Stock Returns Prediction with Chinese Text Using Machine Learning

Man Yin Michael Yeung

supervised by Prof. Haifeng You

to fulfil the requirement of UROP1100E under
Undergraduate Research Opportunities Program on the project
"Investment Analysis with Machine Learning"
The Hong Kong University of Science and Technology

December 21, 2021

Abstract

Machine learning has facilitated the process to extract sentiment, the underlying subjective information, from financial text corpus. Recent literatures have introduced various models to make inference and predictions. In this study, different text-mining methodologies, including SESTM model, Word2Vec, and dictionary-based methods (as benchmark) are implemented to predict the stock returns of Chinese stocks. In additional, a portfolio analysis is conducted to examine the return and risk performances of some trading strategies based on the sentiment signals obtained by the models. A dataset containing Chinese analyst reports is adopted to perform empirical investigation on the model performances. The empirical result shows that different models have similar test performances, with Spearman correlations of approximately 0.1 to the actual 2-day return. Portfolio analysis shows a Sharpe ratio of optimally 0.96 in the test period horizon.

1. Introduction

Technological advancement has fostered the feasibility to investigate large of amount of data using machines and reduced human efforts. Textual data is believed to be the fastest growing data from academic research. However, seemingly it is not that direct to deal with textual data as the numerical representations of text data are often ultra-high dimensional and sometimes sparse, which is computationally challenging. Empirical research must confront this barrier when attempting to unleash its potential.

The subjective information behind financial texts which describes the "degree of positiveness" the author of an article is having is sometimes referred to as "sentiment". Hence, if one can correctly interpret the sentiment behind a large amount of financial text, he/she can somehow estimate the investors' confidence on individual assets which is arguably a good predictor of the returns of the respective underlying assets.

Ke, Kelly, and Xiu (2020) presented a model-based approach to understand the sentimental structure of a text corpus without pre-existing dictionaries. It is believed that return-predictive content of an event is reflected in both the news article text and the returns of the asset. The model developed was named SESTM (Sentiment Extraction via Screening and Topic Modelling) consisting of three steps.

In this study, various models including SESTM, dictionary-based method, and Word2Vec (used together with SESTM or other supervised learning models) are implemented to compare their test performances. In addition, a portfolio analysis is conducted to examine the performance of trading strategies based on the models implemented. Difference performance metrics are also computed for comparison.

The rest of this paper is organized as follows. Section 2 presents the Chinese Text pre-processing method, Jieba, used to divide sentences into phrases. Section 3 reviews the literature by Ke et al. (2020), which describes the SESTM model in a more detailed manner. Section 4 presents other models used in this study, including dictionary-based method and the Word2Vec model. Section 5 briefly introduces the dataset used in this study and explain the hurdles in it to be overcome. Section 6 includes the empirical results obtained by implementing the methodologies using Python. Sections 7 and 8 contains the discussions, limitations of this study, and the conclusions made. Finally, Section 9 suggests future work subsequent to this study.

2. Chinese Text Pre-processing

Jieba is a model which is used for pre-processing Chinese text. It can divide Chinese sentences into phrases, similar to the "tokenization" step commonly used in the natural language processing of English texts. In addition, it can provide part-of-speech tagging, meaning that each of the phrases in the output is tagged with a part-of-speech.

For example, if we input a sentence "我的衣服真漂亮" (meaning "my clothes are really beautiful") into the Jieba model, the output will be[1]:

[('我', 'r'), ('的', 'uj'), ('衣服', 'n'), ('真', 'd'), ('漂亮', 'a')]

Informally Translated: [('Me', 'r'), ('’s', 'uj'), ('clothes', 'n'), ('really', 'd'), ('beautiful', 'a')]

The sentence is cut (or "tokenized") into phrases and are tagged with a part-of-speech. The full list of part-of-speech can be found in the appendix.

3. Literature Review: SESTM Model

The SESTM model introduced by Ke et al. (2020) has three steps: 1) screening of *sentiment-charged* words, 2) learning sentiment topics, and 3) scoring new articles. Prior to these steps, a probabilistic model is first introduced.

