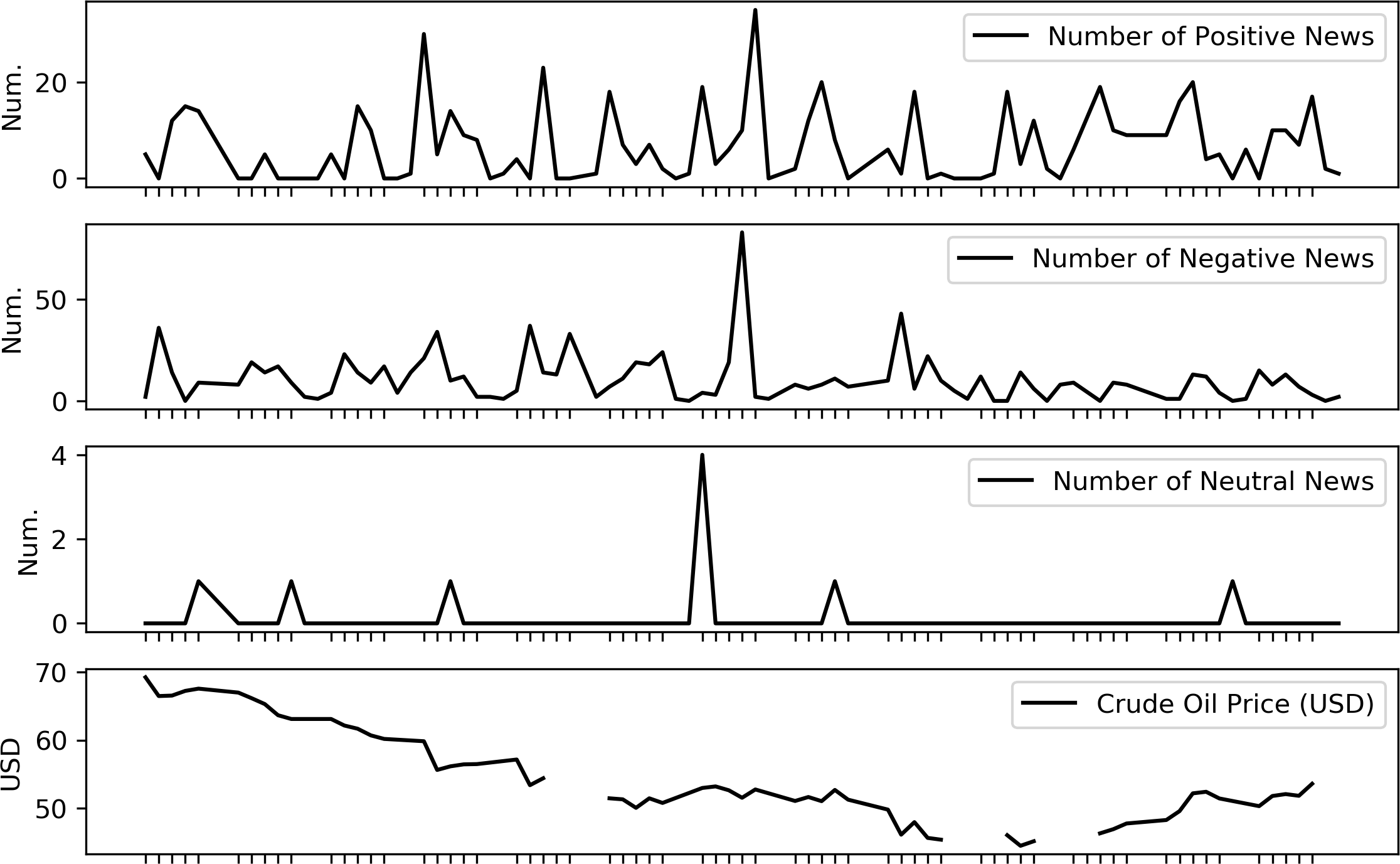


Figure 28:

