

Automated News Analysis

by

Julia Sarayeva

A Thesis Submitted to the Graduate
Faculty of Rensselaer Polytechnic Institute

in Partial Fulfillment of the
Requirements for the degree of

Master of Science

Major Subject: Computer Science

Approved:

Malik Magdon-Ismail, Thesis Adviser

Rensselaer Polytechnic Institute
Troy, New York

December, 2009

CONTENTS

Automated News Analysis	i
LIST OF TABLES.....	iv
LIST OF FIGURES	v
ABSTRACT	ix
1. Introduction.....	1
2. Data and General Description of the Analysis	2
2.1 Data	2
2.2 General Description of Data Processing	3
2.3 Preliminary Preprocessing	4
2.4 Word Usage.....	6
3. Building Informative Indicators	7
3.1 Informativeness	7
3.2 Informativeness and Short Term Moves of S&P 500	8
3.3 In-sample Performance of the Informative Indicators	8
3.4 Out-of-sample Performance of the Informative Indicators	13
4. Building Financial Indicators	19
4.1 Financial Words	19
4.2 General Description of the Financial Weights Assignment.....	19
4.3 Financial Indicators.....	20
4.3.1 In-sample Testing of the Financial Indicators.....	20
4.3.2 Out-of-sample Testing of the Financial Indicators	23
5. High Level Based Indicators	26
5.1 Financial Weights of Topic Codes and Keywords.....	26
5.2 In-sample Testing of the High Level Based Indicators.....	26
5.3 Out-of-sample Testing of the High Level Based Indicators	30
6. Predicting Short Term Intraday Volatility.....	35

6.1	General Description of the Sliding Windows' Processing.....	35
6.2	Out-of-sample Testing for Each Sliding Window	36
6.3	Out-of-sample Testing for all Windows' Data Combined.....	44
6.3.1	Additional Experiments Using 25% of the Words.....	46
7.	Discussions and Conclusion	48
	LITERATURE CITED	49
	APPENDIX A Processing of Corrected (Updated) News.....	50
	APPENDIX B Keywords Detection and Text Conversion	52
B.1	Keywords Detection.....	52
B.2	Text Conversion to One Paragraph.....	54
	APPENDIX C Case Modification	55
	APPENDIX D Similarity.....	57
	APPENDIX E Number of Data Points	59

LIST OF TABLES

LIST OF FIGURES

Figure 2.1 Number of words used in Reuters news in 2006.....	6
Figure 3.1 Average absolute returns of S&P vs. $InformS_1(news)$, in-sample testing of news of 2003-2004	9
Figure 3.2 Average absolute returns of S&P vs. $InformS_1(news)$, in-sample testing of news of 2006-2007	9
Figure 3.3 Average absolute returns of S&P vs. $InformS_2(news)$, in-sample testing of news of 2003-2004	10
Figure 3.4 Average absolute returns of S&P vs. $InformS_2(news)$, in-sample testing of news of 2006-2007	10
Figure 3.5 Average absolute returns of S&P vs. $InformS_3(news)$, in-sample testing of news of 2003-2004	11
Figure 3.6 Average absolute returns of S&P vs. $InformS_3(news)$, in-sample testing of news of 2006-2007	12
Figure 3.7 Average absolute returns of S&P vs. $InformS_4(news)$, in-sample testing of news of 2003-2004	12
Figure 3.8 Average absolute returns of S&P vs. $InformS_4(news)$, in-sample testing of news of 2006-2007	13
Figure 3.9 Average absolute returns of S&P vs. $InformS_1(news)$, out-of-sample testing of news of 2005.....	14
Figure 3.10 Average absolute returns of S&P vs. $InformS_1(news)$, out-of-sample testing of news of 2008	14
Figure 3.11 Average absolute returns of S&P vs. $InformS_2(news)$, out-of-sample testing of news of 2005	15
Figure 3.12 Average absolute returns of S&P vs. $InformS_2(news)$, out-of-sample testing of news of 2008	15
Figure 3.13 Average absolute returns of S&P vs. $InformS_3(news)$, out-of-sample testing of news of 2005	16
Figure 3.14 Average absolute returns of S&P vs. $InformS_3(news)$, out-of-sample testing of news of 2008	17

Figure 3.15 Average absolute returns of S&P vs. $InformS_4(news)$, out-of-sample testing of news of 2005	18
Figure 3.16 Average absolute returns of S&P vs. $InformS_4(news)$, out-of-sample testing of news of 2008	18
Figure 4.1 Average absolute returns of S&P vs. $FinWeight_1(news)$, in-sample testing of news of 2003-2004	21
Figure 4.2 Average absolute returns of S&P vs. $FinWeight_2(news)$, in-sample testing of news of 2003-2004	21
Figure 4.3 Average absolute returns of S&P vs. $FinWeight_1(news)$, in-sample testing of news of 2006-2007	22
Figure 4.4 Average absolute returns of S&P vs. $FinWeight_2(news)$, in-sample testing of news of 2006-2007	22
Figure 4.5 Average absolute returns of S&P vs. $FinWeight_1(news)$, out-of-sample testing of news of 2005	23
Figure 4.6 Average absolute returns of S&P vs. $FinWeight_2(news)$, out-of-sample testing of news of 2005	24
Figure 4.7 Average absolute returns of S&P vs. $FinWeight_1(news)$, out-of-sample testing of news of 2008	24
Figure 4.8 Average absolute returns of S&P vs. $FinWeight_2(news)$, out-of-sample testing of news of 2008.....	25
Figure 5.1 Average absolute returns of S&P vs. $FinWeight(topics)$, in-sample testing of news of 2003-2004	27
Figure 5.2 Average absolute returns of S&P vs. $FinWeight(topics)$, in-sample testing of news of 2006-2007	27
Figure 5.3 Average absolute returns of S&P vs. $FinWeight(keywords)$, in-sample testing of news of 2003-2004	28
Figure 5.4 Average absolute returns of S&P vs. $FinWeight(keywords)$, in-sample testing of news of 2006-2007	29
Figure 5.5 Average absolute returns of S&P vs. $FinWeight(topics \text{ and } keywords)$, in-sample testing of news of 2003-2004	29

Figure 5.6 Average absolute returns of S&P vs. <i>FinWeight(topics and keywords)</i> , in-sample testing of news of 2006-2007	30
Figure 5.7 Average absolute returns of S&P vs. <i>FinWeight(topics)</i> , out-of-sample testing of news of 2005	31
Figure 5.8 Average absolute returns of S&P vs. <i>FinWeight(topics)</i> , out-of-sample testing of news of 2008.....	31
Figure 5.9 Average absolute returns of S&P vs. <i>FinWeight(keywords)</i> , out-of-sample testing of news of 2005.....	32
Figure 5.10 Average absolute returns of S&P vs. <i>FinWeight(keywords)</i> , out-of-sample testing of news of 2008.....	32
Figure 5.11 Average absolute returns of S&P vs. <i>FinWeight(topics and keywords)</i> , out-of-sample testing of news of 2005.....	33
Figure 5.12 Average absolute returns of S&P vs. <i>FinWeight(topics and keywords)</i> , out-of-sample testing of news of 2008.....	33
Figure 6.1 \hat{M} vs. \hat{I}_1 , out-of-sample testing of news, [2005-01-01, 2005-07-01)	36
Figure 6.2 \hat{M} vs. \hat{I}_2 , out-of-sample testing of news, [2005-01-01, 2005-07-01).....	37
Figure 6.3 \hat{M} vs. \hat{I}_1 , out-of-sample testing of news, [2005-07-01, 2006-01-01)	37
Figure 6.4 \hat{M} vs. \hat{I}_2 , out-of-sample testing of news, [2005-07-01, 2006-01-01).....	38
Figure 6.5 \hat{M} vs. \hat{I}_1 , out-of-sample testing of news, [2006-01-01, 2006-07-01)	38
Figure 6.6 \hat{M} vs. \hat{I}_2 , out-of-sample testing of news, [2006-01-01, 2006-07-01).....	39
Figure 6.7 \hat{M} vs. \hat{I}_1 , out-of-sample testing of news, [2006-07-01, 2007-01-01)	39
Figure 6.8 \hat{M} vs. \hat{I}_2 , out-of-sample testing of news, [2006-07-01, 2007-01-01).....	40
Figure 6.9 \hat{M} vs. \hat{I}_1 , out-of-sample testing of news, [2007-01-01, 2007-07-01)	40
Figure 6.10 \hat{M} vs. \hat{I}_2 , out-of-sample testing of news, [2007-01-01, 2007-07-01).....	41
Figure 6.11 \hat{M} vs. \hat{I}_1 , out-of-sample testing of news, [2007-07-01, 2008-01-01)	41
Figure 6.12 \hat{M} vs. \hat{I}_2 , out-of-sample testing of news, [2007-07-01, 2008-01-01).....	42
Figure 6.13 \hat{M} vs. \hat{I}_1 , out-of-sample testing of news, [2008-01-01, 2008-07-01)	42
Figure 6.14 \hat{M} vs. \hat{I}_2 , out-of-sample testing of news, [2008-01-01, 2008-07-01).....	43

Figure 6.15 \hat{M} vs. \hat{I}_1 , out-of-sample testing of news, [2008-07-01, 2009-01-01)	43
Figure 6.16 \hat{M} vs. \hat{I}_2 , out-of-sample testing of news, [2008-07-01, 2009-01-01)	44
Figure 6.17 \hat{M} vs. \hat{I}_1 , out-of-sample testing of news, all windows combined	45
Figure 6.18 \hat{M} vs. \hat{I}_2 , out-of-sample testing of news, all windows combined.....	45
Figure 6.19 \hat{M} vs. \hat{I}_1 , out-of-sample testing of news, all windows combined, 25% of the words.....	46
Figure 6.20 \hat{M} vs. \hat{I}_2 , out-of-sample testing of news, all windows combined, 25% of the words.....	47
Figure B.1 Keywords detection procedure for news published from October 2005 to March 2006.....	52
Figure B.2 Keywords detection procedure for news published from March, 2006 to present.....	53
Figure D.1 Similarity function used in duplicate removal stage of the preliminary preprocessing (part 1)	57
Figure D.2 Similarity function used in duplicate removal stage of the preliminary preprocessing (part 2)	58
Figure E.1 Number of data points for the results in Figure 6.17	59
Figure E.2 Number of data points for the results in Figure 6.18.....	59
Figure E.3 Number of data points for the results in Figure 6.19	60
Figure E.4 Number of data points for the results in Figure 6.20.....	60

ABSTRACT

When news data becomes available, development of automated news trading system is of great interest of many financial institutions. Since the stock market moves are dependent on the news releases that dependency can be analyzed. In this thesis we analyze news from Reuters and their influence on the moves of S&P 500 market index. We first investigate the data and develop preliminary preprocessing needed for the actual analysis, and then move to building the indicators for predicting short term intraday volatility. Performance of the indicators is considered in terms of the index price moves, and is used for the future development volatility prediction tools.

1. Introduction

It is a well-known fact that news stories influence the stock market prices and consequently the whole economy in general. Stock traders look at economic releases and announcements about corporate profits, interest rates, unemployment, and other big events such as elections, wars, political changes, and catastrophes in order to predict stock price movements.

With the advent of internet, news data became more accessible and therefore more useful for the financial institutions. Vendors such as Bloomberg or Reuters provide reliable news data almost immediately for subscribers. For many financial companies timely access to information and understanding of market moves are essential factors. So the development of automated news trading is realistic when the news data is available in electronic form.

Previous work relevant to this topic is sparse; immense research is done in the area of text processing, but its application to the financial markets is not well developed and widely used. This thesis presents the analysis of Reuters news data by means of applying text processing tools and statistical methods for supporting further development of volatility prediction and automated news trading system as a final goal. The scope of the thesis is mainly concerned with studying the news data, preprocessing it for the actual analysis and observing the performance of different indicators in regard to short term moves of S&P 500 index prices.

2. Data and General Description of the Analysis

2.1 Data

The news analysis presented in this thesis is based on the Reuters news data and S&P 500 minute price data, years 2003 – 2008. Data training is performed on 2003-2004 and 2006-2007 years, and testing is done on 2005 and 2008 years, respectively. Although, the data analysis is performed on two separate periods, the greater attention should be paid to the later one since its data is more recent and presumably more reliable.

Reuters database consists of events (rows) describing the elements (columns) (headline, story body, and codes) of a story: “ALERT”, “HEADLINE”, “STORY_TAKE_OVERWRITE”, “STORY_TAKE_APPEND” and “DELETE”. Each story has either “ALERT” or “HEADLINE” or both.

“ALERT” transmits a short sentence in uppercase, and is used to quickly report a breaking news information. “HEADLINE” event provides a short summary of a story. When “ALERT” or “HEADLINE” is present, the other events can be sent subsequently during the development of a story. “STORY_TAKE_OVERWRITE” provides text for the story’s body. “STORY_TAKE_APPEND” event is used to append an additional text if a story needs to be updated. “DELETE” event signifies that the story previously published (identified by its unique index) is no longer valid and needs to be neglected; majority of news with this event mainly occur when they are corrected or changed.

For simplicity of data processing “STORY_TAKE_APPEND” and “DELETE” events are not used in our analysis, assuming they do not play a significant role in affecting the market moves (refer Appendix A for details).

Each story is identified by its unique index “UNIQUE_STORY_INDEX”; so all events transmitted for a particular story share the same index.

A basic Reuters story consists of zero to many alerts, one to many bodies, and one to many headlines. Appending, correcting or updating a story produces multiple number of “ALERT”, “HEADLINE” and “STORY_TAKE_OVERWRITE” events issued for the same story – that makes data difficult to process. For this reason, preliminary data preprocessing is necessary.

To aid data preprocessing and eventually the whole news analysis, Reuters provides headline tags, story types, keywords and codes. The stories have an optional headline tag as a part of its headline text. For instance, “GLOBAL MARKETS” tag means that the given story is about “major global markets during time of exceptional activity”. Besides, each story has a story type, language it’s written in, and topics codes to convey the idea of what the story might be about. For instance, story type “M” stands for “Market Report”; stories written in English have language attribute “EN”, and topic code “RESF” stands for “Corporate Results Forecasts”.

2.2 General Description of Data Processing

We are mostly interested in dependence between news and US stock market, so US related news is the primary information for our analysis. The classification of news in the database as US-related can be determined based on the description of codes provided by Reuters. Every story has topic codes and optional named item codes, so, if it has at least one of these codes related to the US the news is marked as US-related.

The statistics presented in Chapters 2, 3 and 4 are based on the following scheme of data processing:

- (1) Retrieve the time series of a relevant time period from price data;
- (2) For each time T :
 - (a) Retrieve the news published from time ($T - n$ minutes) to T ;
 - (b) Retrieve the index prices (closing prices) at time T and ($T + k$ minutes).

Since we are interested in the short term moves of the index prices, we set the time interval n to 30 minutes, and return period k to 60 minutes at each data point. We compute a simple 1 hour return R_T as $\frac{P_{(T+k)} - P_T}{P_T}$, where P_T is the index price at T and $P_{(T+k)}$ is the index price in 1 hour.

Therefore, each data point at time T represents two values: the first is a value of an indicator for the news published from time T to ($T - n$ minutes), and the second is 1 hour return taken from T to ($T + k$ minutes).

Price data is given only for the days when the market is open, so taking all the time series of the price data allows us to automatically ignore the news published on weekends and holidays.

All the results shown on charts in Chapters 2, 3, 4 are produced by using the most active data regions (there are at least 200 data points at each bin of the indicators).

2.3 Preliminary Preprocessing

It is common for any news article to place the most important information into its headline and 1st paragraph. So our data analysis is based on the processing of headlines and 1st paragraphs only, simplifying the analysis and speeding up the computations.

Likewise, the words set is large, but only small portion of it is used consistently; for instance, in news of 2006 (written in English), words with frequency of at least 33 per year comprise only 10% of all words used in that year. We need to exclude infrequent words from analysis in order to make results statistically significant. Main steps of data conversion:

- (a) News not in English are ignored.
- (b) All corrections are ignored so there is no more than one headline and body exists for each story (see Appendix A for details).
- (c) News identified by “SERVICE ALERT” tag are ignored since they do not affect the market moves.
- (d) Keywords are extracted from text of a story and written into a separate field (see Appendix B for details).
- (e) The whole text of news bodies is reduced to one paragraph (or a maximum of 80 words) (see Appendix B for details).
- (f) Words’ characters are changed into lower or upper case based on the original case and frequency of the words (for details, see Appendix C). Punctuation characters and digits are removed.
- (g) All words are stemmed except named entities; we define a named entity as a string of all uppercase characters, after stage (f) is performed.
- (h) Although, the whole set of words is significantly reduced after the stemming procedure, similar words still remain there. For instance, such words as “academi”,

“academia”, “ACADEMICA”, “ACADEMIE” and “ACADEMY” (used in 2006) are all different words is character-wise sense. Since they all have the common prefix “academ” we can map each word to the most frequent word “ACADEMIA”, - reducing the set of 5 words to 1. So for this mapping we use the length of words and the length of a common prefix. As in the given example, the words are grouped together because they all fall into the group of words with length [7, 9] and the common prefix of length 6. By using this procedure, we are able to decrease the number of words by roughly 36000 in year 2006 (after stemming).

- (i) Since “STORY_TAKE_OVERWRITE” event transmits both the headline and the body, there is no need to keep track of the “HEADLINE” event. So we ignore the “HEADLINE” events if they are transmitted within 5 seconds from the “STORY_TAKE_OVERWRITE”, otherwise, we treat them as a separate story.
- (j) There are cases when Reuters transmit identical or very similar news but with different unique story indexes, - as a result, repeating the same news more than once. In this stage we delete duplicate news.

To define similarity of the news, we use the words intersection and the “informativeness” of those words from the compared stories (“informativeness” is discussed in Chapter 3.1). The similarity of two stories S_1 and S_2 is computed as follows:

$$Similarity = \frac{\sum_{k=1}^n \min(FreqCy(w_k \in S_1), FreqCy(w_k \in S_2)) \times InformS(w_k)}{\max(InformS(S_1), InformS(S_2))},$$

where w_k is a word from the intersection of S_1 and S_2 , $InformS(w_k)$ is the informativeness of w_k , $FreqCy(w \in S)$ is the frequency of the word w from S and $InformS(S)$ is the total informativeness of all words from S (the code snippet can be found in Appendix D).

To find a duplicate we set similarity threshold by comparing the pairs of news and observing their similarity value. So the news with similarity value in range [0.65, 1] are considered to be duplicates of each other. For efficiency, we remove a duplicate if it’s published within one hour period from the original story.

For instance, using this procedure the number of news of 2006 year is reduced from 1366513 to 990897.

2.4 Word Usage

Although, the size of the word set is reduced significantly after the preprocessing procedure, the number of words still remains large. For instance, the set of all words used in 2006 news after the preprocessing stage comprises 190759 different words, but the number of frequent words is much smaller; the Figure 2.1 shows the percentage of the words on y-axis and their frequency on x-axis. As we can see, the majority of the words are very uncommon: 42% of the words are used only once in that year. In fact, 70% are used less than 7 times and only 10% have frequency of at least 44.

To aid efficiency in our news analysis we can ignore the uncommon words and consider only words that are relatively frequent. So, for the analysis we only take the words appearing in at least 20 days per year of a particular training period.

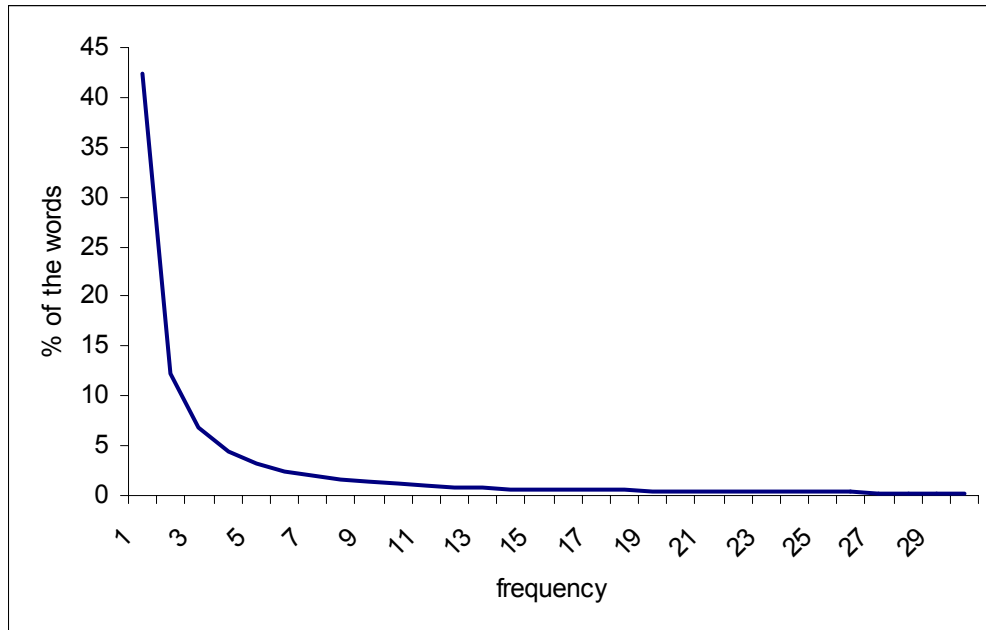


Figure 2.1 Number of words used in Reuters news in 2006

3. Building Informative Indicators

3.1 Informativeness

Any sentence consisting of one or more words can form a meaningful statement. Some words may appear very often regardless of the subject, other words may appear in more specific news. For instance, stop words such as “the” or “about” can be found in any news. On the other hand, words “market” or “bank”, for example, are more likely encountered in financial news than in any other news. Hence, we can say that the words “market” and “bank” hold more meaningful information for our analysis than the words “the” or “about”.

To measure the “informativeness” of a word w we can use frequency of the words appearing with w and compare it with the general frequency distribution of the same words taken from all news, - that is, the bigger the change in the frequency distributions the bigger the informativeness of w .

The informativeness of w is computed as follows:

$$InformS(w) = \sqrt{\frac{\sum_{i=1}^n (genFreqCy(w_i) - coFreqCy(w_i))^2}{n}},$$

where $genFreqCy(w_i)$ is the general frequency of the word w_i from all the news, $coFreqCy(w_i)$ is the frequency of the word w_i from only the articles where the word w appears, and n is the total number of all words used.