3.1. Probabilistic Model

Considering a collection of n news and a dictionary of m words, the document term matrix $D = [d_1, \dots, d_n] \in \mathbb{R}^{m \times n}$ contains article vectors $d_i \in \mathbb{R}_+^m$ so that $d_{j,i}$ is the number of times word j occurs in article i. The submatrix $D_{[S],\cdot}$ contains a subset of rows from D, where only *sentiment-charged* words (set S) is included (remaining words are defined as *sentiment-neutral*). It is assumed that each article possesses a sentiment score $p_i \in [0,1]$, where $p_i = 1$ represents maximally positive article sentiment, vice versa. In particular, for stock return y_i

$$Pr(sgn(y_i) = 1) = g(p_i), \text{ for a monotone increasing function } g(\cdot)$$

The *sentiment-charged* word counts, $d_{[S],i}$, are generated by a mixture of multinomial distribution of the form

[1] Using the Python code: `[tuple(x) for x in jieba.posseg.cut('我的衣服真漂亮', use_paddle = True)]`

$$d_{[S],i} \sim Multinomial(s_i, p_i O_+ + (1 - p_i) O_-)$$

$O_\pm$ describe the expected word frequencies in maximally positive and negative sentiment articles respectively. It captures the information on both the frequency of words as well as their sentiment. They can be reorganized to the *vector of frequency*, F, and *vector of tone*, T

$$F = \frac{1}{2}(O_+ + O_-) \qquad T = \frac{1}{2}(O_+ - O_-)$$

3.2. Screening of Sentiment-Charged Words

Sentiment-neutral words are considered as noise dominating the data in terms of number of terms and in total of counts. Hence, the subset of *sentiment-charged* words is isolated and leaving *sentiment-neutral* words unmodelled. As such, a feature selection procedure is carried out. Only terms having counts above threshold and associated with positive returns are considered. To define "associated with positive returns", f_j is introduced (with a variant f_j^*)

$$f_j = \frac{count\ of\ word\ j\ in\ article\ with\ sgn(y) = +1}{count\ of\ word\ j\ in\ all\ articles}$$

$$f_j^* = \frac{count\ of\ articles\ including\ word\ j\ AND\ sgn(y) = +1}{ccount\ of\ articles\ including\ word\ j}$$

f_j^* is adopted in this study as it is more robust to outliers. Positive sentiment words are expected to have high f_j^* (i.e., above $\hat{\pi}$, the empirical proportion of the number of articles associated with positive returns), vice versa. Moreover, a minimum threshold κ is applied on the count of words. Hence, the estimate of set S is defined by

$$\hat{S} = \{j: f_j^* \geq \hat{\pi} + \alpha_+ \text{ or } f_j^* \leq \hat{\pi} + \alpha_-\} \cap \{j: k_j \geq \kappa\}$$

where $(\alpha_+, \alpha_-, \kappa)$ are hyperparameters that can be tuned by validation.

3.3. Learning Sentiment Topics

Let $h_i = d_{[S],i}/s_i$ denote the $|S| \times 1$ vector of word frequencies. The multinomial distribution previously introduced implies that:

$$\mathbb{E}h_i = \mathbb{E}\left[d_{[S],i}/s_i\right] = p_i O_+ + (1 - p_i) O_- \Leftrightarrow \mathbb{E}H = OW$$

where $W = \begin{bmatrix} p_1 & \cdots & p_n \\ 1-p_1 & \cdots & 1-p_n \end{bmatrix}$ and $H = [h_1, h_2, \cdots, h_n]$. Here, H is unobserved and is estimated using $\widehat{h_i} = d_{[S],i}/\widehat{s_i}$ where $\widehat{s_i} = \sum_{j \in \hat{S}} d_{j,i}$.

The standardized ranks of returns are used to estimate W, i.e., $\widehat{p_i} = \frac{rank\ of\ y_i\ in\ \{y_l\}_{l=1}^{n}}{n}$. After $O = [O_+, O_-]$ is obtained by regression, it may contain negative entries. Hence, all these entries are set to zero and O is re-normalized.