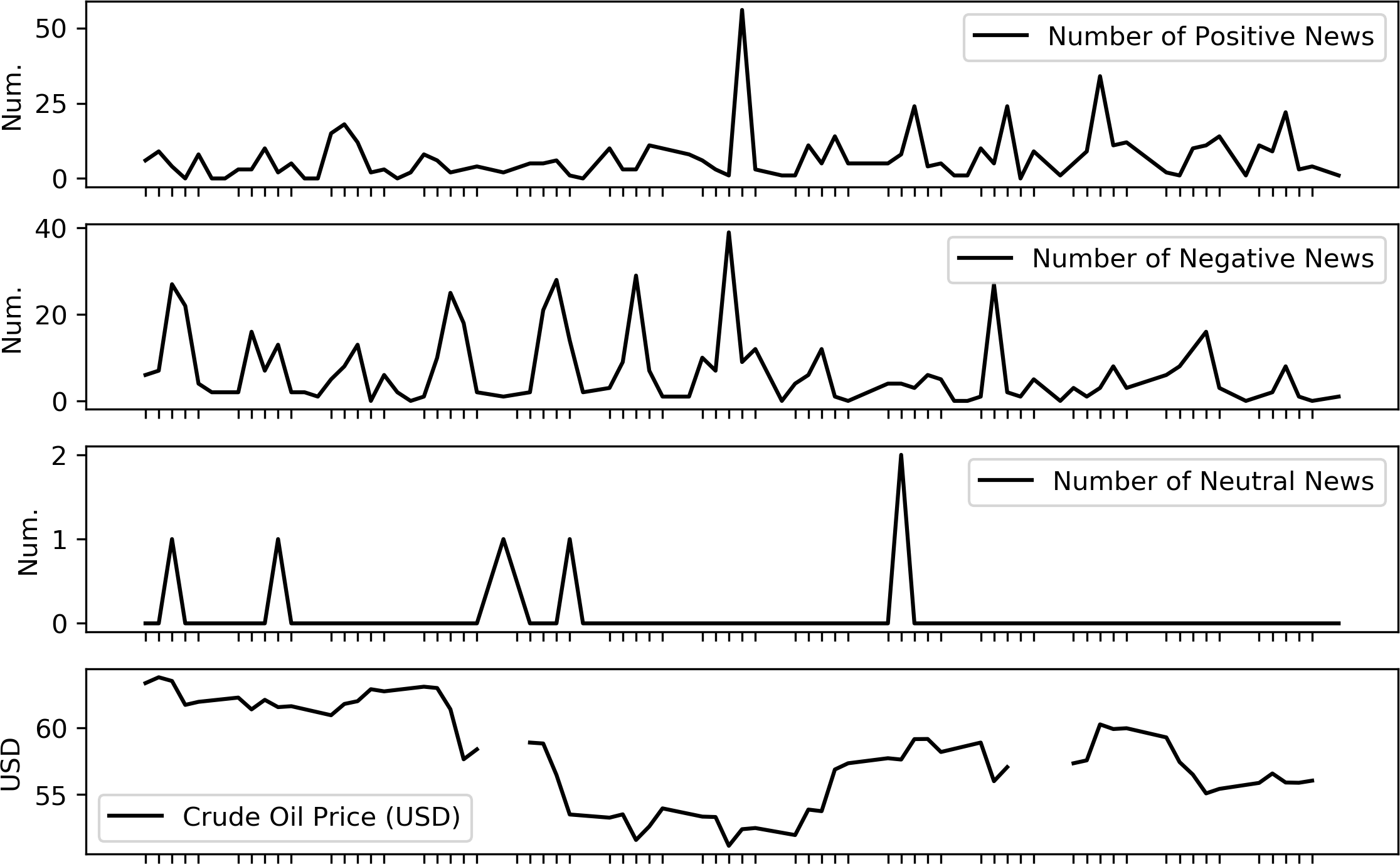


Table 30: All Categories of Positive News

|  |  |
| --- | --- |
| 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 31: 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 32: 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 33: 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%) |