For example, the word “REUTERS” has a small value of the informativeness of 0.227×10^{-3} because it appears in most of the news as source of an article; on the other hand, the word “BLOOMBERG” has bigger informativeness of 0.713×10^{-3} since it appears in smaller number of news and, likely, in more specific news related to the Bloomberg itself.¹

Although, the informativeness is used in duplicate removal (Chapter 2.4), the main purpose of the current chapter is to examine the news informativeness with respect to the absolute moves of S&P 500 index prices.

¹ In the given example, the informativeness is computed based on the news of 2006 and 2007 years.

3.2 Informativeness and Short Term Moves of S&P 500

Analyzing the news' informativeness we define the potential informative indicators presented below; their performance is examined on the training data of 2003-2004 and 2006-2007 years and testing data of 2003-2004 and 2006-2007, respectively:

$$(a) \text{Inform}S_1(\text{news}) = \sum_{i=1}^n \sum_{j=1}^k \text{Inform}S(w_j);$$

$$(b) \text{Inform}S_2(\text{news}) = \frac{\sum_{i=1}^n \sum_{j=1}^k \text{Inform}S(w_j)}{n};$$

$$(c) \text{Inform}S_3(\text{news}) = \frac{\sum_{i=1}^n \sum_{j=1}^k \text{Inform}S(w_j)}{\sum_{i=1}^n k};$$

$$(d) \text{Inform}S_4(\text{news}) = \sum_{i=1}^n \frac{\sum_{j=1}^k \text{Inform}S(w_j)}{k},$$

where n is the number of news, and k is the number of words in each story.

Also, note that at this point we do not categorize the news other than into the group of US news (for details, see Chapter 1.4), - all types of news whether financial or not are considered. Chapters 3.3 and 3.4 present the results obtained from the in-sample and out-of-sample testing; each Figure shows the average absolute returns in basis points on y-axis at each bin of the informative indicators on x-axis.

3.3 In-sample Performance of the Informative Indicators

The dependency of the market index on indicators is comparable in both training data of 2003-2004 and 2006-2007.

$\text{Inform}S_1(\text{news})$ produces unstable difference in average absolute return by about 5 basis points. This outcome may be explained by the following reasons: first, the number of words can vary significantly from one story to another, and second, the news irrelevant to financial markets combined with the financial news may cause data distortion.

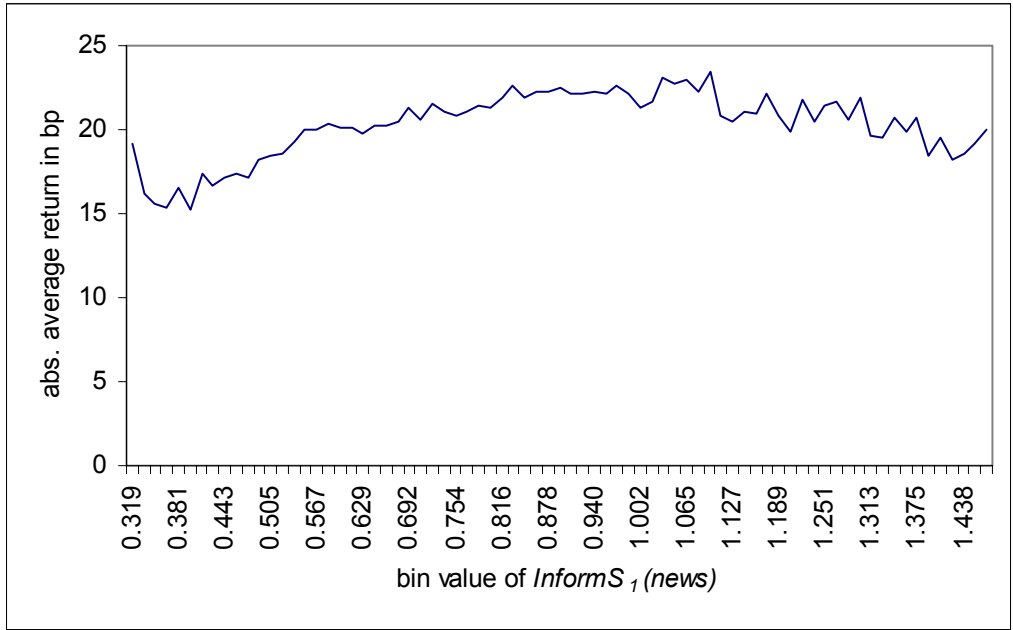


Figure 3.1 Average absolute returns of S&P vs. $InformS_1(news)$, in-sample testing of news of 2003-2004

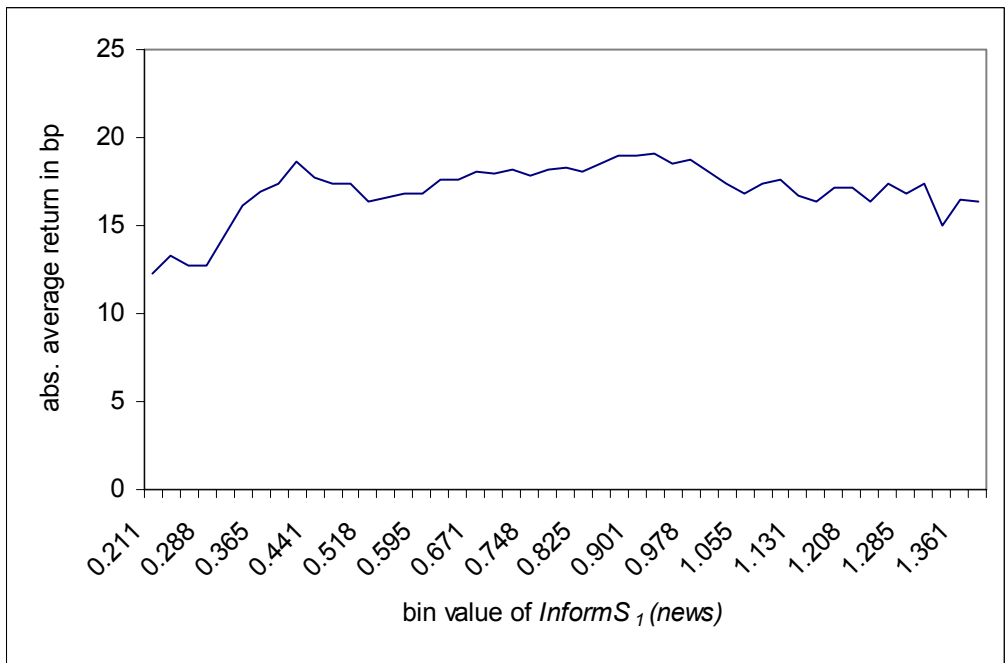


Figure 3.2 Average absolute returns of S&P vs. $InformS_1(news)$, in-sample testing of news of 2006-2007

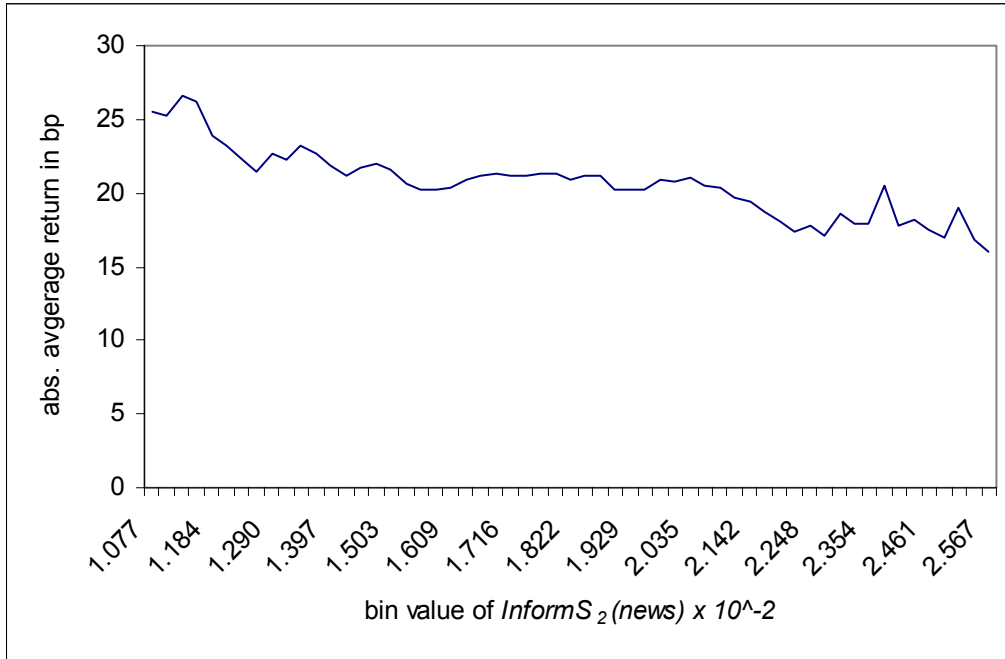


Figure 3.3 Average absolute returns of S&P vs. $InformS_2(news)$, in-sample testing of news of 2003-2004

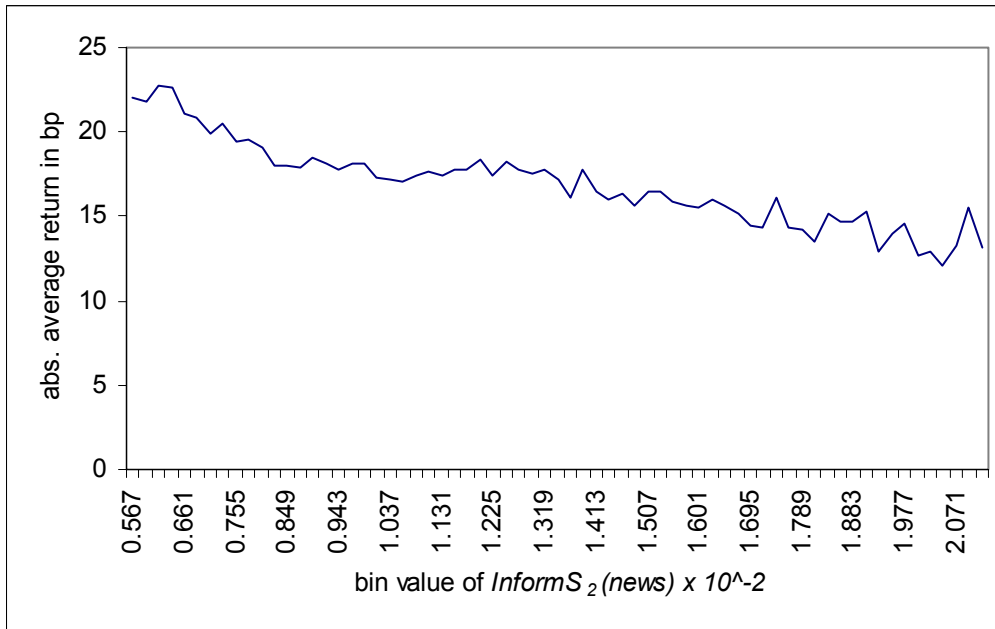


Figure 3.4 Average absolute returns of S&P vs. $InformS_2(news)$, in-sample testing of news of 2006-2007

On the other hand, the results shown for $InformS_2(news)$ indicator reveal that in both cases during the training periods the difference in returns for different bin numbers is up to 10 basis points. We can explain this behavior by stating that if the news is very

informative and relevant to the financial markets it should be concise and relatively short. Therefore, transmitting many news such as alerts, for instance, can lead to the reduction of $InformS_2(news)$ and increase of the absolute returns, accordingly.

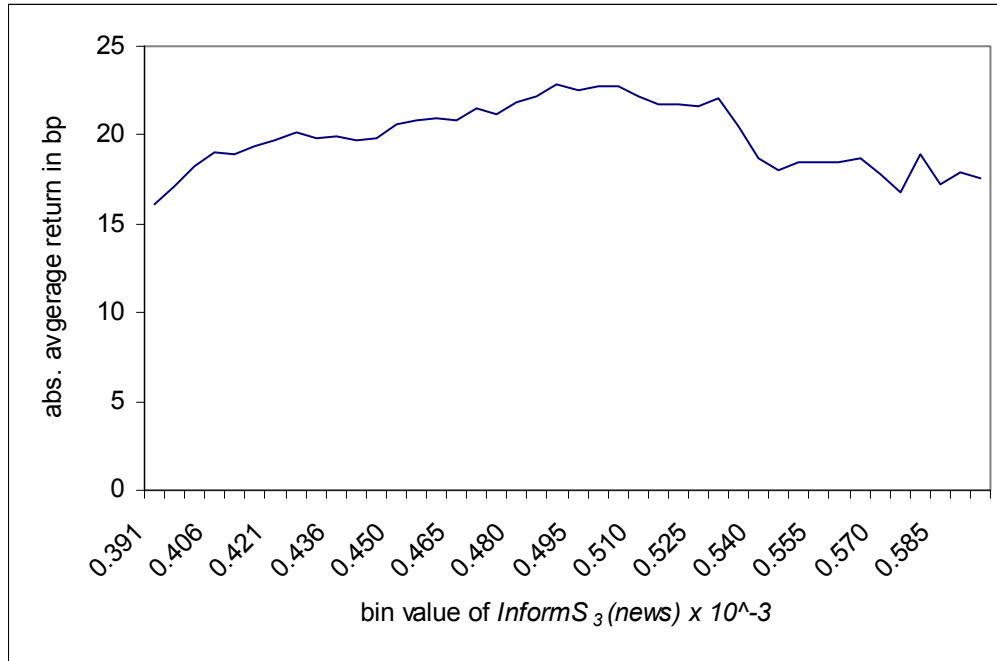


Figure 3.5 Average absolute returns of S&P vs. $InformS_3(news)$, in-sample testing of news of 2003-2004

$InformS_3(news)$ indicator shows how the average informativeness per word for the news at a particular data point influences absolute market index returns. Observing $InformS_3(news)$ in Figures 3.5 and 3.6 we can see that the moves are not greatly affected by this indicator; the absolute returns are fluctuating up and down by roughly 2-5 basis points.

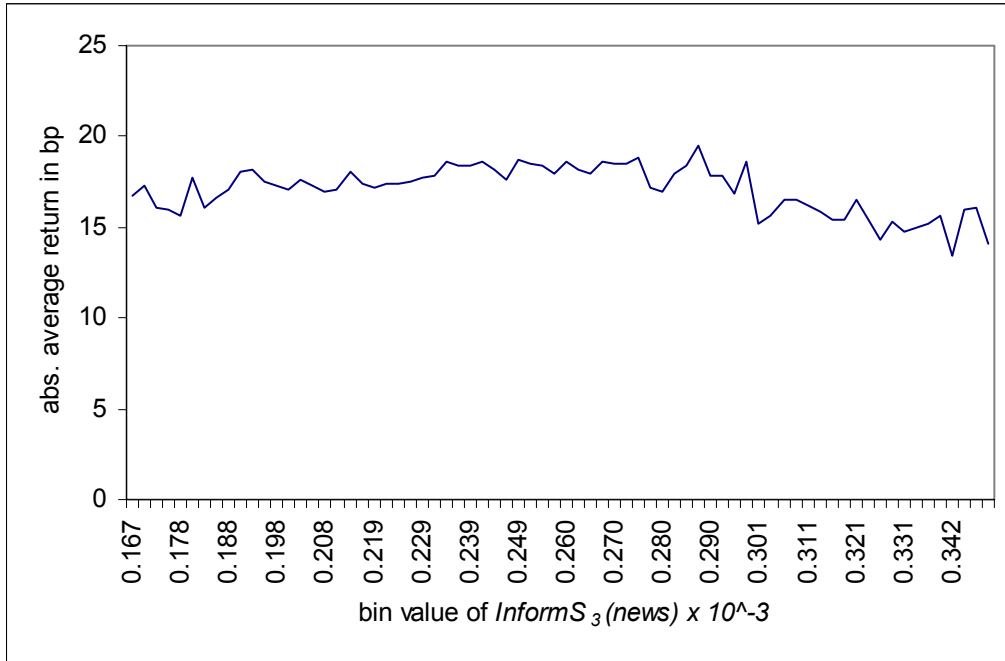


Figure 3.6 Average absolute returns of S&P vs. $InformS_3(news)$, in-sample testing of news of 2006-2007

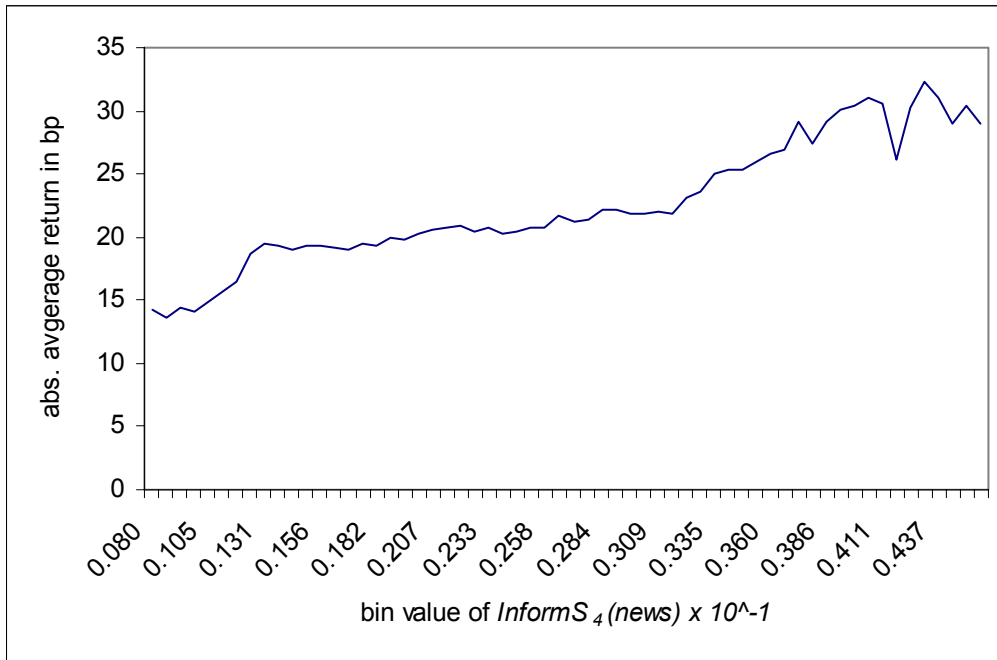


Figure 3.7 Average absolute returns of S&P vs. $InformS_4(news)$, in-sample testing of news of 2003-2004

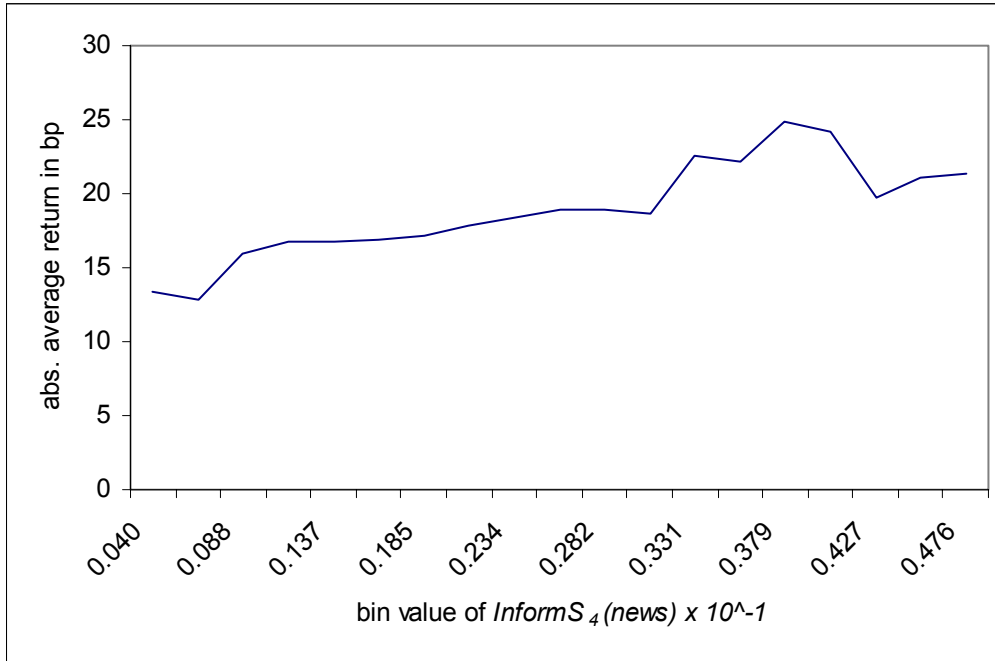


Figure 3.8 Average absolute returns of S&P vs. $InformS_4(news)$, in-sample testing of news of 2006-2007

On the other hand, summing up the average informativeness per word per story gives us much better performance as shown in Figures 3.7 and 3.8. In both training periods the returns are gradually growing with the increase of $InformS_4(news)$, recording the growth of up to 10-15 basis points.

From the constructed results in the Figures above we can see that $InformS_4(news)$ performs better than the other informative indicators proposed so far; for supporting this statement we should carry the additional experiments on the out-of-sample data as shown below.

3.4 Out-of-sample Performance of the Informative Indicators

The out-of-sample processing of $InformS_1(news)$ generating the moves on both testing data of 2005 and 2008 years shows similar results as performed in-sample.

So far, we do not have a good evidence that $InformS_1(news)$ could be useful in the volatility prediction since the trend remains unstable as shown in Figures 3.9 and 3.10.