3.4. Scoring New Articles

Given $\hat{S}$ and $\hat{O}$, the estimated sentiment of a new article, p, can be obtained using maximum likelihood estimation with a penalty term (having tuning parameter $\lambda > 0$).

$$\hat{p} = \arg\max_{p \in [0,1]} \{\hat{s}^{-1} \sum_{j \in \hat{S}} d_j \log(p_i \hat{O}_{+,j} + (1-p_i)\hat{O}_{-,j}) + \lambda \cdot \log(p(1-p))\}$$

Imposing the penalty shrinks the estimate toward a score of 0.5 (as most articles should have a neutral sentiment), where the amount of shrinkage depends on λ.

4. Other Models

4.1. Dictionary-based Method

The dictionary-based method computes a set of 12 different scores of the articles and use these scores as an estimator of the target variable – n-day return. The Spearman correlation between a score and the n-day return acts as the performance metric of this method.

Pre-trained dictionaries are retrieved from different online sources[2]. Each of these dictionaries contains a set of positive words and a set of negative words. This defines the "positive" and "negative" phrases used when computing the 12 scores. The 12 scores defined in this project, from 1 to 12, are as follows:

- Scores 1-3: Phrase-based sentiments with non-stopword count as the denominators.

[2] Dictionaries 1&2: 金融领域中文情绪词典. 姚加权，冯绪，王赞钧，纪荣嵘，张维. 语调、情绪及市场影响：基于金融情绪词典. 管理科学学报，2021. 24(5), 26-46. [Online]. Available: https://github.com/dictionaries2020/SentimentDictionaries
Dictionary 3: 中文金融情感词典. 姜富伟、孟令超、唐国豪，“媒体文本情绪与股票回报预测”，《经济学(季刊)》，2021 年第 4 期，第 1323-1344 页。Fuwei Jiang, Joshua Lee, Xiumin Martin, and Guofu Zhou.“Manager Sentiment and Stock Returns” Journal of Financial Economics 132(1), 2019,126-149. [Online]. Available: https://github.com/MengLingchao/Chinese_financial_sentiment_dictionary

Positive phrases count, negative phrases count, and the difference of the two are used as the numerators respectively.

- Scores 4-6: Similar to Scores 1-3, with *Sentiment-charged words* count used as the denominator instead
- Scores 7-12: Sentence-based version of Scores 7-12

For sentence-based scores, the sentiment of a sentence is defined as majority sentiment of the phrases within the sentence. The count of positive and/or negative -sentiment sentences are then used to compute the scores.

Spearman (and Pearson) correlations[3] to 2-day and [t+2,t-6] returns are then computed, using the full dataset and test data split respectively.

4.2. Word2Vec

The Word2Vec model transforms words or phrases into high-dimensional numerical vectors, which are much easier to process. Two different ways of implementing Word2Vec is implemented in this study, including to train the 100-dimensional vectors using the set of sentiment charged words in the SESTM model, and to use available pre-trained 300-dimensional vectors[4] directly.

After that, the phrase-level vectors are transformed into article-level vectors by averaging the phrase-level vectors in the articles. These vectors are then used as predictors in different models, including SESTM (using the vector entries as predictors) with (updated) MLE, directly calculating the vector difference between the article vectors and the "positive" and "negative" vectors O, and using the vector entries as the predictors in various supervised learning models.