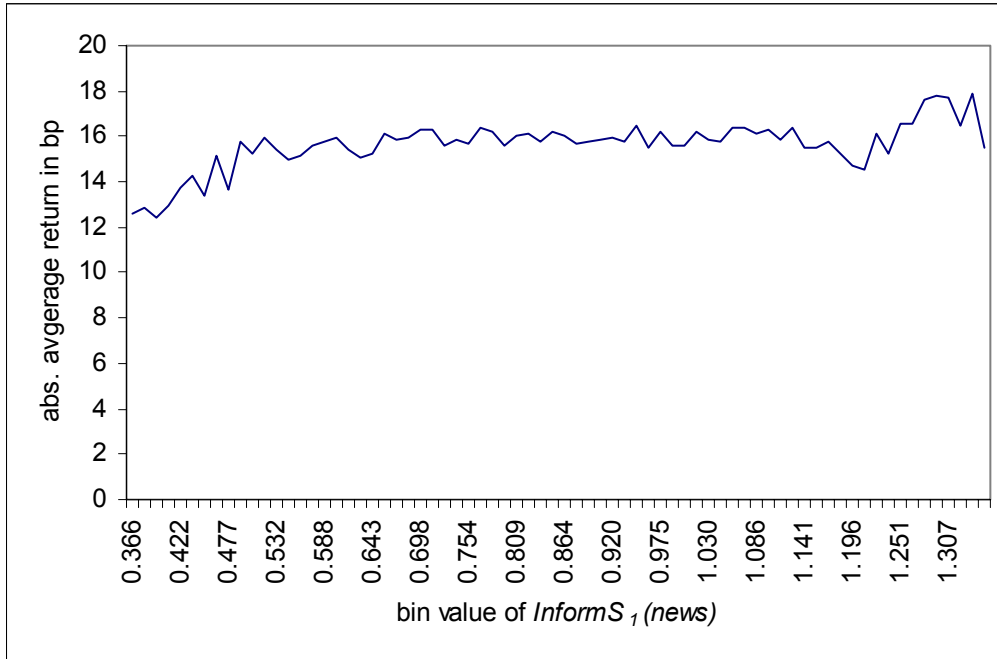


Figure 3.9 Average absolute returns of S&P vs. $InformS_1(news)$, out-of-sample testing of news of 2005

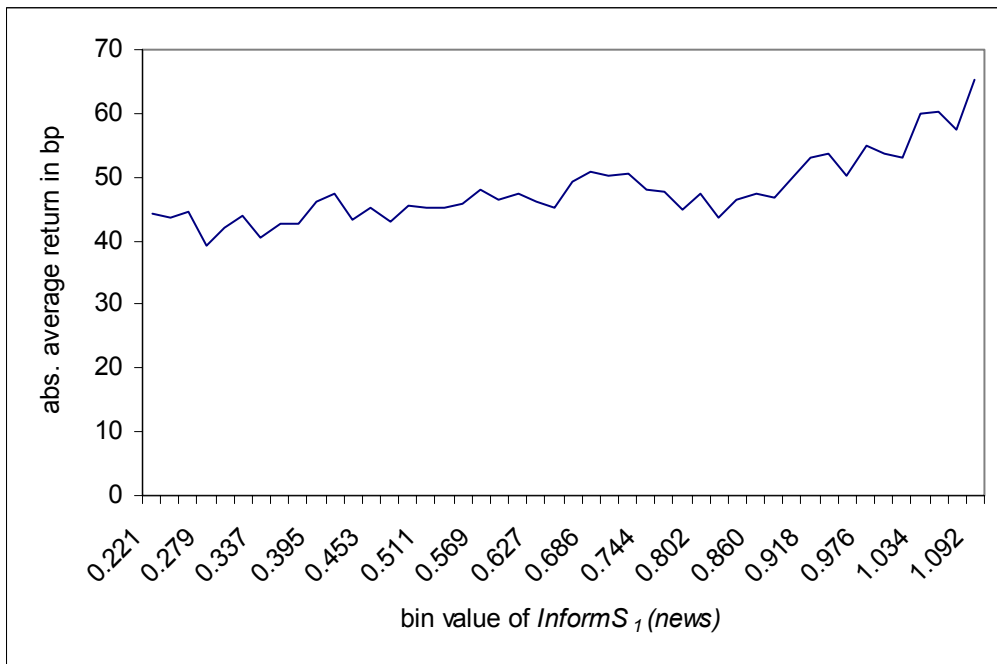


Figure 3.10 Average absolute returns of S&P vs. $InformS_1(news)$, out-of-sample testing of news of 2008

As with the results demonstrated on the training data, the constructed out-of-sample results of $InformS_2(news)$ also indicate significant trend of absolute returns in descending

direction; the moves are gradually moving down, approximately, for up to 5 basis points in 2005 and for up to 40 basis points in 2008.

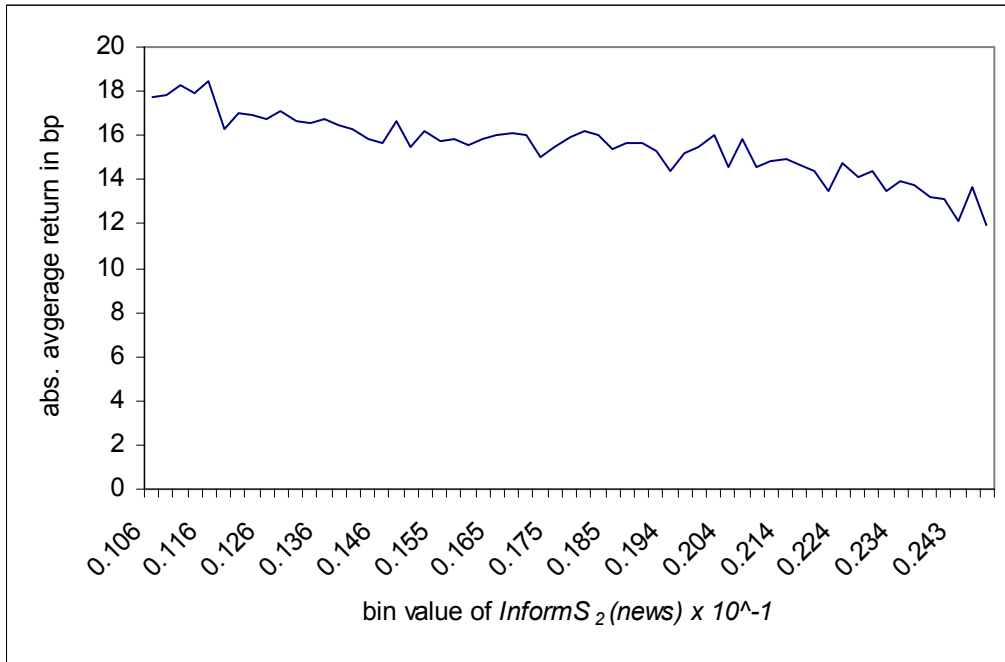


Figure 3.11 Average absolute returns of S&P vs. $InformS_2(news)$, out-of-sample testing of news of 2005

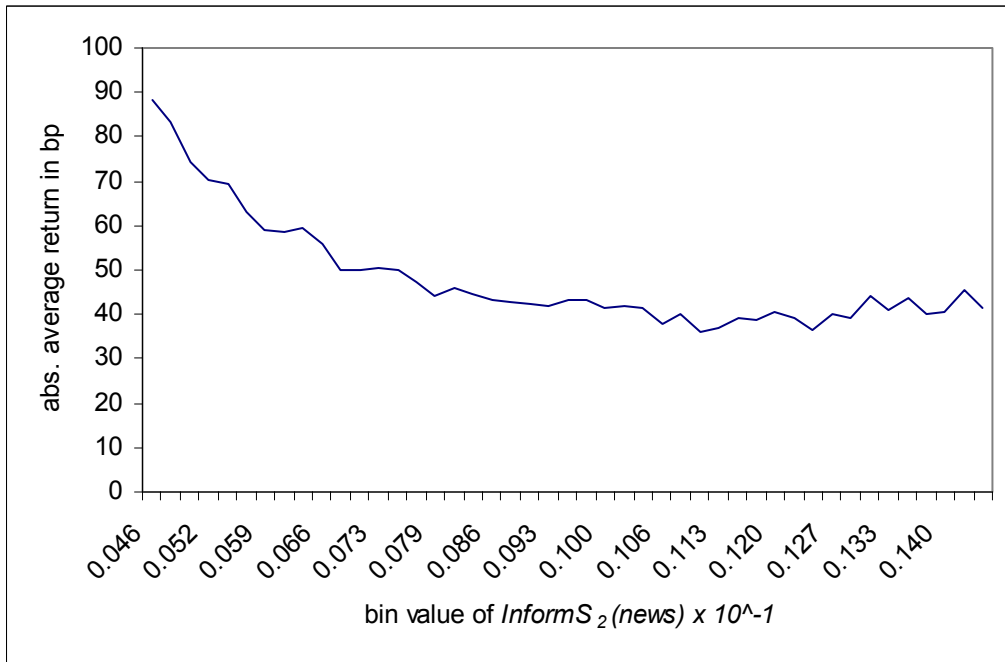


Figure 3.12 Average absolute returns of S&P vs. $InformS_2(news)$, out-of-sample testing of news of 2008

The out-of-sample testing of 2005 shows similar results as with in-sample testing of 2003-2004 for $InformS_3(news)$. However, the out-of-sample results of 2008 are not comparable to the in-sample results of 2006-2007; the moves are going down by at least 30 basis points.

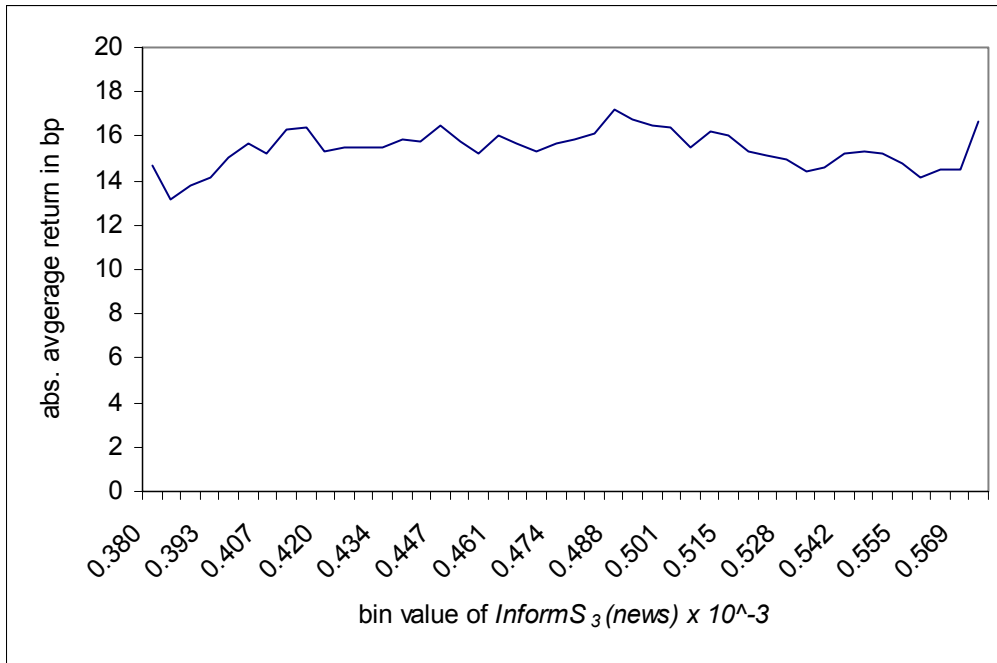


Figure 3.13 Average absolute returns of S&P vs. $InformS_3(news)$, out-of-sample testing of news of 2005

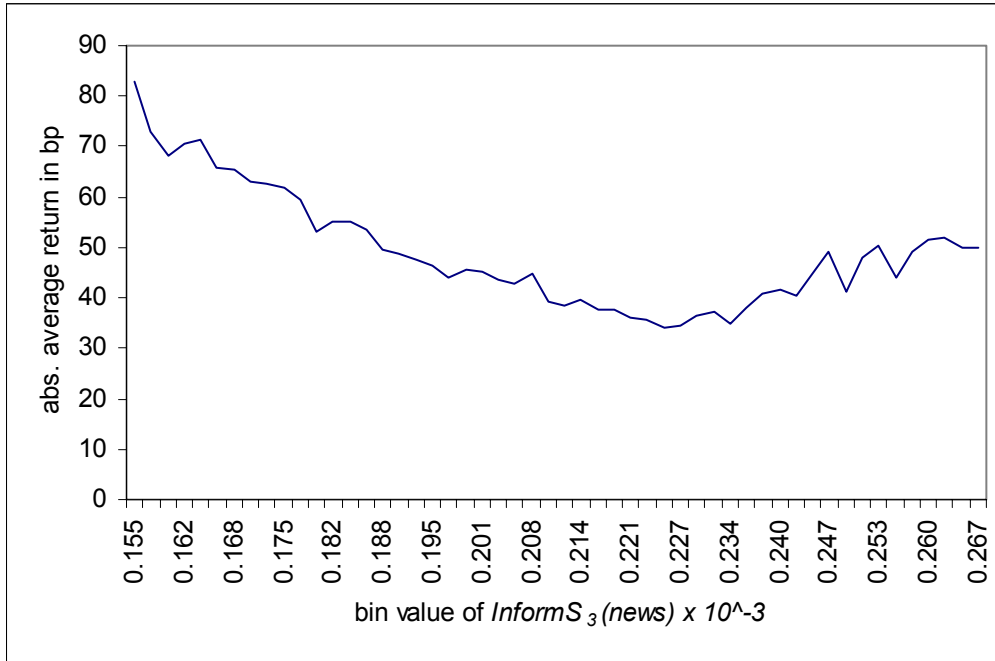


Figure 3.14 Average absolute returns of S&P vs. $InformS_3(news)$, out-of-sample testing of news of 2008

The data testing of 2005 year for $InformS_4(news)$ shows unstable results in Figure 3.15. The returns are not changing much during the testing; there is a barely visible trend of the absolute returns for this indicator. Surprisingly, the moves resulted from testing of 2008 are going steadily up with increase of the indicator, and are similar to the moves of the in-sample testing. Since these results are produced from processing of the most recent data, we may conclude that $InformS_4(news)$ is indeed the most productive indicator among all.

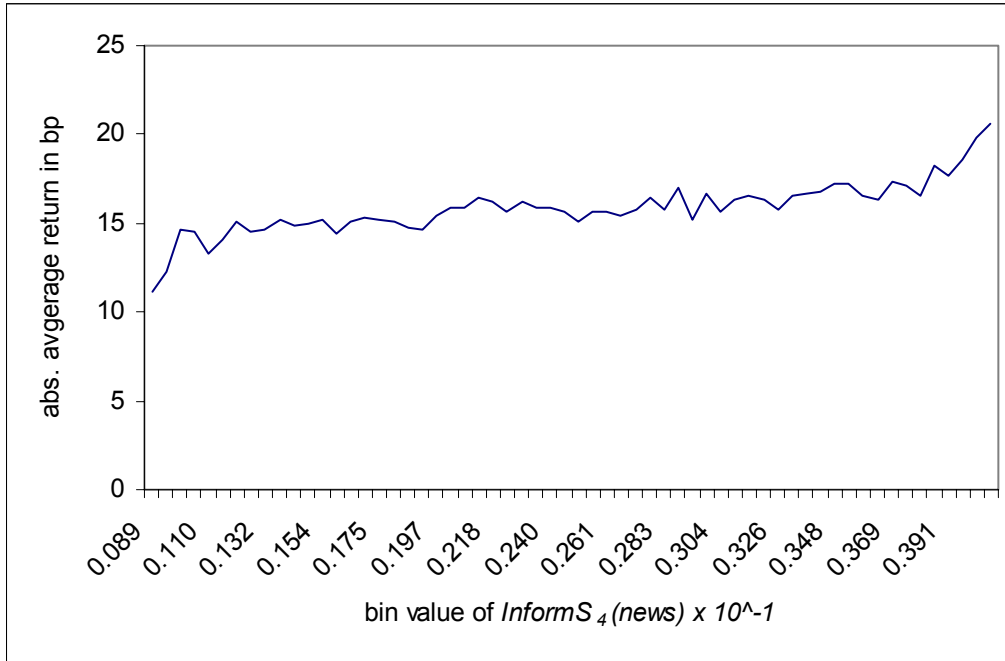


Figure 3.15 Average absolute returns of S&P vs. $InformS_4(news)$, out-of-sample testing of news of 2005

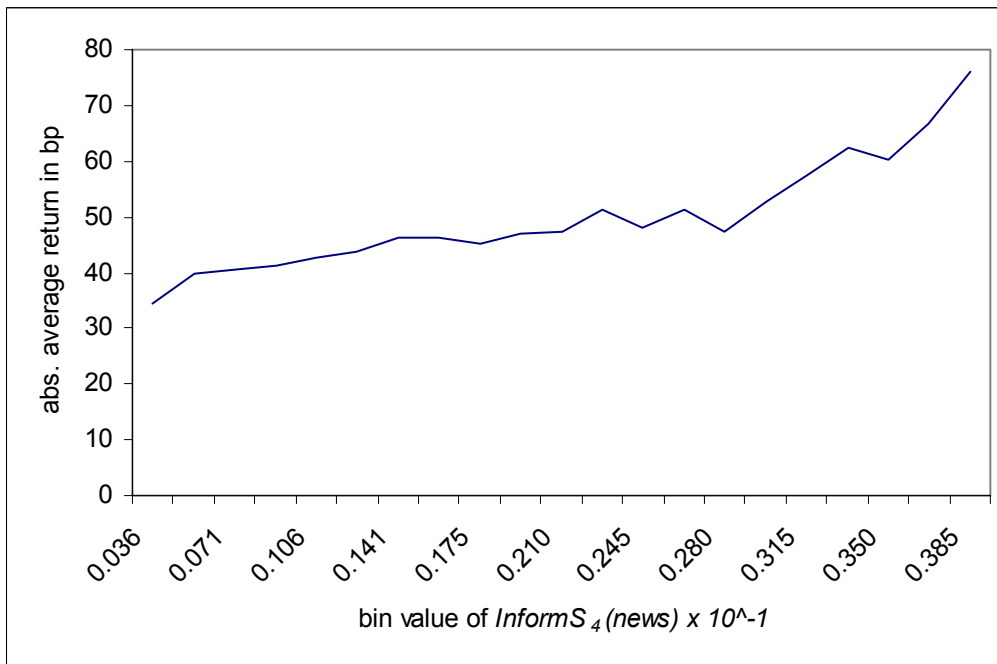


Figure 3.16 Average absolute returns of S&P vs. $InformS_4(news)$, out-of-sample testing of news of 2008

4. Building Financial Indicators

4.1 Financial Words

Any article related to finance, whether it's about a stock market, oil prices, bonds, mortgages, or the government's budget, - is always composed of the "financial" words describing a matter of a story. So we define a financial word as any word that potentially may affect the stock market moves; and, therefore, imply that most of them appear in the financial articles. So our goal in this chapter is to determine the financial words from the whole pool of the US news, and eventually using them in the short-term intraday volatility prediction. Such words can be determined in two ways: first, is to manually pick the words based on their underlying meaning; second, is to statistically assign a financial weight to each word based on the average of the returns associated with them. Because the second method is automated in nature we use it in the analysis.

4.2 General Description of the Financial Weights Assignment

We define the financial weight of a word as an average of 1 hour absolute returns right after the news (with that word) is published; in other words, if the news is published at time T the simple absolute return would be taken from time T to $T + k$ minutes and recorded for each word from the news. Therefore, the financial weight of the words is equal to the average of all of the absolute returns associated with them.

The assignment of the weights is performed as follows:

- (1) Retrieve news from time T_{open} to time $(T_{closed} - k \text{ minutes})$, where T_{open} is time when the market is open and T_{closed} is time when the market is closed, $k = 60$;
 - (2) For each story;
 - (a) Make the set of words;
 - (b) Get absolute return from time T to $(T + k \text{ minutes})$, where T is time of the closing index price; record it for each word from the set;
 - (3) For all the given words find the average of the absolute returns associated with them.
- Also, note that since the price data covers only the days when the market is open, we can ensure that news published on weekends and holydays are not used in the processing.

4.3 Financial Indicators

When the financial weights are defined the following potential indicators are useful for determining the financial value of the news:

$$(a) \text{FinWeight}_1(\text{news}) = \sum_{i=1}^n \sum_{j=1}^k \text{FinWeight}(w_j);$$

$$(b) \text{FinWeight}_2(\text{news}) = \sum_{i=1}^n \frac{\sum_{j=1}^k \text{FinWeight}(w_j)}{k},$$

where n is the number of news, and k is the number of words from the set of words in each news.

Since we are interested in the performance of words with the most prominent weight the data processing is made using only 50% of all the words with highest financial weights. The in-sample and out-of-sample results are presented below for each indicator.

4.3.1 In-sample Testing of the Financial Indicators

From the Figures 4.1-4.4 we can see that by processing the financial weights on training data we produce expected results for absolute returns. Also, note that both indicators are relatively similar; the absolute return difference is up to 15-20 basis points in both cases.

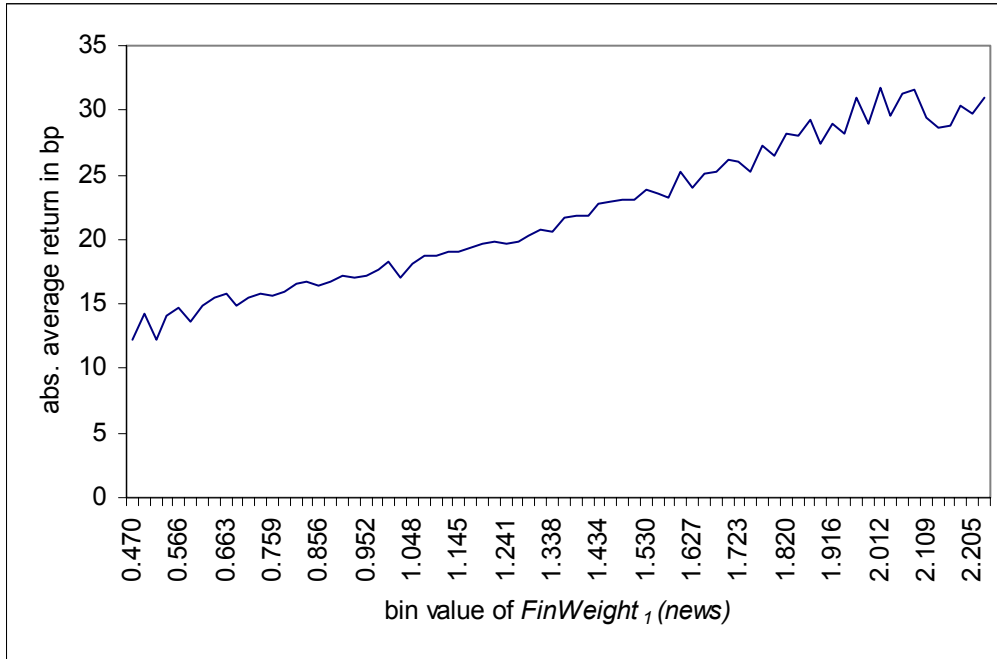


Figure 4.1 Average absolute returns of S&P vs. $FinWeight_1(news)$, in-sample testing of news of 2003-2004

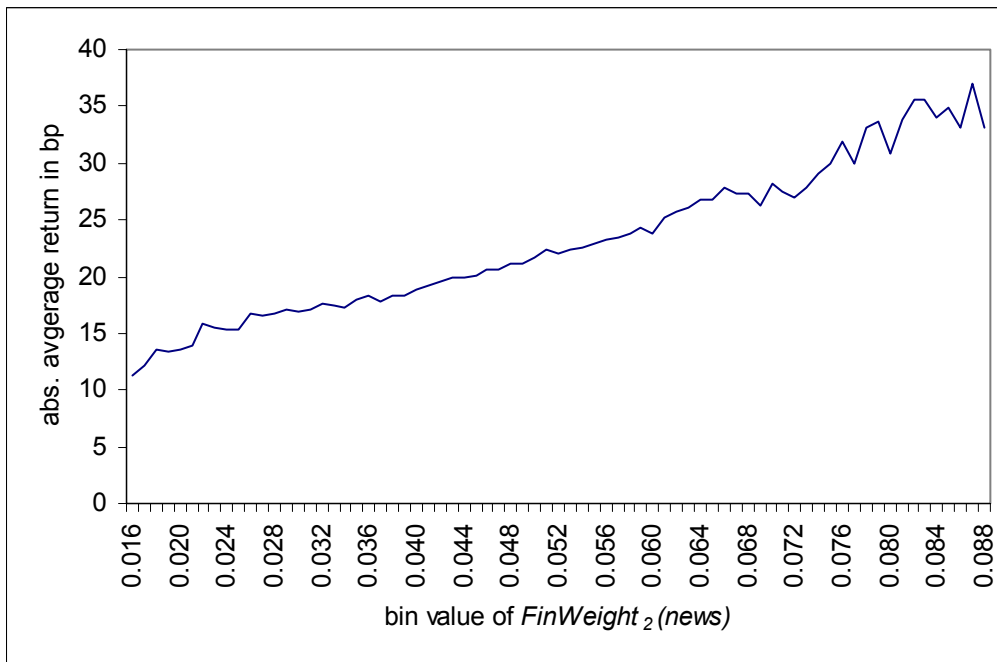


Figure 4.2 Average absolute returns of S&P vs. $FinWeight_2(news)$, in-sample testing of news of 2003-2004

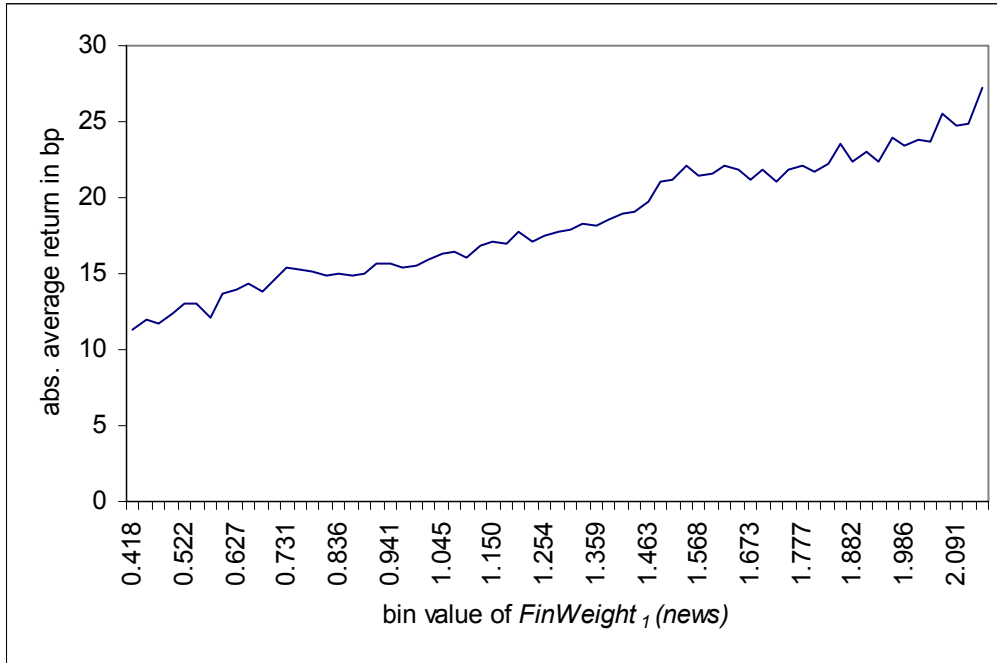


Figure 4.3 Average absolute returns of S&P vs. $FinWeight_1(news)$, in-sample testing of news of 2006-2007

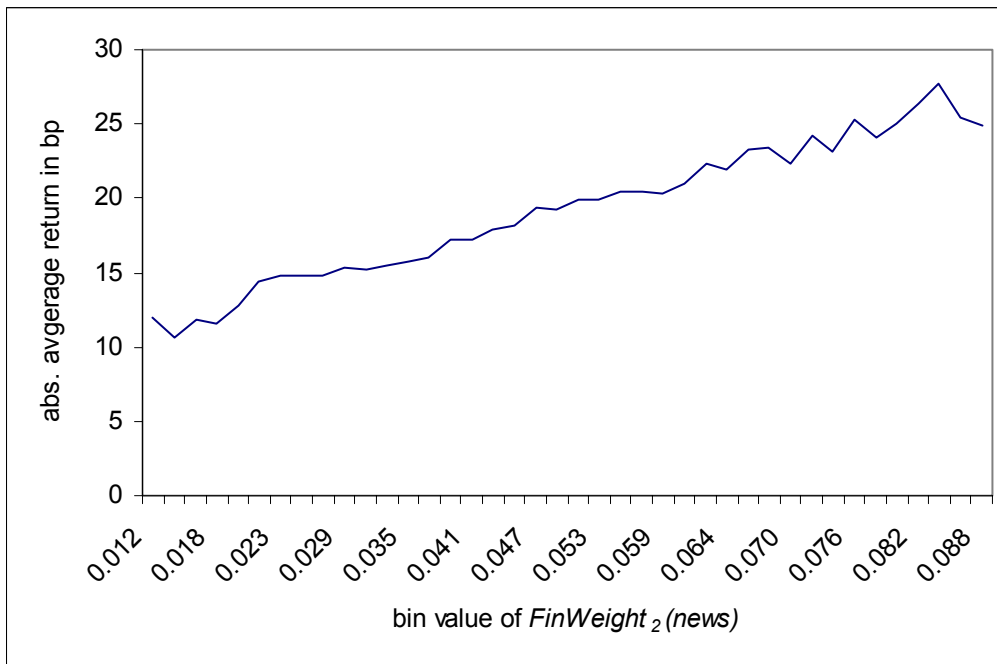


Figure 4.4 Average absolute returns of S&P vs. $FinWeight_2(news)$, in-sample testing of news of 2006-2007

Next step is to check the financial weights obtained from the training data on the testing data.

4.3.2 Out-of-sample Testing of the Financial Indicators

The out-of-sample testing is performed on 2005 and 2008 years using the financial weights of 2003-2004 and 2006-2007, respectively.

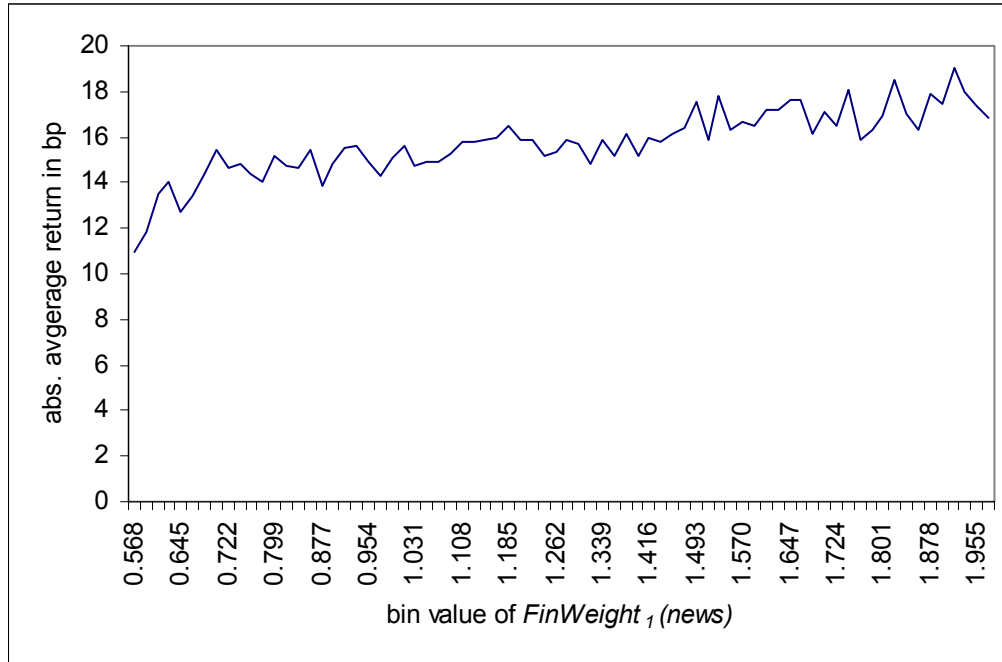


Figure 4.5 Average absolute returns of S&P vs. $FinWeight_1(news)$, out-of-sample testing of news of 2005

Figures 4.5-4.6 show that the absolute returns are gradually growing with the increase of the financial indicators, roughly, for up to 7 basis points; this increase of the moves is not as good as in in-sample testing, but still is promising.

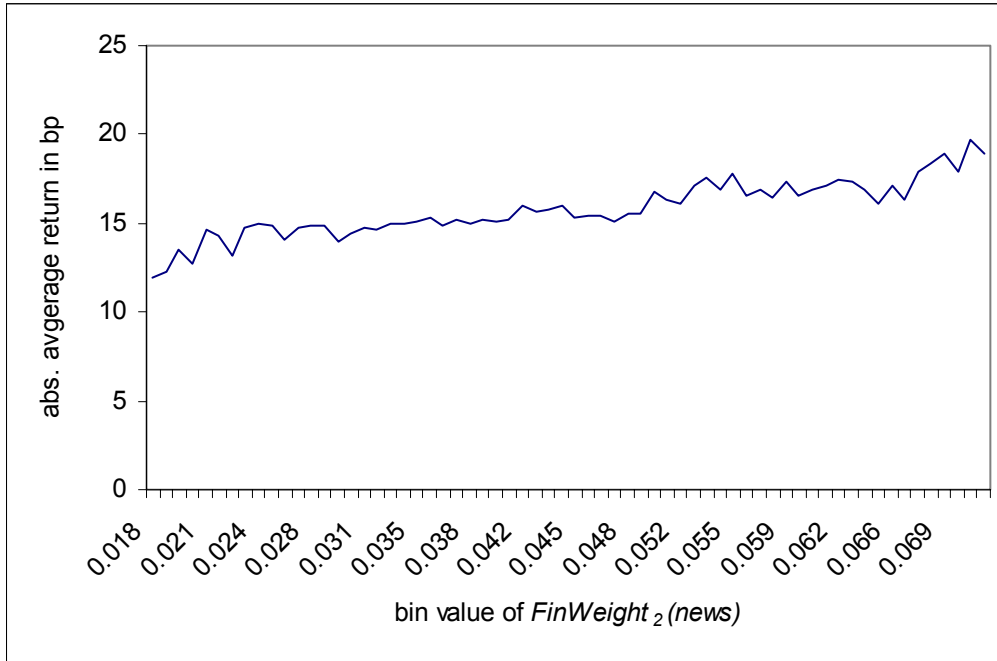


Figure 4.6 Average absolute returns of S&P vs. $FinWeight_2(news)$, out-of-sample testing of news of 2005

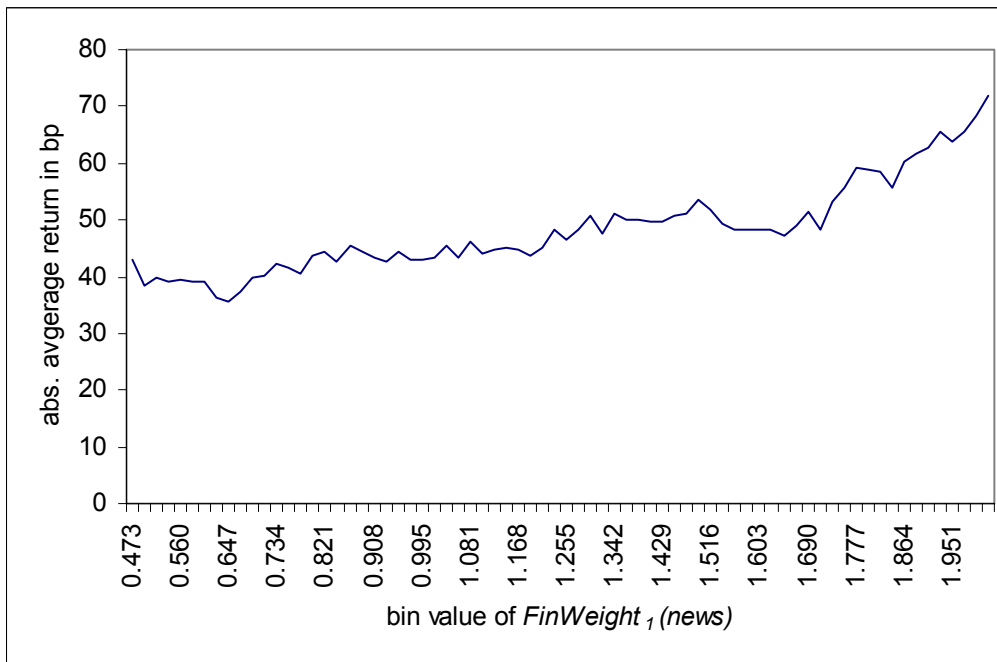


Figure 4.7 Average absolute returns of S&P vs. $FinWeight_1(news)$, out-of-sample testing of news of 2008

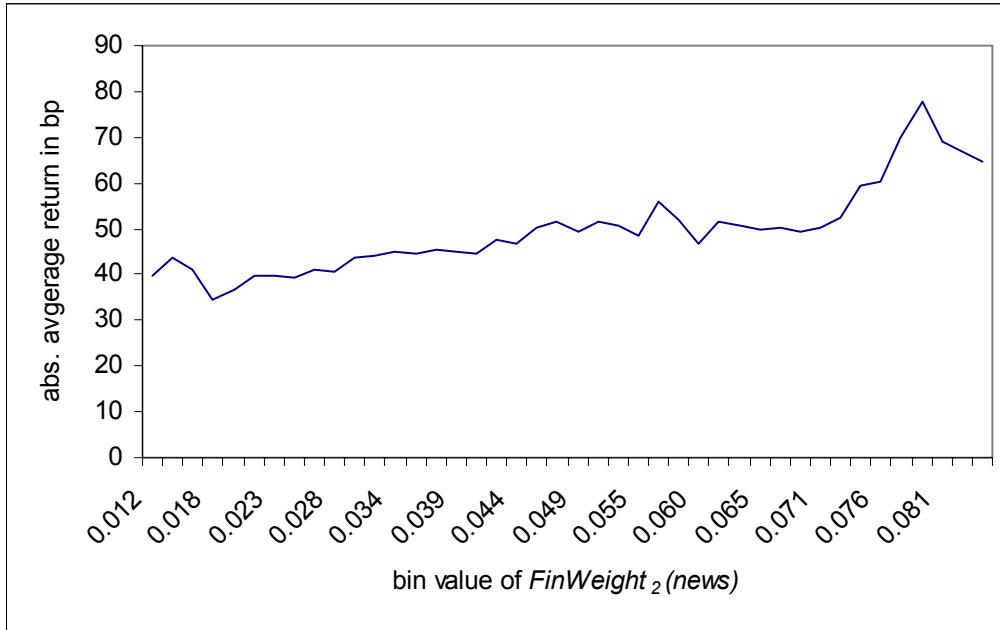


Figure 4.8 Average absolute returns of S&P vs. $FinWeight_2(news)$, out-of-sample testing of news of 2008

From Figures 4.7-4.8 we can see that the performance of the indicators in the testing of 2008 is significantly better than of 2005 data; absolute average returns are going up at least 20 basis points.

5. High Level Based Indicators

5.1 Financial Weights of Topic Codes and Keywords

Reuters provide topic codes and keywords to categorize a story. Topic codes are given as a sequence of codes in the field “TOPICS” for each story; they are assigned by journalists to each story, describing the subject of the story, language the story, or region the story is about. Likewise, keywords consist of two or three words used internally; primarily, the first word indicates the general topic or sector, the second names the country or institution to which the story refers and the third specifies the subject matter.

The purpose of this chapter is to determine whether the exclusive use of topics and/or keywords can influence the market moves. As with the financial words discussed in the previous chapter, we assign a financial weight to each topic (keyword) obtained from the training data; consequently, the weights are also checked on the testing data. The financial indicator is defined as follows:

$$FinWeight(topics / keywords) = \sum_{i=1}^n \frac{\sum_{j=1}^k FinWeight(w_j)}{k},$$

where n is the number of news, and k is the number of topics (keywords) from the set of topic (keyword) words in each story.

We do not consider “ALERT” events in our analysis for the reason that the same topic codes are repeated in the “STORY_TAKE_OVERWRITE” events, and also, “ALERT” events are not provided with the keywords.

Additionally, the weights are taken only for the relatively frequent topics (keywords) – appeared at least 250 (200) times per year, respectively. The total number of such topics (keywords) in 2003-2004 and 2006-2007 years is 292 (504) and 397 (433) words, respectively. Moreover, we include only 50% of the topics (keywords) with highest weight in the similar way as in the processing of the financial words in Chapter 4.3.

5.2 In-sample Testing of the High Level Based Indicators

Figures 5.1-5.6 show the results obtained from the training data. As expected, the absolute returns are going up with the increase of $FinWeight(topics/keywords)$ indicator.

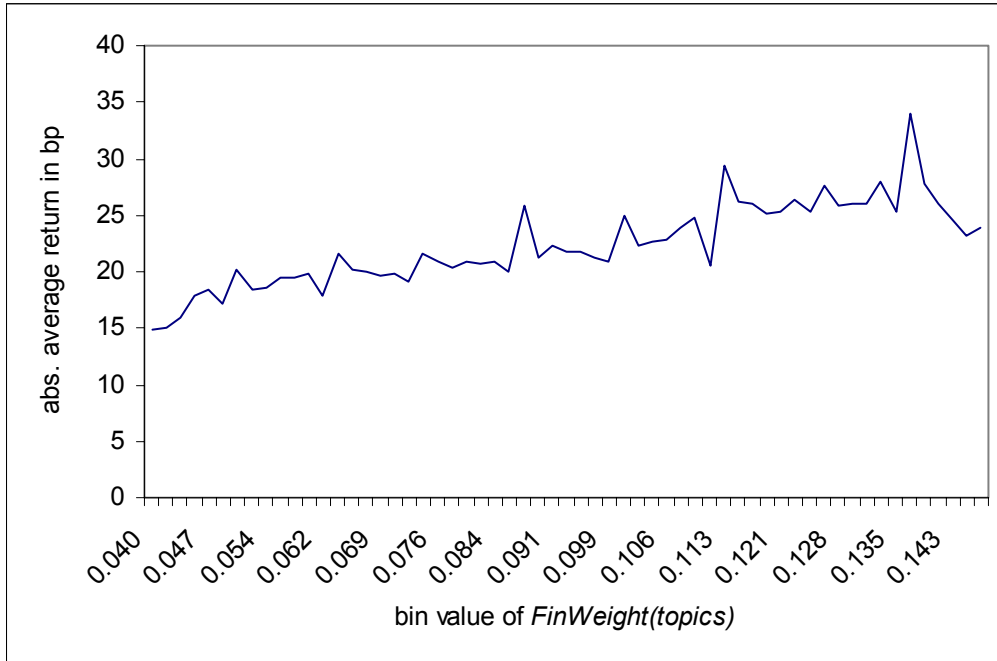


Figure 5.1 Average absolute returns of S&P vs. *FinWeight(topics)*, in-sample testing of news of 2003-2004

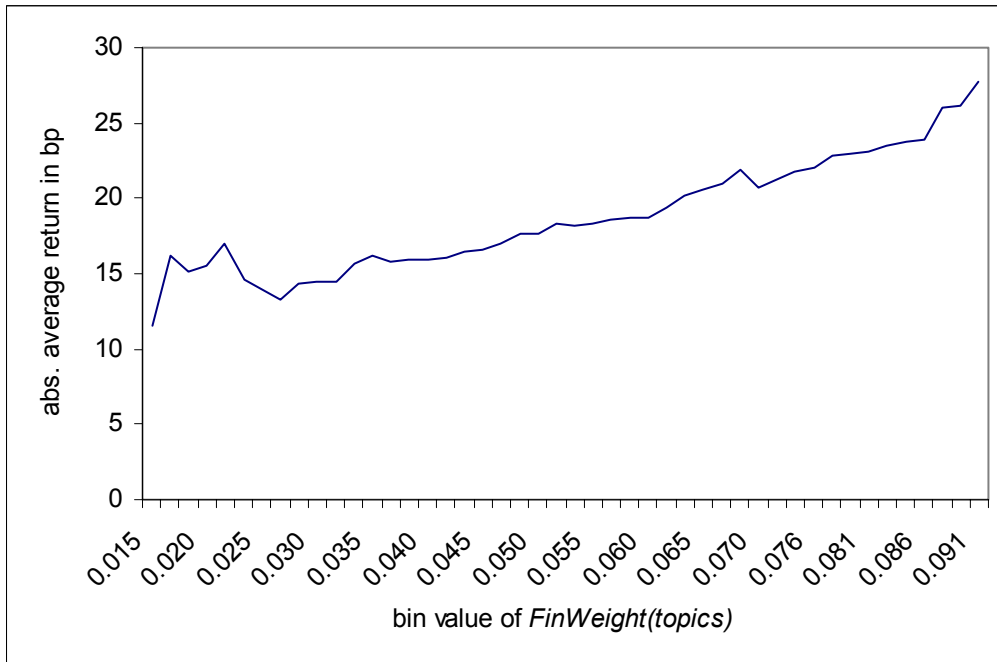


Figure 5.2 Average absolute returns of S&P vs. *FinWeight(topics)*, in-sample testing of news of 2006-2007

From Figures 5.1-5.2 we see that testing results of 2003-2004 are less stable than the results of the 2006-2007 data; the moves of the later period are growing steadier.

On the other hand, the keywords testing on 2003-2004 period produces more stable results than the testing of 2006-2007 period; this can be explained by the fact that keyword 2006-2008 were used to generate keywords for 2003-2005 (refer Appendix B for details).

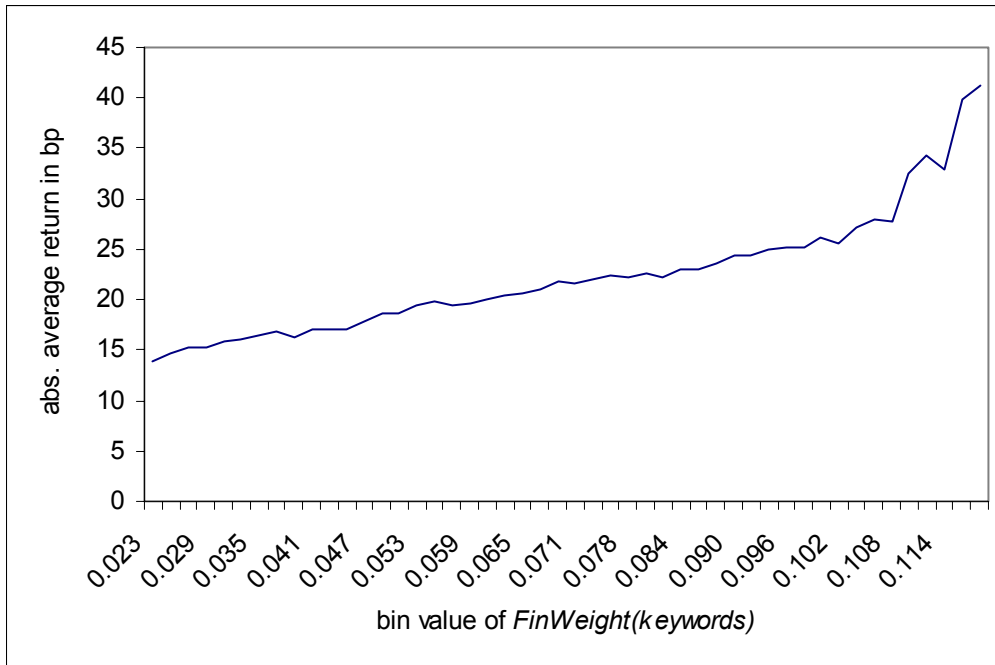


Figure 5.3 Average absolute returns of S&P vs. *FinWeight*(keywords), in-sample testing of news of 2003-2004

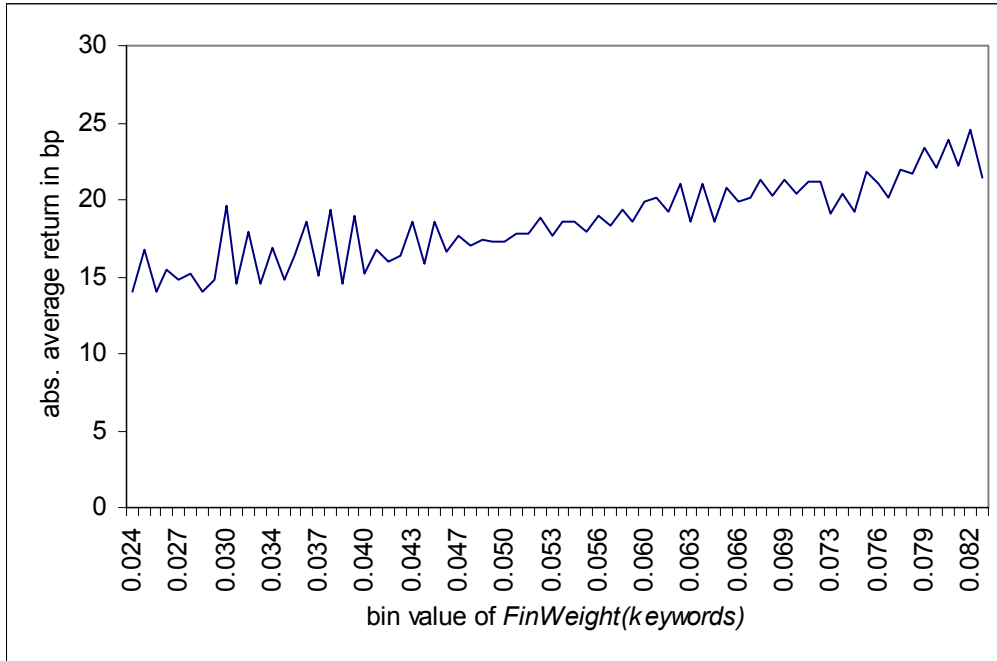


Figure 5.4 Average absolute returns of S&P vs. *FinWeight*(keywords), in-sample testing of news of 2006-2007

Combining the financial weights of topics and keywords gives similar performance of the financial indicator in both training periods, Figures 5.5-5.6.

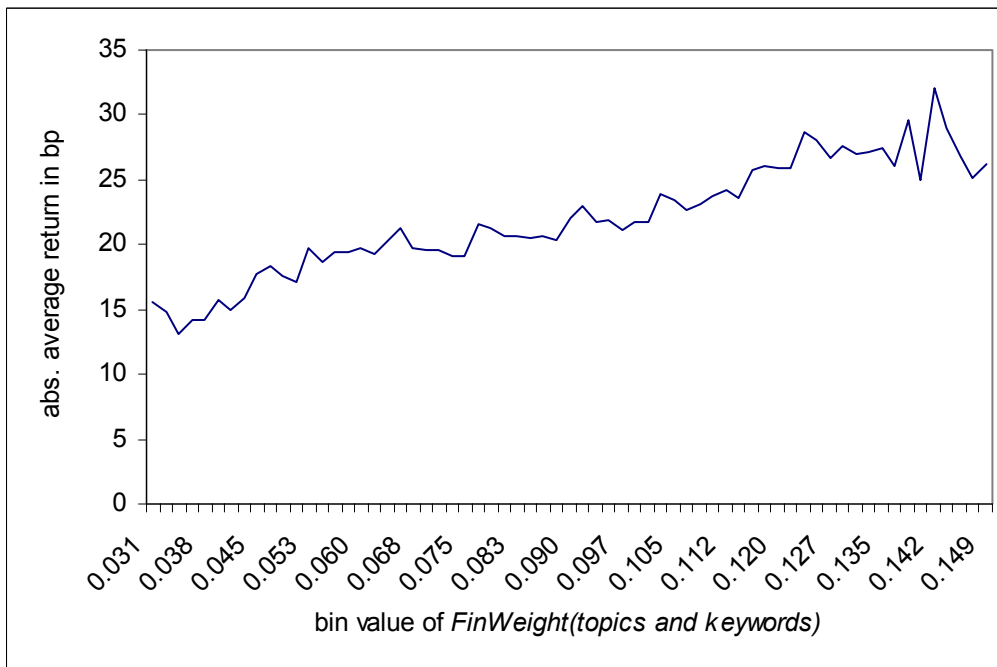


Figure 5.5 Average absolute returns of S&P vs. *FinWeight*(topics and keywords), in-sample testing of news of 2003-2004

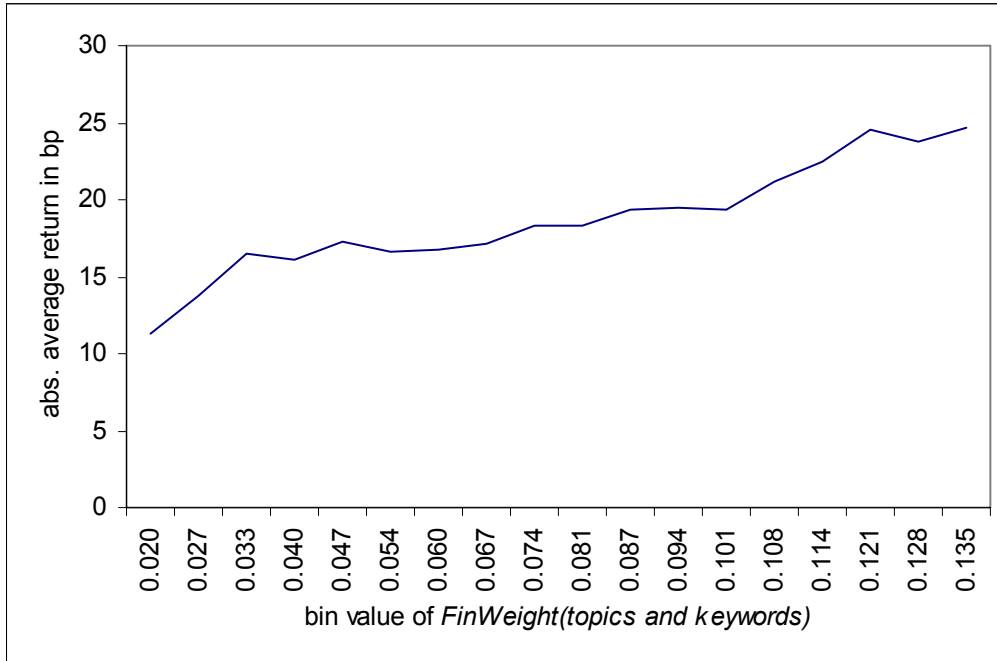


Figure 5.6 Average absolute returns of S&P vs. *FinWeight*(topics and keywords), in-sample testing of news of 2006-2007

5.3 Out-of-sample Testing of the High Level Based Indicators

Figures 5.7-5.12 show the results obtained from the out-of-sample testing. For both periods of testing on 2005 and 2008 years, *FinWeight*(topics/keywords) does not perform well; the moves are irregular and do not conform with the trends of the in-sample testing.

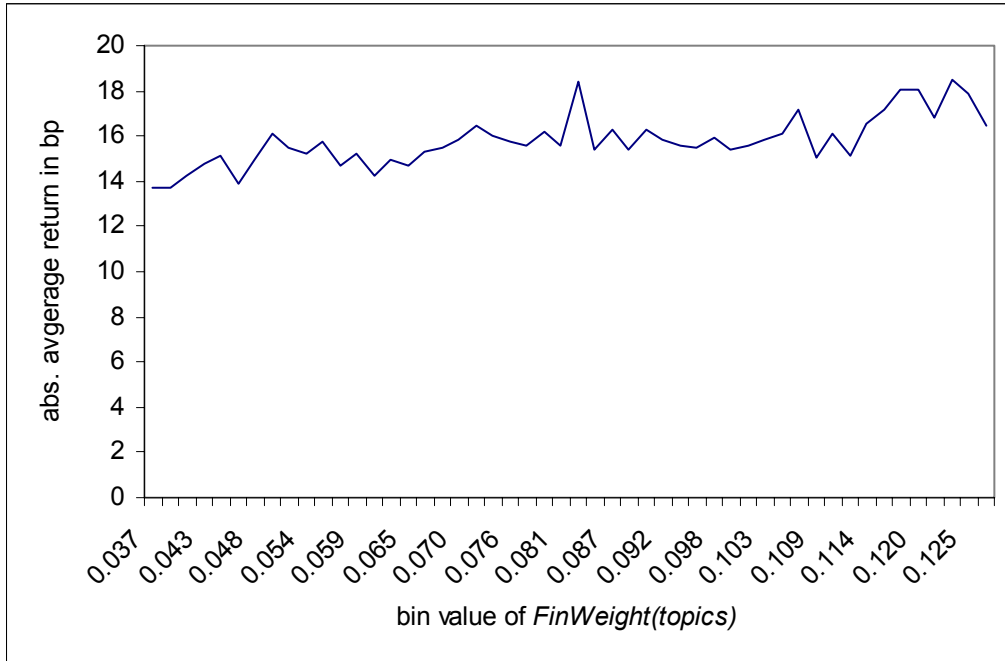


Figure 5.7 Average absolute returns of S&P vs. *FinWeight(topics)*, out-of-sample testing of news of 2005

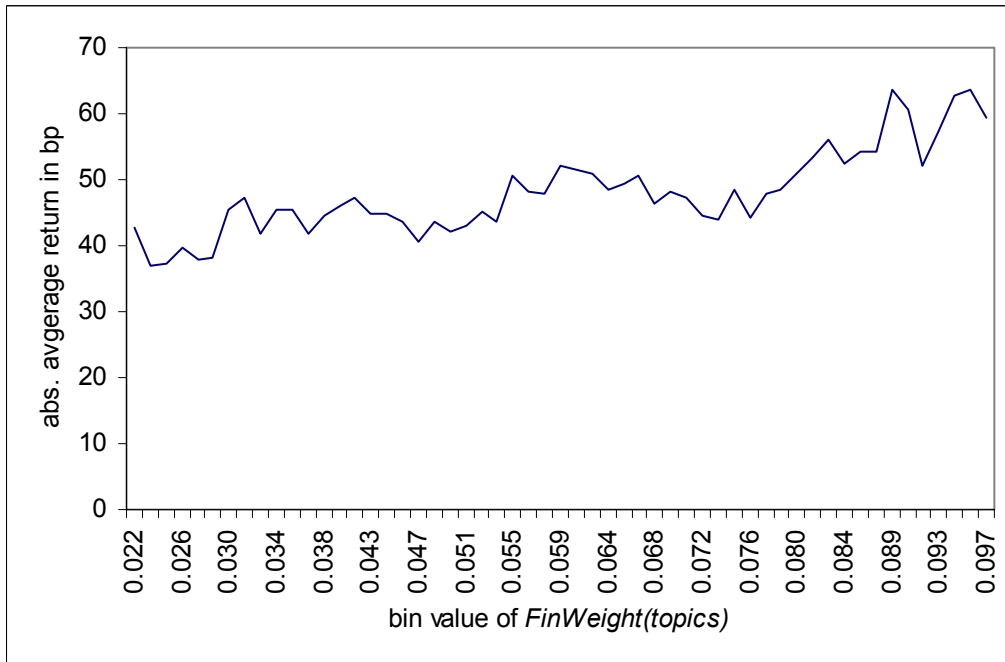


Figure 5.8 Average absolute returns of S&P vs. *FinWeight(topics)*, out-of-sample testing of news of 2008

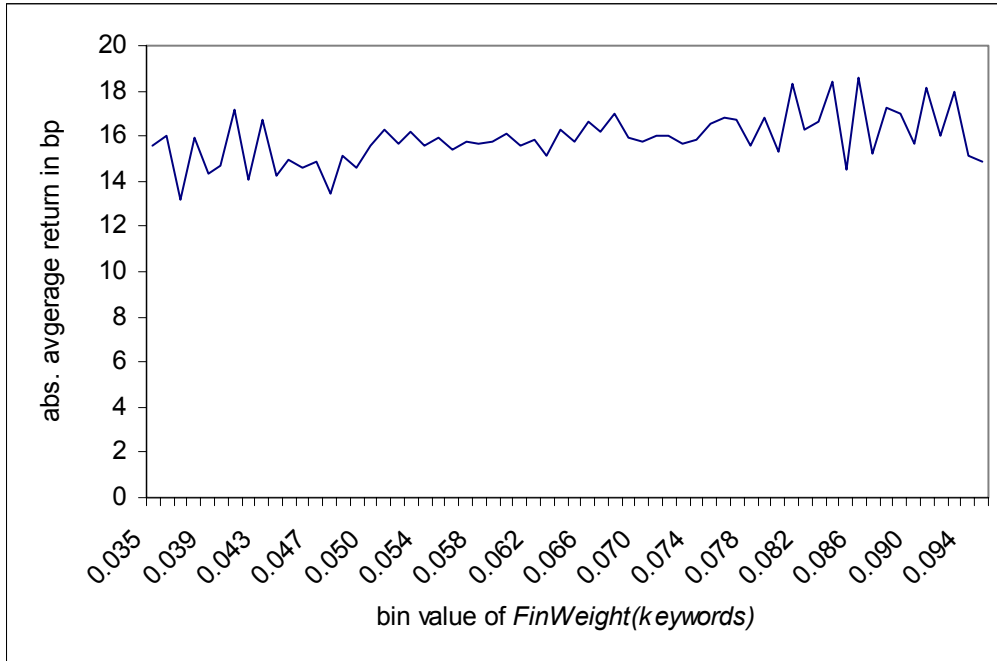


Figure 5.9 Average absolute returns of S&P vs. *FinWeight(keywords)*, out-of-sample testing of news of 2005

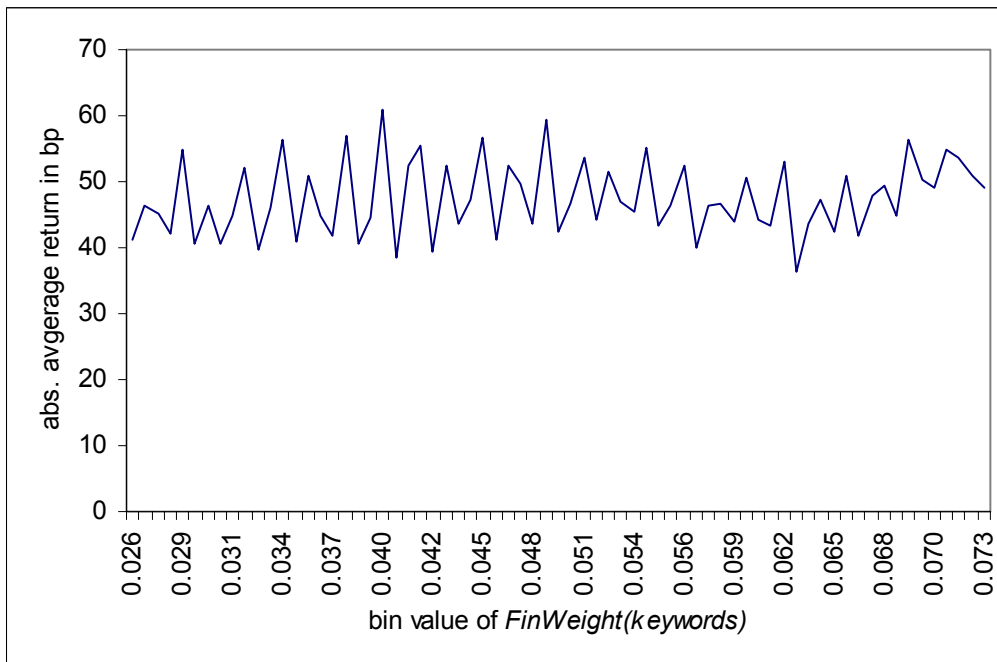


Figure 5.10 Average absolute returns of S&P vs. *FinWeight(keywords)*, out-of-sample testing of news of 2008

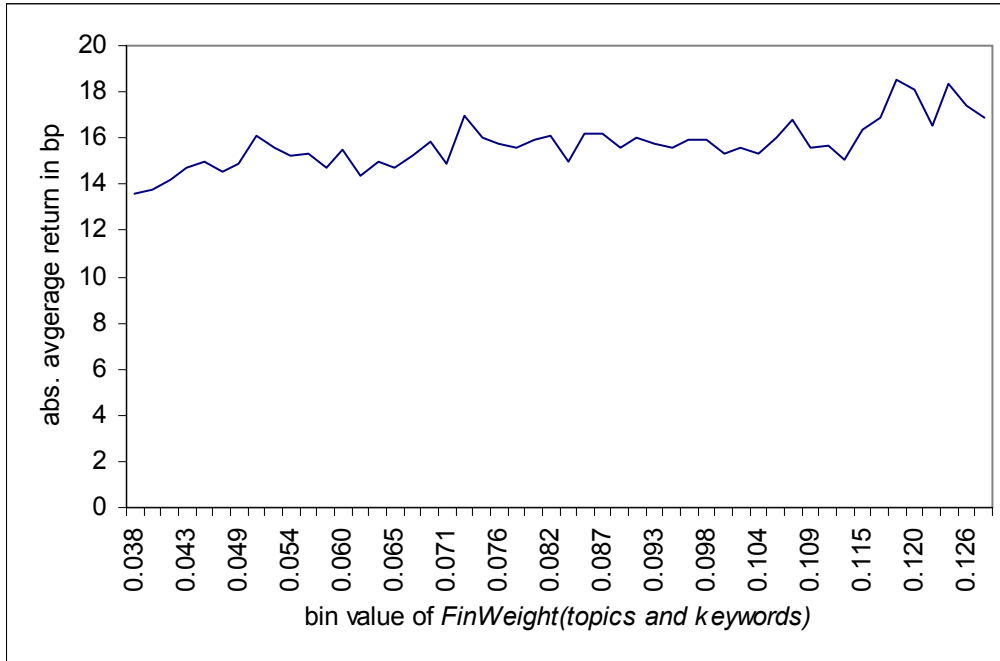


Figure 5.11 Average absolute returns of S&P vs. *FinWeight*(topics and keywords), out-of-sample testing of news of 2005

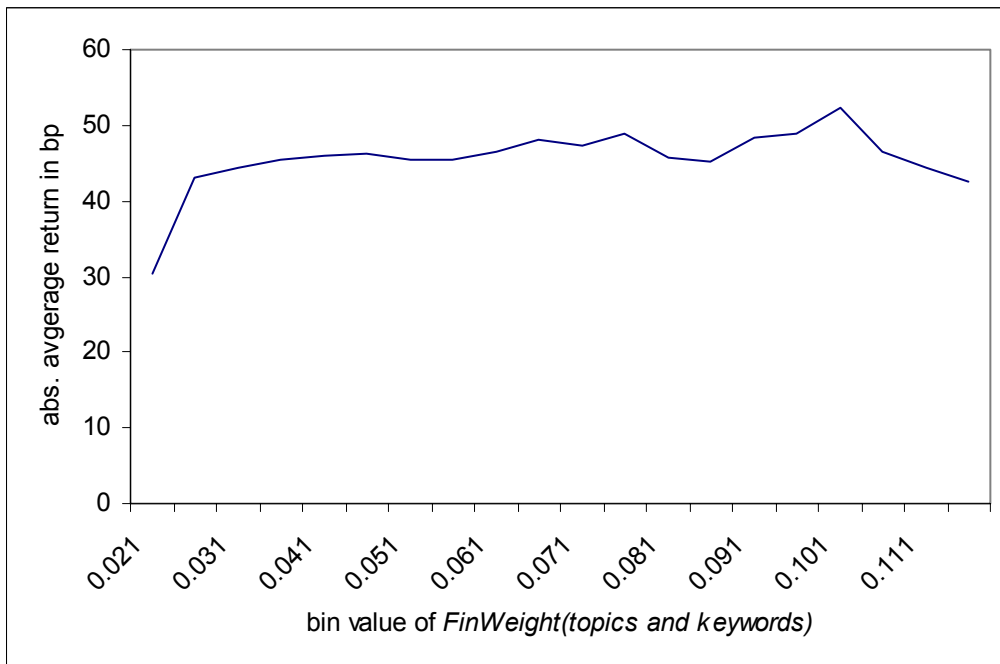


Figure 5.12 Average absolute returns of S&P vs. *FinWeight*(topics and keywords), out-of-sample testing of news of 2008

Concluding the results presented above we can say that topics or/and keywords should not be taken alone for the volatility prediction. Indeed, the number of the used

topics (keywords) is small, and if they are distributed unevenly on the whole data set, the relevant topic (keyword) could not be so easily determined.

6. Predicting Short Term Intraday Volatility

By observing the informative and financial indicators on the testing sets (Chapters 3-4) we can clearly see that $InformS_4(news)$, $FinWeight_1(news)$ and $FinWeight_2(news)$ are dependent on the index price moves. Hence, we can use them to predict the short term intraday volatility. First, we divide the whole news data of 2003-2008 into the sliding windows; each sliding window represent 2 years of the training data and next 6 months of the testing data. In this way we get eight frames producing eight 6-month testing data points. Next, we combine these points from all the windows into one data set to build a general diagram of the index price dependency to a particular indicator. In this chapter we present $FinWeight_1(news)$ and $FinWeight_2(news)$ for the analysis.

6.1 General Description of the Sliding Windows' Processing

To build the volatility predicting tool we use the following data processing steps performed on all the sliding windows of the whole data set.

(1) For each window W :

(1.1) Obtain financial weights from the training data (2 years);

(1.2) Obtain from last n months of the training data ($n = 6$):

(a) $Avg(I_{train})$ and $Std(I_{train})$;

(b) $Avg(M_{train})$ and $Std(M_{train})$,

where $Avg(I_{train})$ and $Std(I_{train})$ are the average and standard deviation of a particular indicator, and $Avg(M_{train})$ and $Std(M_{train})$ are the average and standard deviation of absolute returns (in basis points), respectively.

(1.3) For each data point in the testing data compute:

$$(a) \hat{I} = \frac{I_{test} - Avg(I_{train})}{Std(I_{train})};$$

$$(b) \hat{M} = \frac{M_{test} - Avg(M_{train})}{Std(M_{train})},$$

where I_{test} and M_{test} are the current values of each data point (for details see Chapter 2.2).

(2) Combine all the transformed data points into one set; so each data point represents two values \hat{I} and \hat{M} .

Since the volatility may vary in each window period, we should transform the values of the data points into \hat{I} and \hat{M} as in (1.3); this allows us to resolve the volatility trend issue when the average volatility differs from one sliding window period to another.

Again, we use 50% of the words with highest financial weight in the processing and only US-related data is considered.

Furthermore, the out-of-sample testing results for each frame and for all combined frames are conveyed in the next Chapters 6.2 and 6.3; the bin value of \hat{I} is on x-axis and average of \hat{M} values is on y-axis. In the figures below we notate \hat{I} as I_1 and I_2 (computed using $FinWeight_1(news)$ and $FinWeight_2(news)$, respectively) and \hat{M} as M .

6.2 Out-of-sample Testing for Each Sliding Window

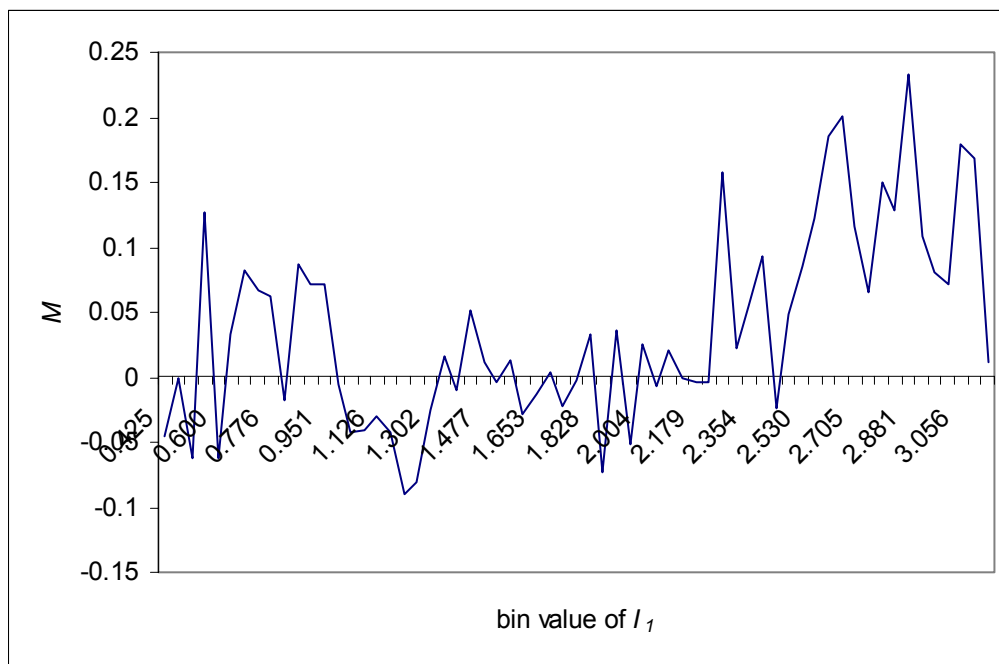


Figure 6.1 \hat{M} vs. \hat{I}_1 , out-of-sample testing of news, [2005-01-01, 2005-07-01)

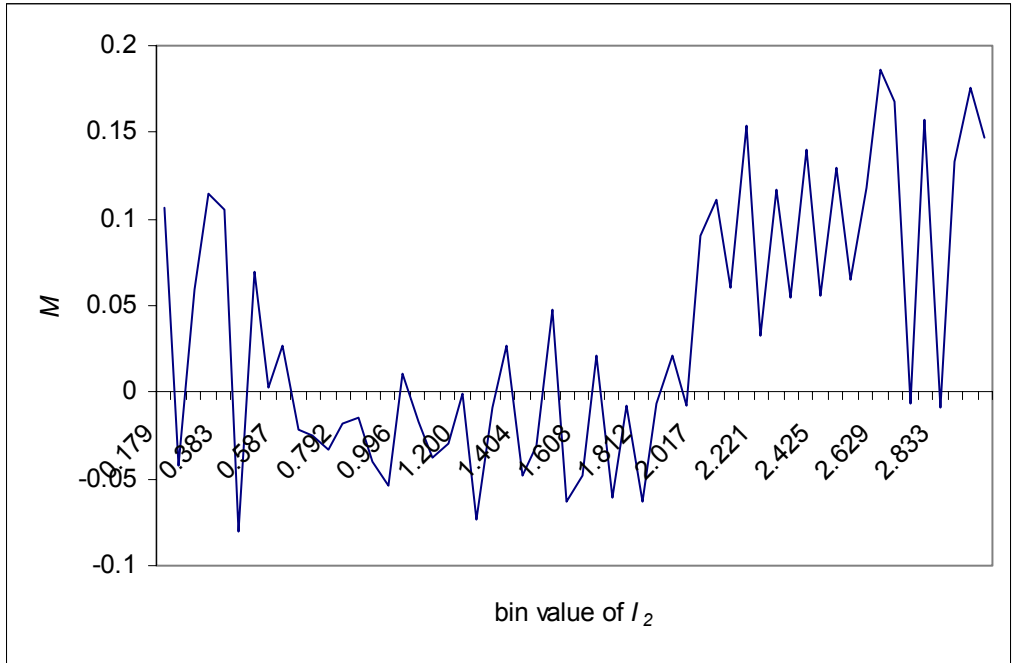


Figure 6.2 \hat{M} vs. \hat{I}_2 , out-of-sample testing of news, [2005-01-01, 2005-07-01]

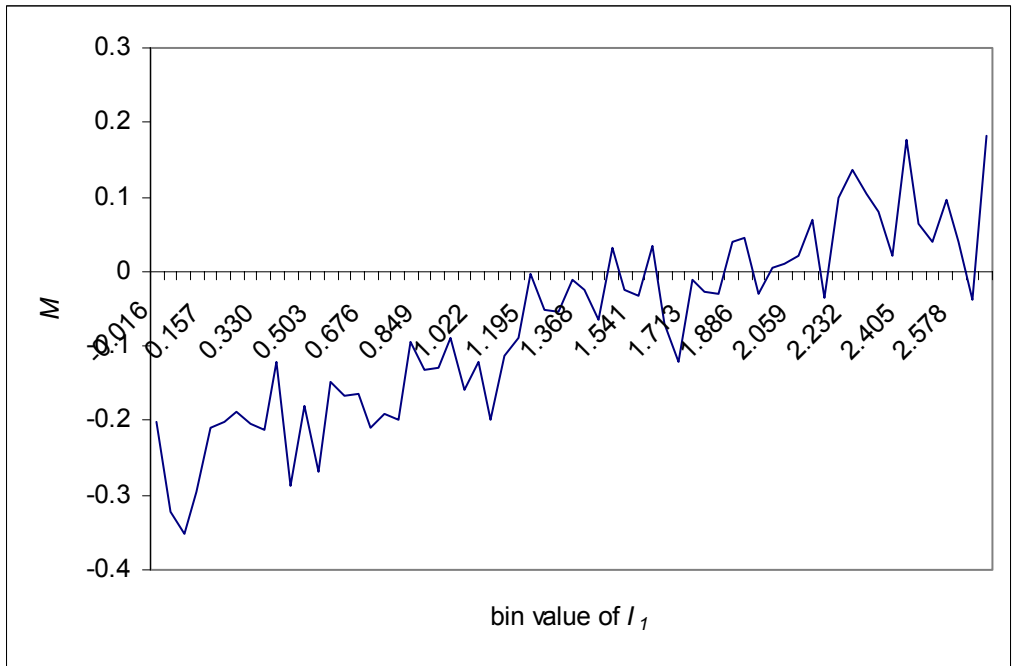


Figure 6.3 \hat{M} vs. \hat{I}_1 , out-of-sample testing of news, [2005-07-01, 2006-01-01]

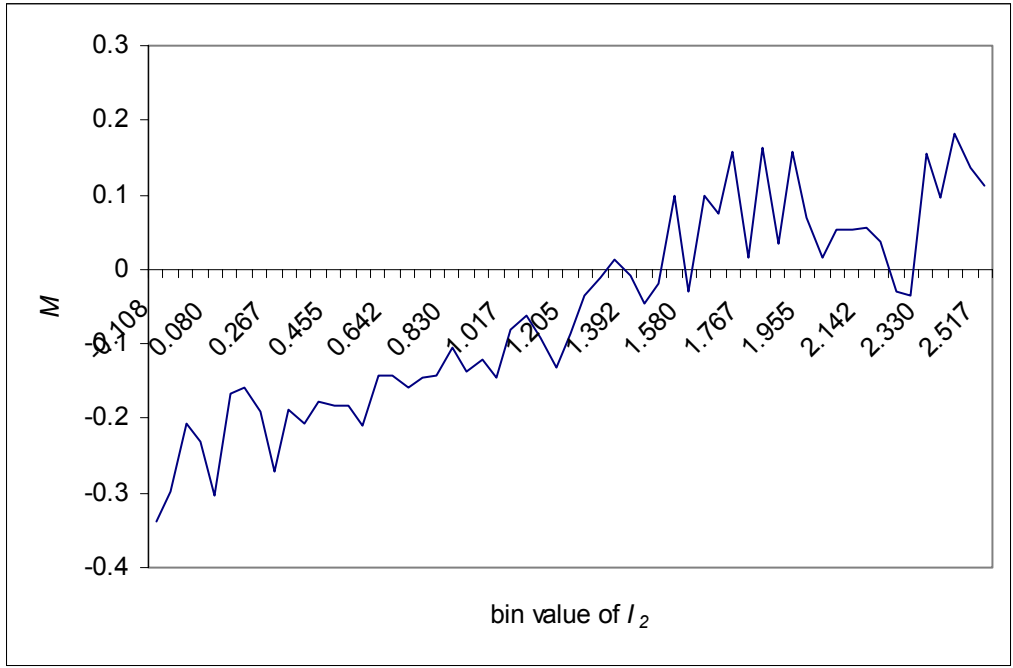


Figure 6.4 \hat{M} vs. \hat{I}_2 , out-of-sample testing of news, [2005-07-01, 2006-01-01)

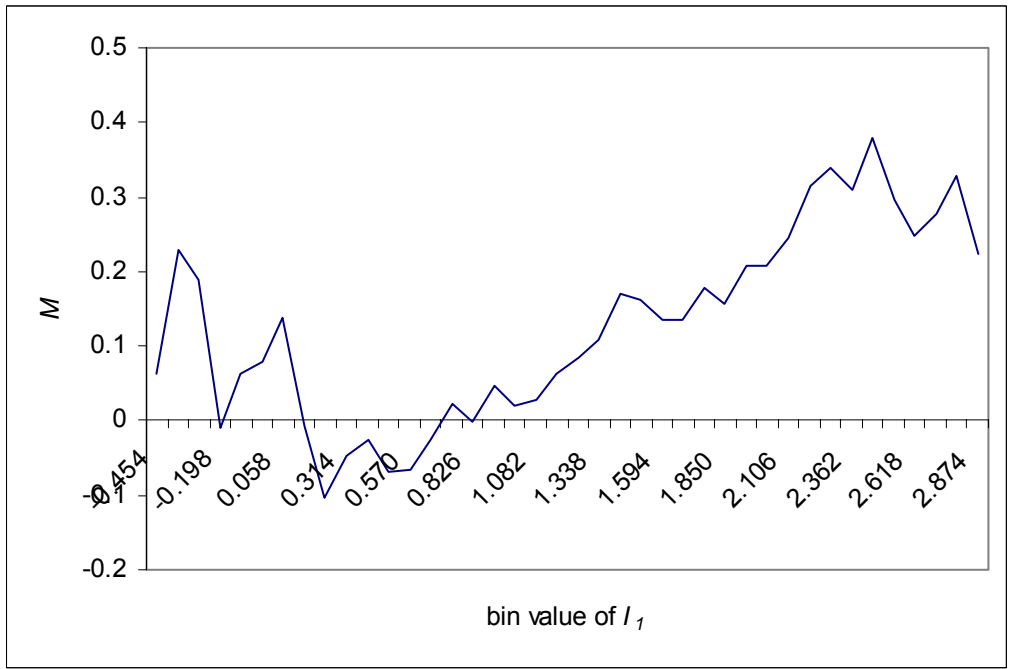


Figure 6.5 \hat{M} vs. \hat{I}_1 , out-of-sample testing of news, [2006-01-01, 2006-07-01)

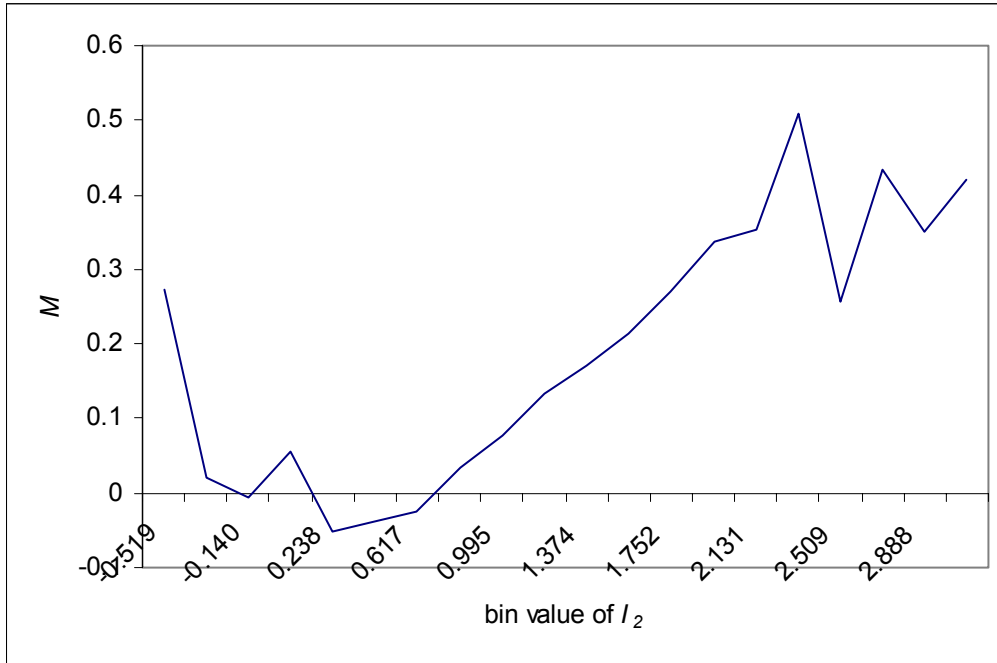


Figure 6.6 \hat{M} vs. \hat{I}_2 , out-of-sample testing of news, [2006-01-01, 2006-07-01)

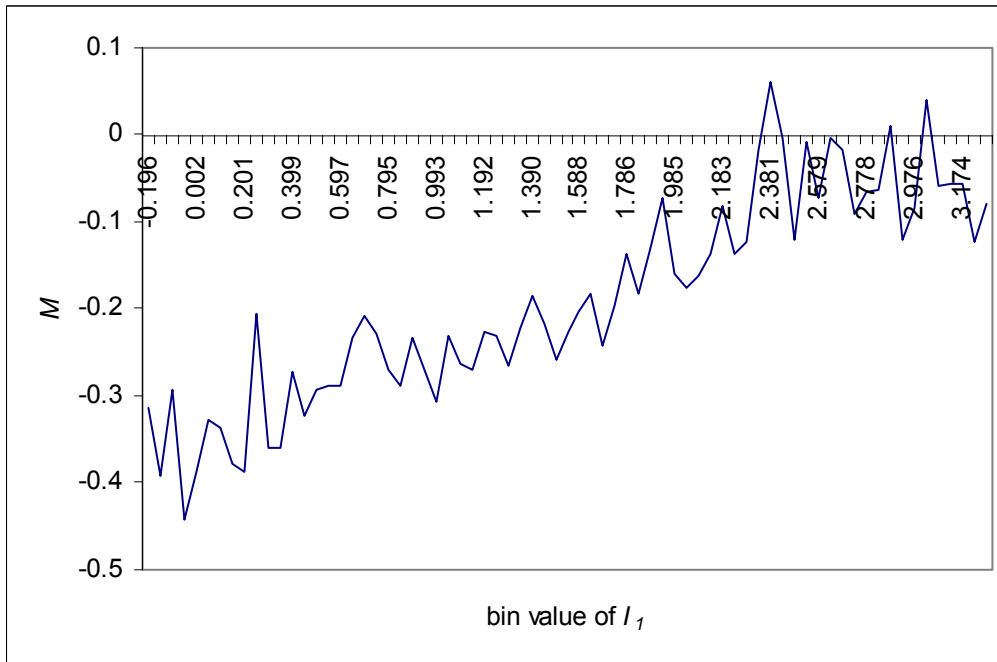
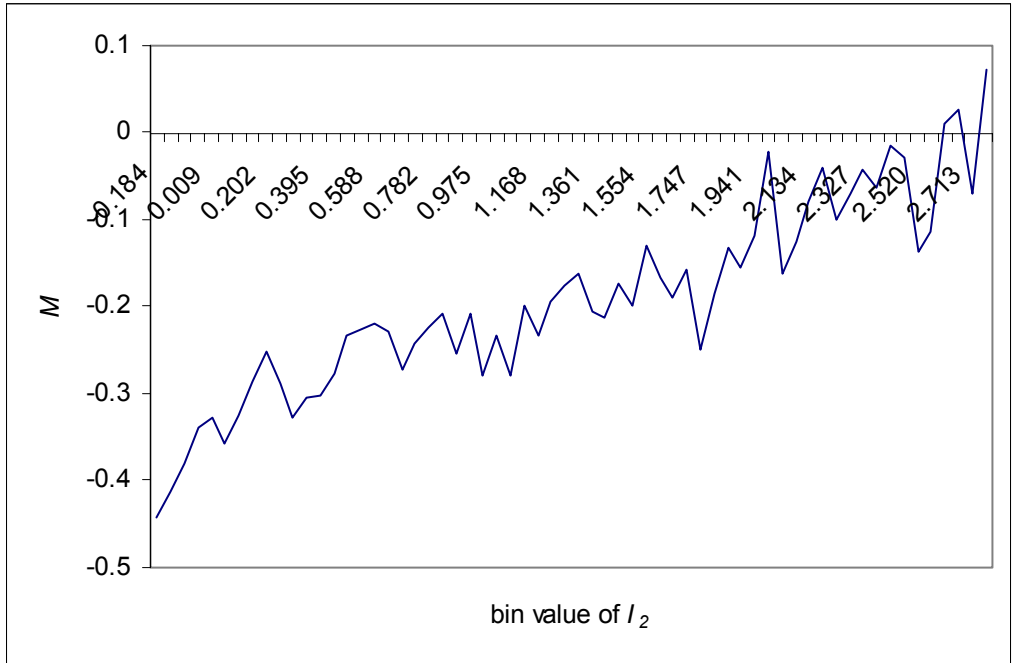


Figure 6.7 \hat{M} vs. \hat{I}_1 , out-of-sample testing of news, [2006-07-01, 2007-01-01)



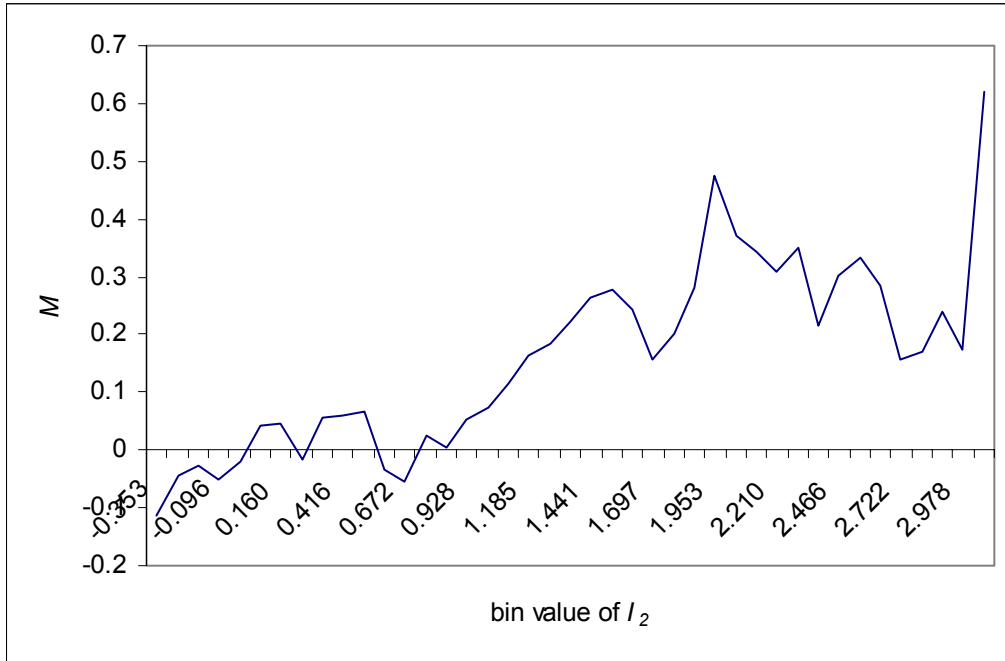


Figure 6.10 \hat{M} vs. \hat{I}_2 , out-of-sample testing of news, [2007-01-01, 2007-07-01)

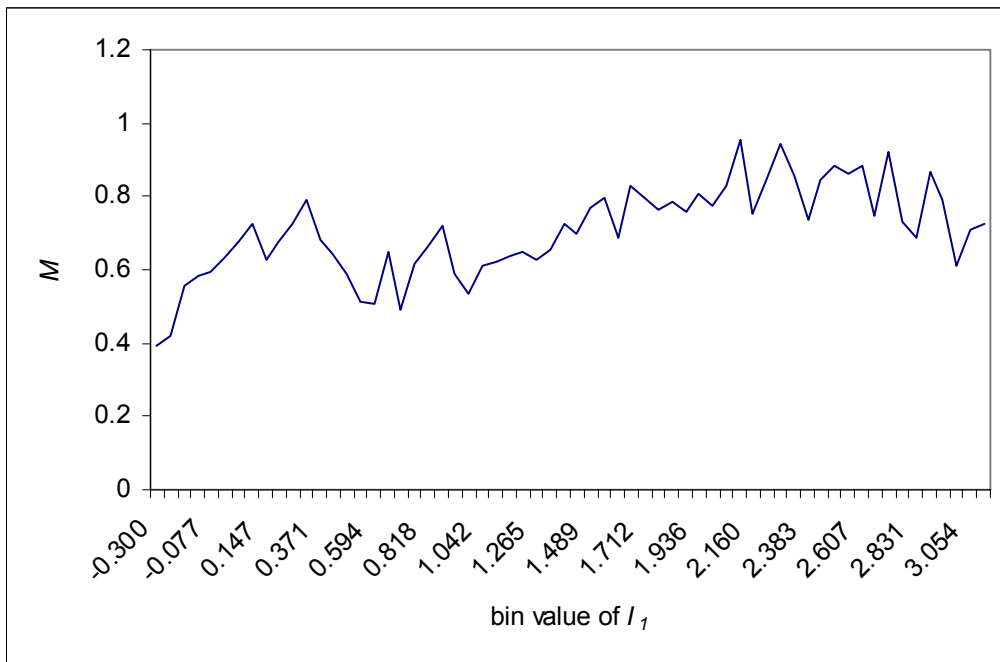


Figure 6.11 \hat{M} vs. \hat{I}_1 , out-of-sample testing of news, [2007-07-01, 2008-01-01)

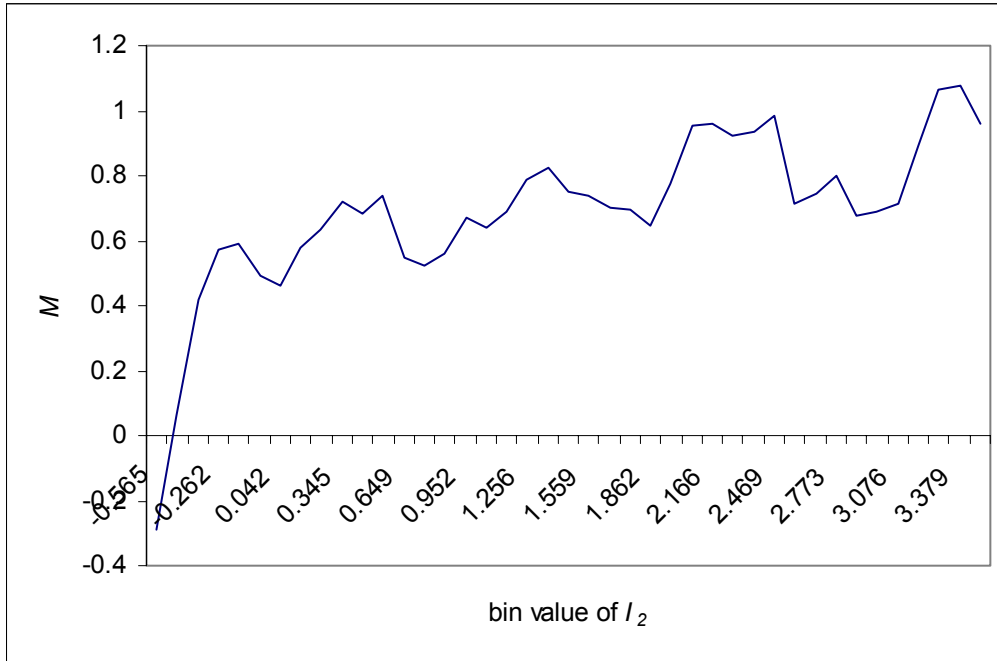


Figure 6.12 \hat{M} vs. \hat{I}_2 , out-of-sample testing of news, [2007-07-01, 2008-01-01)

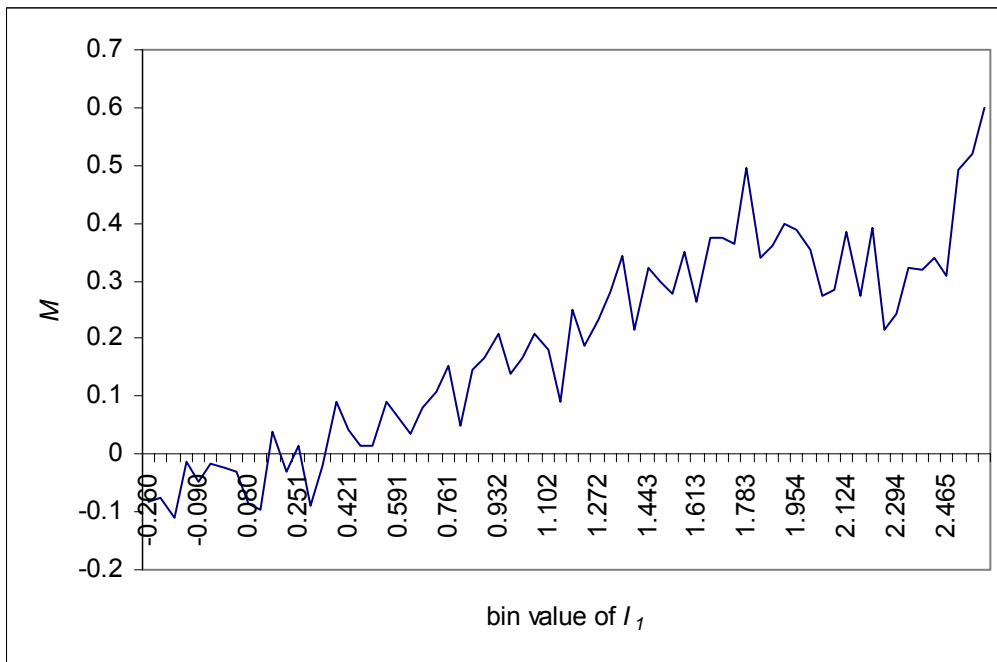


Figure 6.13 \hat{M} vs. \hat{I}_1 , out-of-sample testing of news, [2008-01-01, 2008-07-01)

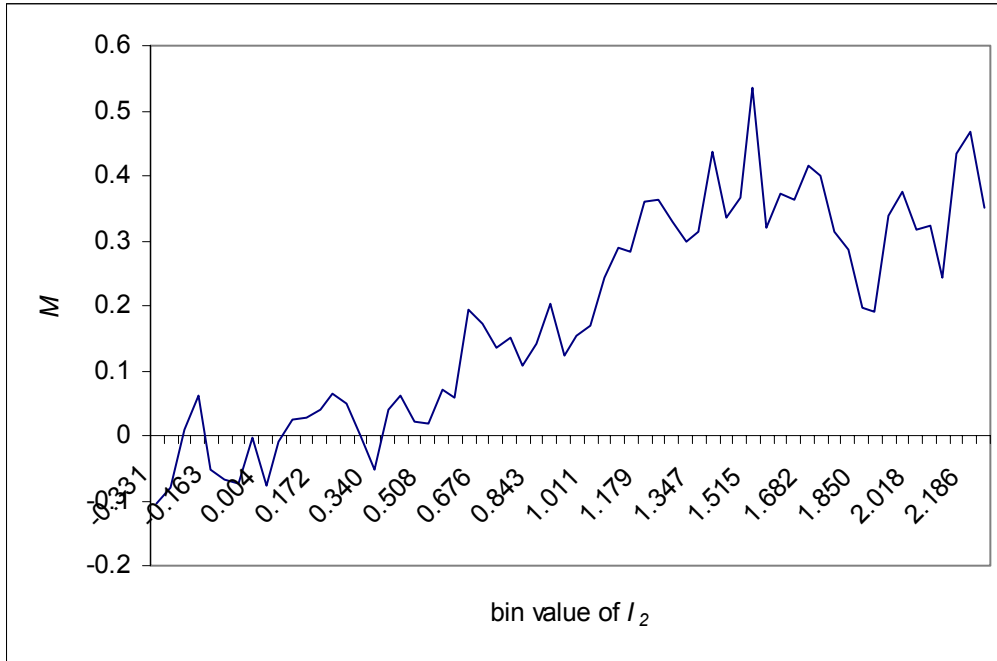


Figure 6.14 \hat{M} vs. \hat{I}_2 , out-of-sample testing of news, [2008-01-01, 2008-07-01)

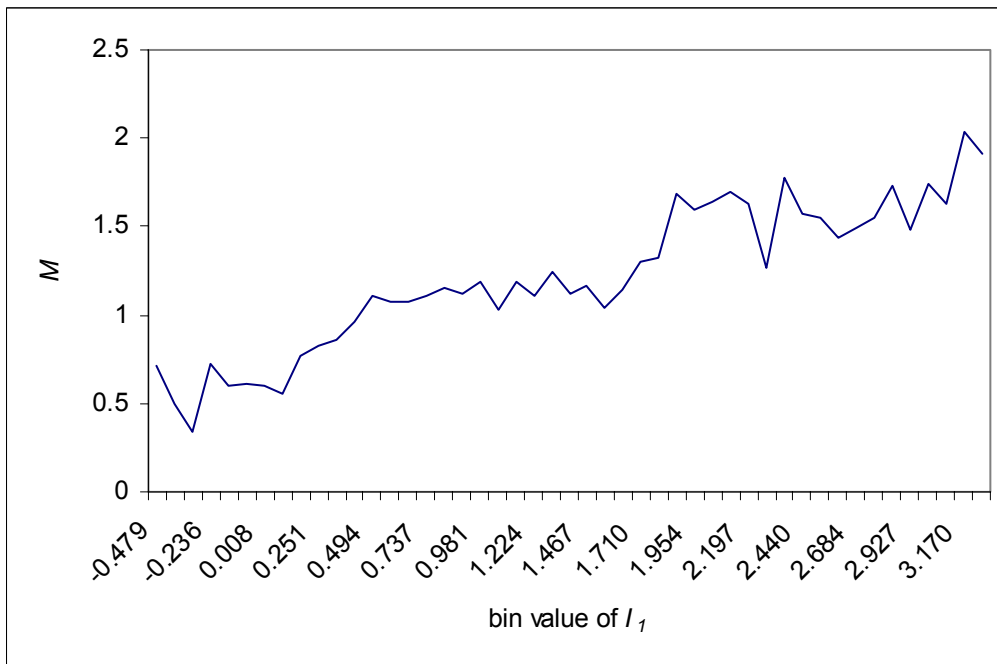


Figure 6.15 \hat{M} vs. \hat{I}_1 , out-of-sample testing of news, [2008-07-01, 2009-01-01)

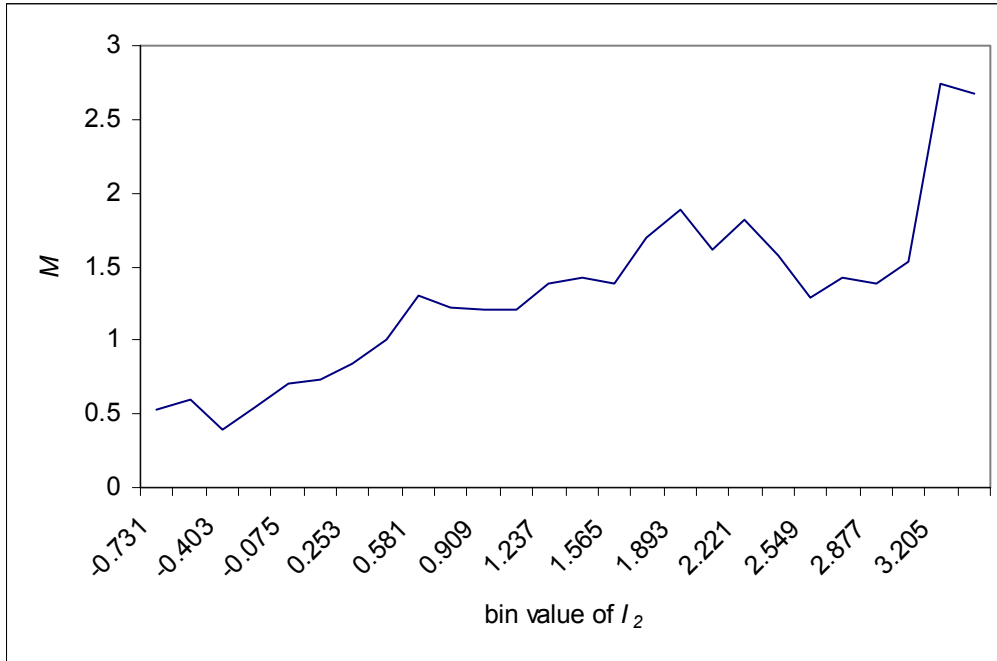


Figure 6.16 \hat{M} vs. \hat{I}_2 , out-of-sample testing of news, [2008-07-01, 2009-01-01]

6.3 Out-of-sample Testing for all Windows' Data Combined

Combining all testing data points from the sliding windows presented above, we build a general diagram of the index price dependency to \hat{I}_1 and \hat{I}_2 in Figures 6.17 and 6.18, respectively (the number of data points at each bin can be found in Appendix E). We also fit a line using OSL linear regression model for each of our predictors.

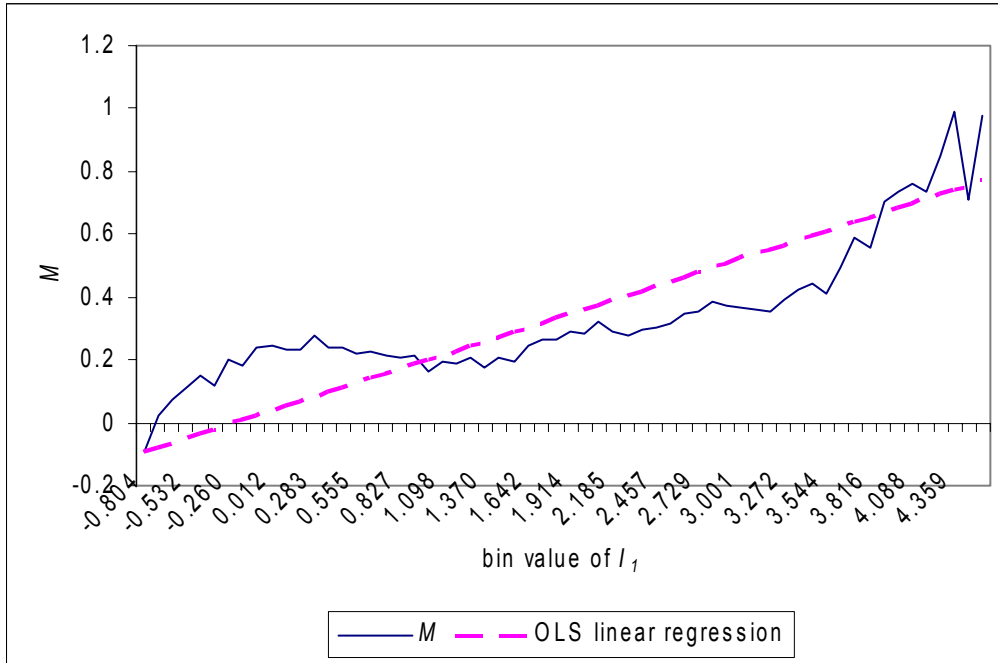


Figure 6.17 \hat{M} vs. \hat{I}_1 , out-of-sample testing of news, all windows combined

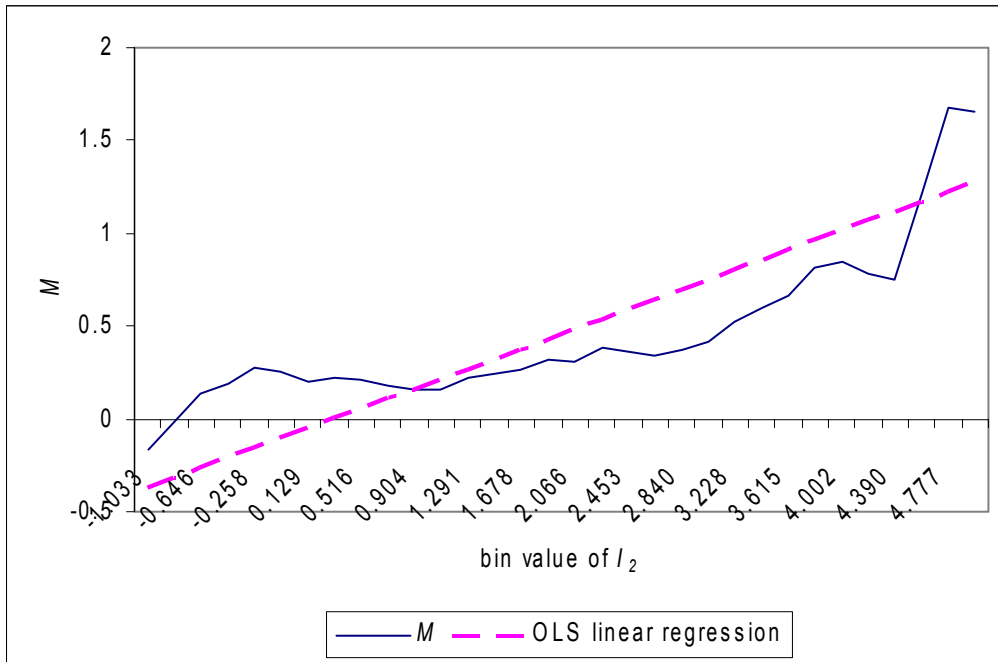


Figure 6.18 \hat{M} vs. \hat{I}_2 , out-of-sample testing of news, all windows combined

6.3.1 Additional Experiments Using 25% of the Words

All results shown so far are based on 50% of the words with highest financial weight. We do additional experiments using 25% of the words and present the them in Figures 6.19 and 6.20 (the number of data points at each bin can be found in Appendix E).

As we can see the regression line fits very well in the predictive models with 25% of the words. In particular, the results for \hat{I}_1 show a closer fit comparing to \hat{I}_2 .

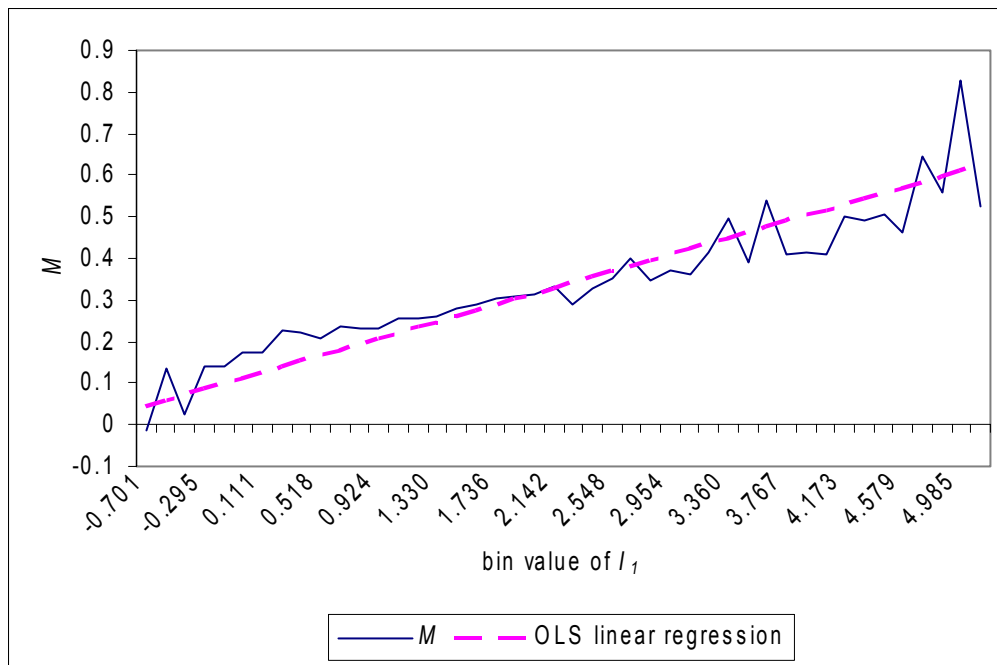


Figure 6.19 \hat{M} vs. \hat{I}_1 , out-of-sample testing of news, all windows combined, 25% of the words

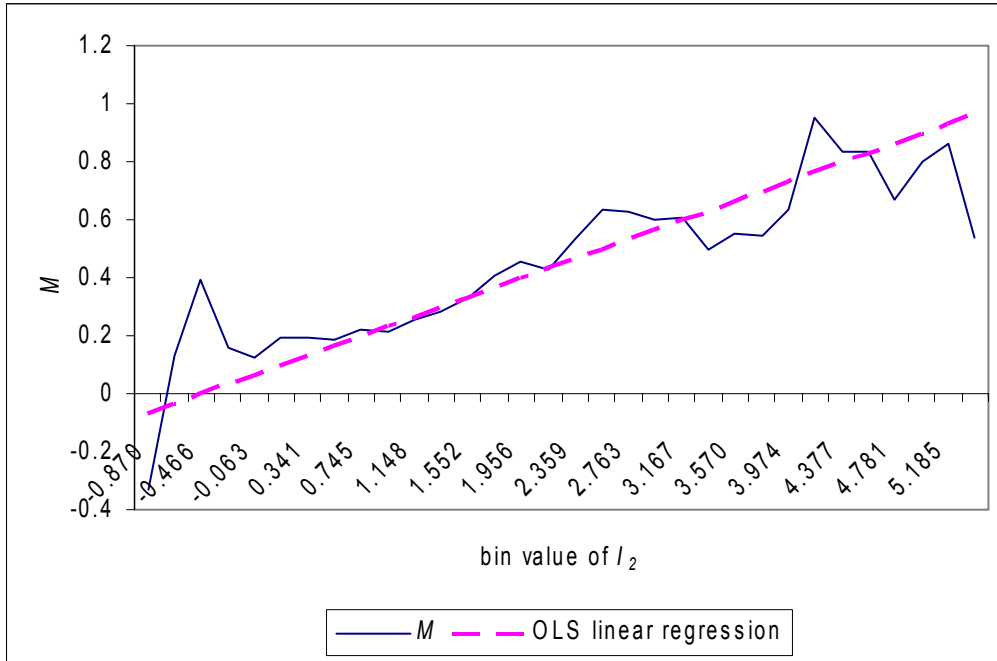


Figure 6.20 \hat{M} vs. \hat{I}_2 , out-of-sample testing of news, all windows combined, 25% of the words

7. Discussions and Conclusion

In this thesis we have presented the analysis of Reuters news data with respect to the absolute moves of S&P 500.

First, preliminary data processing is performed. The notion of informativeness is introduced for duplicate detection as well as for building the indicators.

Second, the performance of the proposed informative indicators is examined. As a result, we have found that $InformS_4(news)$ indicator performs best among the others, in terms of the average absolute returns.

Third, financial indicators are analyzed based on the financial weights obtained from the training data. Similar to $InformS_4(news)$, $FinWeight_1(news)$ and $FinWeight_2(news)$ have also shown good results during in-sample as well as out-of-sample testing. Consequently, we have presented a general volatility predictive model by combining the testing data points of the sliding windows into one data set. Moreover, we have found that using 25% of the words with highest financial weight produces a better fit of the OSL regression line to the predictive models, specifically, of \hat{I}_1 .

Further, financial weights of topics and keywords are examined using the same procedure of the financial weights analysis performed on the stories' text. The results produced from the out-of-sample testing for both topics and keywords have not shown any trend of the market index moves, so we come to the conclusion that at this point they cannot be used in the volatility prediction.

Future work may consist of volatility prediction using informative indicators, and development of news trading strategy.

LITERATURE CITED

- [1] Reuters Limited (March 2008). *User Guide v2.3*. Reuters NewsScope Archive v2.0.
- [2] Robert P. Schumaker, Hsinchun Chen (2009). *Textual analysis of stock market prediction using breaking financial news: The AZFin text system*. ACM Trans. Inf. Syst. 27(2).
- [3] Thomas, J.D. and K. Sycara (2002). *Integrating Genetic Algorithms and Text Learning for Financial Prediction*. Genetic and Evolutionary Computation Conference (GECCO), Las Vegas, NV.
- [4] Mittermayer, M.-A. (2004). *Forecasting Intraday Stock Price Trends with Text Mining Techniques*. Proceedings of the 37th Hawaii International Conference on Social Systems, Hawaii.
- [5] G. Gidófalvi and C. Elkan (2003). *Using News Articles to Predict Stock Price Movements. Technical Report*. Department of Computer Science and Engineering, University of California, San Diego.
- [6] R. P. Schumaker and H. Chen (2006). *Textual Analysis of Stock Market Prediction Using Financial News Articles*. Proceedings of the 12th Americas Conference on Information Systems, paper 185, Acapulco, Guerrero, Mexico.
- [7] Kloptchenko, A., T. Eklund, et al. (2004). *Combining Data and Text Mining Techniques for Analysing Financial Reports*. Intelligent Systems in Accounting, Finance & Management 12(1): 29-41.
- [8] Seo, Y.-W., J. Giampapa, et al. (2002). *Text Classification for Intelligent Portfolio Management*. Robotics Institute, Canegie Mellon University.
- [9] Fung, G.P.C., J.X. Yu, et al. (2002). *News Sensitive Stock Trend Prediction*. Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD), Taipei, Taiwan.
- [10] Victor Lavrenko, Matt Schmill, Dawn Lawrie, Paul Ogilvie, David Jensen, and James Allan (2000). *Language Models for Financial News Recommendation*. Proceedings of the Workshop on Text Mining at the Ninth International Conference on Information Knowledge Management (CIKM 2000), pp. 389-396.
- [11] Michael Lewis. *Jonathan Lebed: Stock manipulator, S.E.C. nemesis – and 15*. New York Times Magazine, February 2001.

APPENDIX A Processing of Corrected (Updated) News

A trader makes a buy/sell decision after original news is published, so that decision cannot be changed when the news is suddenly corrected; due to this reason, we do not include the corrected news into our analysis. They are removed using the correction rules as follows:

- a) When a story body needs to be corrected “STORY_TAKE_OVERWRITE” event is transmitted with the corrected body text: so, we remove all subsequent “STORY_TAKE_OVERWRITE” events (using unique story index they share).
- b) When a story headline needs to be replaced and the body text appended, “HEADLINE” event is transmitted with the corrected headline text and “HEADLINE_SUBTYPE” attribute equaled to 4; so, we remove this event (the body text is handled by (a)) .
- c) When both the headline and body need to be corrected, “HEADLINE” event is transmitted with the corrected headline text, “HEADLINE_SUBTYPE” attribute equaled to 5, and new unique story index; the body is replaced subsequently by “STORY_TAKE_OVERWRITE” event having the new unique story index. These two events can arrive in either order – corrected “HEADLINE” followed by “STORY_TAKE_OVERWRITE”, or vice-versa; the 1st case is handled by taking the unique story index and deleting all other events with this index; the 2nd – by removing all subsequent “STORY_TAKE_OVERWRITE” events as in (a), and then removing “HEADLINE” event by using its new unique story index.
- d) When the alert needs to be corrected, “ALERT” event is transmitted with the corrected alert text containing headline tag “CORRECTED” or “CORRECTION” and the new unique story index; all other events associated with the old alert are repeated with the new unique story index of the corrected alert. We remove all corrected alerts and repeated events associated with them (by using new unique index).
- e) When additional meta data needs to be added (for example, additional topic code), “HEADLINE” event is transmitted with the updated code attributes and “HEADLINE_SUBTYPE” attribute equaled to 3; this event is redundant and also removed.

f) When additional text needs to be appended, “STORY_TAKE_APPEND” event is transmitted; this event is removed.

g) When news is no longer valid and needs to be ignored, “DELETE” event is transmitted with the unique story index equaled to the index of the invalid news; majority of the news with this event occur when they need to be corrected or changed, - they share “PNAC” attribute with the original news but contain different unique story indexes. The corrected news are deleted transmitted after “DELETE” event and identified by the “PNAC” of the original story and different index (“DELETE” event is also removed).


```

1  function GetKeywords (& $text, $strSearch = "Keywords:")
2  //for the keywords used after March 2006
3  {
4      $pos = strrpos($text, $strSearch);//finds the last occur. of the $strSearch
5      if($pos !== false)
6      {
7          $strPart = trim(substr($text, $pos + strlen($strSearch)));
8          if($strPart == strtoupper($strPart)) //check for all uppercase letters
9              return trim(str_replace('/', ' ', $strPart)); //replace '/' by space
10         else return false;
11     }
12     else return false;
13 }

```

Figure B.2 Keywords detection procedure for news published from March, 2006 to present

The keyword detection is performed on data from 2006 to 2008. As a result the number of news with no keywords is as follows:

- (a) 289081 stories with no keywords (out of 638373) in 2006 news;
- (b) 175616 stories with no keywords (out of 683233) in 2007 news;
- (c) 182835 stories with no keywords (out of 667059) in 2008 news.

Also, there are no keywords provided for the data of 2003, 2004 and most of 2005. So we further process all the news (“STORY_TAKE_OVEWRITE” events) with no keywords by matching the story body text with the available keywords of 2006-2008 (up to 3 most frequent keywords matched are assigned for a story). As a result the number of news with no keywords is as follows:

- (a) 9760 stories with no keywords (out of 638373) in 2003 news;
- (b) 11284 stories with no keywords (out of 683233) in 2004 news;
- (c) 12449 stories with no keywords (out of 667059) in 2005 news.
- (a) 9914 stories with no keywords (out of 638373) in 2006 news;
- (b) 9597 stories with no keywords (out of 683233) in 2007 news;
- (c) 8370 stories with no keywords (out of 667059) in 2008 news.

B.2 Text Conversion to One Paragraph

The text reduction is carried out using the description of news formatting from Reuters User Guide [1]. Text transmitted by “STORY_TAKE_OVERWRITE” event is processed as follows:

- (1) *Advisory line* is deleted if the open round bracket “(“ is the first character (after text trimming), that is removing all the following characters up to the first occurrence of the closing bracket “)”;
- (2) *Byline* is deleted if string ‘By’ is found in the beginning of the new text;
- (3) Dash character “-“ is searched in the first line of the new text: if found, all the following characters (up to the first occurrence of 4 spaces) are taken as a first paragraph of the story; otherwise up to 80 strings (delimited with a space) are taken as the first paragraph.

APPENDIX C Case Modification

The following stages describe the case modification procedure performed on the news data:

(1) Processing of the headlines and bodies in “STORY_TAKE_OVERWRITE” events:

For each event:

1. Modify the case of all words in the headline:

For each word:

a) Remove [‘s] from the end of the word;

b) If 1st letter of the word is in uppercase:

Search this word in the headline (as a substring):

If found, capitalize all the characters of the word;

Else, lowercase all the characters of the word.

2. Modify the case of all words in the body:

For each word:

a) Remove [‘s] from the end of the word;

b) If 1st letter of the word is in uppercase:

Capitalize all the characters of the word;

Else, lowercase all the characters of the word.

(2) Processing of the headlines in “HEADLINE” events:

For each event:

1. Modify case of all words in the headline:

For each word:

a) Remove [‘s] from the end of the word;

b) If 1st letter of the word is in uppercase:

Capitalize all the characters of the word;

Else, lowercase all the characters of the word.

(3) Processing of the headlines in “ALERT” events:

For each event:

1. Remove [‘S] from the end of each word;

(4) Computing frequency of the words in the headlines and bodies of “STORY_TAKE_OVERWRITE” events;

(5) Processing of the headlines and bodies in all events:

For each event:

Search each word in the list of words and frequencies produced by stage (4):

If found, search for the same word with the opposite case in that list:

If found, compare the frequencies of these words and modify the case of the given word to the one with higher frequency;

Else, leave the case intact;

Else, search the word with the opposite case in the list of words and frequencies produced by stage (4):

If found, modify the case to the opposite;

Else, leave the case intact.

APPENDIX D Similarity

```
1  float Similarity(string desc1, string desc2, map<string, float> &informWords)
2  //compares two strings "desc1" and "desc2",
3  //returns similarity value of the given strings:
4  //the return value is between 0 and 1;
5  //the larger the value the more similar they are;
6  //map "informWords" contains words as keys and informativeness as values
7  //uses explodeStr() to get a vector of strings delimited with spaces
8  {
9      float similarity = 0;          //return value
10     vector<string> interSect;      //intersection of d1 and d2
11     vector<string> d1 = explodeStr(desc1, " ");
12     vector<string> d2 = explodeStr(desc2, " ");
13     set<string> uniqueWordsD1(d1.begin(), d1.end()); //set of words in desc1
14     set<string> uniqueWordsD2(d2.begin(), d2.end()); //set of words in desc2
15     //insert_iterator for intersect:
16     insert_iterator< vector<string> > iterSect(interSect, interSect.begin());
17     set_intersection(uniqueWordsD1.begin(), uniqueWordsD1.end(),
18                     uniqueWordsD2.begin(), uniqueWordsD2.end(), iterSect);
19     //find the freq-cy of each element of the interSect
20     unsigned int freqD1, freqD2, minFreq;
21     float inform;
22     float sum=0;
23     map<string, float>::iterator it; //iterator to the map "informWords"
24     for(unsigned i=0; i< interSect.size(); ++i)
25     {
26         it=informWords.find(interSect[i]);
27         if (it!=informWords.end())
28         {
29             inform = it->second;
30             freqD1 = count(d1.begin(), d1.end(), interSect[i]);
31             freqD2 = count(d2.begin(), d2.end(), interSect[i]);
32             minFreq = min(freqD1, freqD2);
33         }
34         else continue; //in case we don't have the informs (assuming 0)
35         sum = sum + (float)minFreq*inform;
36     }
```

Figure D.1 Similarity function used in duplicate removal stage of the preliminary preprocessing (part 1)

```

37     //find the sum of inform-s in each story:
38     float sumInformD1=0;
39     float sumInformD2=0;
40     for(unsigned int i=0; i<d1.size(); ++i)
41     {
42         it=informWords.find(d1[i]);
43         if(it!=informWords.end())
44             sumInformD1 = sumInformD1 + it->second;
45     }
46     for(unsigned int i=0; i<d2.size(); ++i)
47     {
48         it=informWords.find(d2[i]);
49         if(it!=informWords.end())
50             sumInformD2 = sumInformD2 + it->second;
51     }
52     if(sumInformD1 > 0 || sumInformD2 > 0)
53         similarity = sum / max(sumInformD1, sumInformD2);
54     return similarity;
55 }

```

Figure D.2 Similarity function used in duplicate removal stage of the preliminary preprocessing (part 2)

APPENDIX E Number of Data Points

The following figures accompany the charts used in Chapter 6, showing the number of data points used for the corresponding bin of a particular indicator.

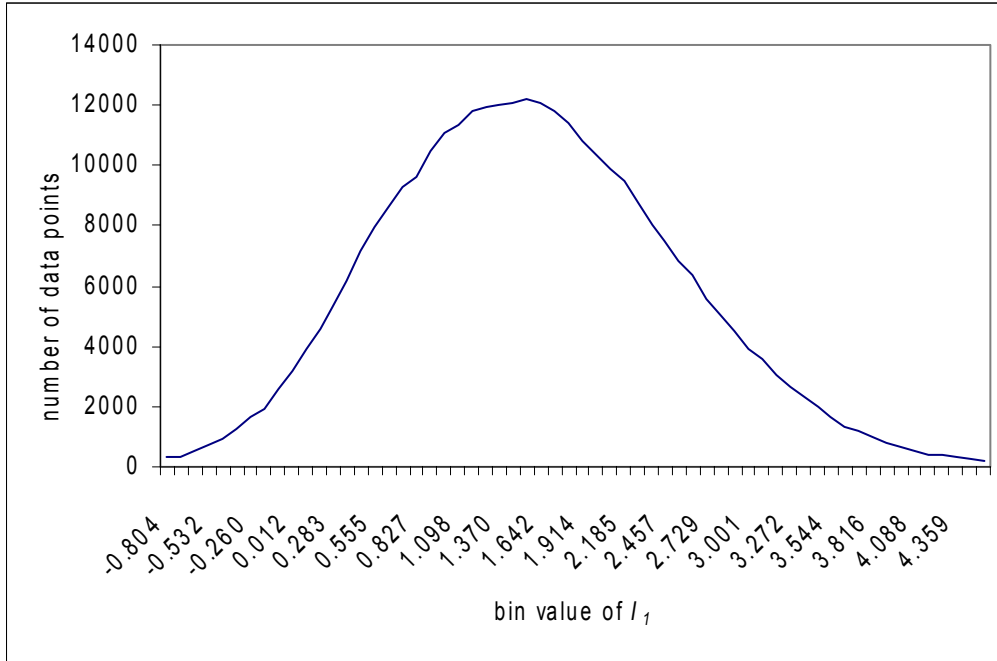


Figure E.1 Number of data points for the results in Figure 6.17

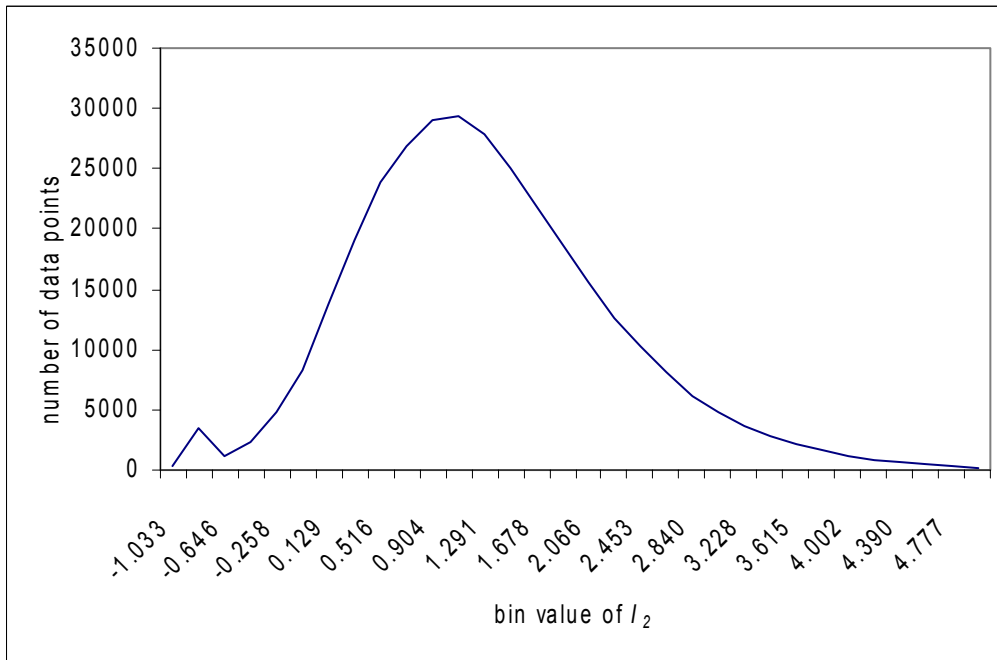


Figure E.2 Number of data points for the results in Figure 6.18

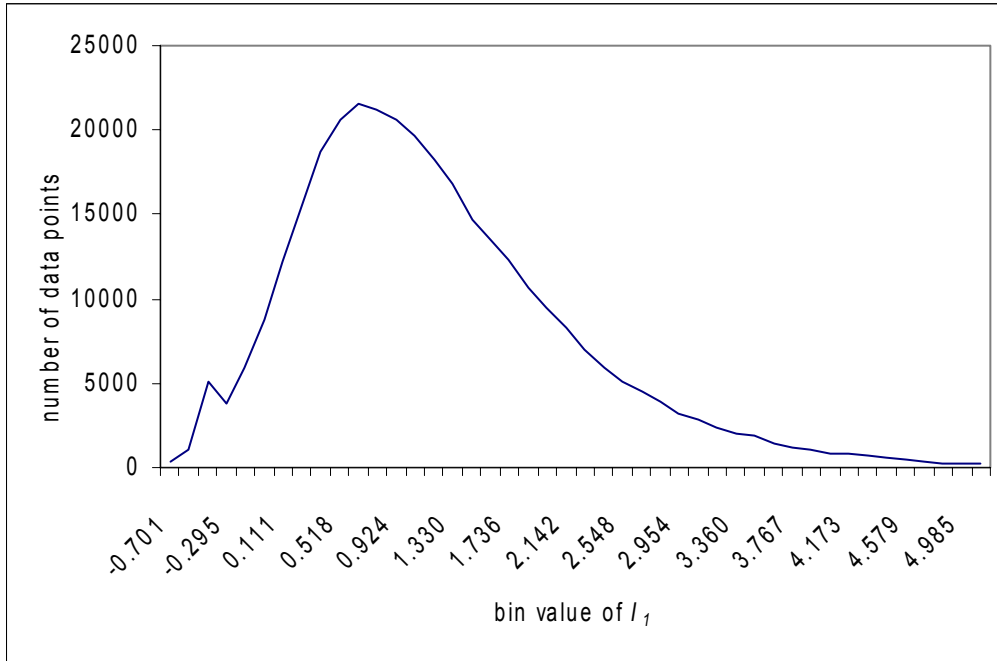


Figure E.3 Number of data points for the results in Figure 6.19

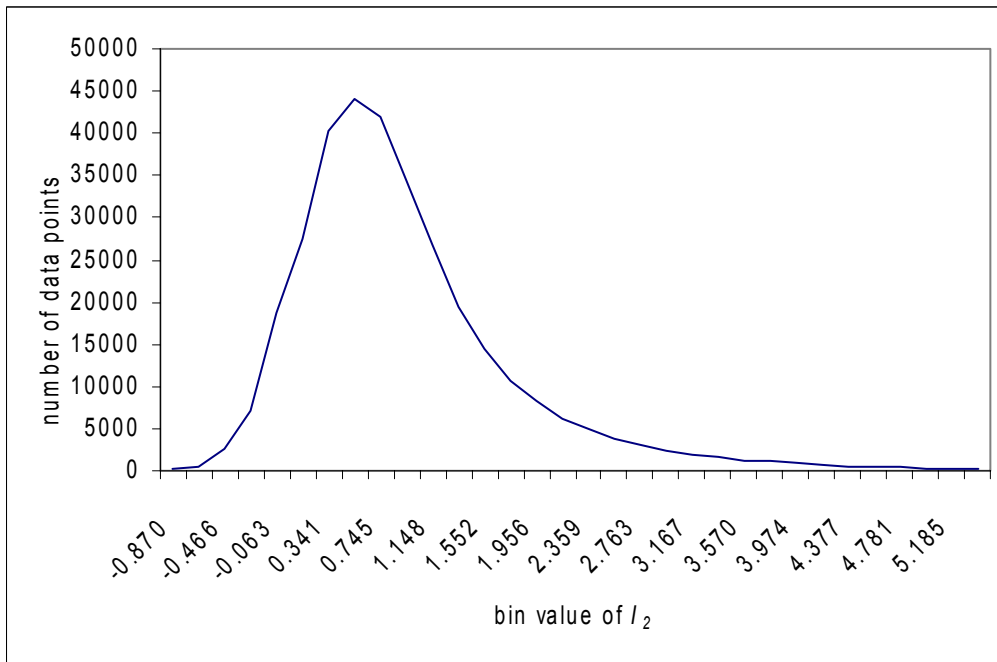


Figure E.4 Number of data points for the results in Figure 6.20