[3] Special Note: In the way that the scores are defined, the correlation for Sentiment-charged words-as -denominator version scores should be equal, e.g., Scores 4-6, 9-12. Proof:

$$\frac{-n}{p+n} = 1 + \frac{p}{p+n} \Rightarrow \text{corr}(\frac{-n}{p+n}, return) = \text{corr}(1 + \frac{p}{p+n}, return) = \text{corr}(\frac{p}{p+n}, return)$$

$$\frac{p-n}{p+n} = 2(\frac{p}{p+n}) - 1 \Rightarrow \text{corr}(\frac{p-n}{p+n}, return) = \text{corr}(2(\frac{p}{p+n}) - 1, return) = \text{corr}(\frac{p}{p+n}, return)$$

[4] Financial News (Word) vectors available at https://github.com/Embedding/Chinese-Word-Vectors

5. Datasets

The textual dataset contains 1,693,001 rows and 11 columns[5]. Each row represents an analyst report. The price dataset (actually the specific return is used, abbreviated "specret") contains 11,455,412 rows of data including the specific returns of the trading dates between 2000-03-31 and 2021-09-17 (YYYY-MM-DD) inclusive of various Chinese stocks. Another dataset contains the monthly returns similarly.

5.1. Data Pre-processing

Duplicated rows of the textual dataset are first dropped to obtain textual data which are distinct in terms of the text data column. Jieba "posseg" is then applied on the remaining data to obtain a list of tuples containing the phrase and its part-of-speech tag. All stop-words, punctuations, and numbers are also pruned in the same step.

Next, the price data is pre-processed to obtain the specific return day t, day t+1, etc. until t+6. They are then summed to obtain the (day-)t return, [t,t+1] return and [t+2,t+6] return of each day. After that, the textual data and price data are merged, so that the textual data is "augmented" to obtain columns of the three types of specific returns.

The process is continued and repeated using a sentence-based approach of the "Jieba cuts", where phrases in the same sentence are contained in nested lists.

Table 1. Summary Statistics

Filter	Remaining Sample Size
Total number of Analyst Reports	1,693,001
Remove Duplicates	597,040
Augmenting Specific Return Columns	472,693

[5] The columns are 'ID', 'SecuCode', 'TITLE', 'content', 'TYPE_ID', 'ORGAN_ID', 'AUTHOR', 'create_date', 'entrytime', 'FYEAR', 'FQTR'

6. Empirical Analysis

6.1. SESTM

The SESTM model is trained and tested using different specifications. The number of sentiment words, the part-of-speech (POS) filtering method, and the restriction on data used for training are varied in addition to the hyperparameters of the SESTM model.

Figures 1 and 2 summarizes the Pearson and Spearman correlations (to 2-day and [t+2, t+6] specific return) obtained from the testing[6]. The 2-day specific return is always used as the training label (target variable). The 2-day correlations are higher than the [t+2, t+6] correlations in general.

The rolling Pearson and Spearman correlations (rolling in the testing process) are recorded and plotted in figures 4 and 5[7]. Some of them have slightly better test correlations, but generally converges to a narrow range (approximately a Spearman correlation of 0.1).

Figure 6 shows a boxplot of the 2-day specific return against the predicted sentiment. The result shows that there is a clear increasing trend, but the wide box implies that the variance to the mean is large. Hence, the signal may contain much noise.

Figure 7 and 8 shows the mean and median 2-day returns of different QCUTs of the predicted sentiment. Similar to the boxplot, it shows a clear increasing trend of 2-day returns with respect to the predicted sentiment. This shows the signals generated by this model is satisfactory in terms of mean (meaning that we must aggregate large number of signals).

---

[6] The rows represent both the "number of positive and negative words" and the correlation method used. For example, "500 Pearson" means 500 positive (sentiment) words and 500 negative words are chosen by the model, and the model uses "Pearson" method to calculate the correlations. The columns represent both the POS used for words (actually, phrases) filtering, and the restriction on the training data. "detailed" means filtering away the words with POS detailed in the table shown in Figure 3, and "A+V" means isolating adjectives and verbs only. For the restriction, "All" means there are no restrictions on training data. "[-2%,+2%]" means only the training data with return within this range are used for training. "]-2%,+2%[" means the opposite (with ±2% included). ±2% is always inclusive.

[7] The labels of the plots are the identifiers of the testing. The identifier is structured:

<Model Name>_<Kappa Quantile>_<Positive & Negative Words>_<Lambda>
<POS Filtering Method><Training Label ("specret_2d")>
_<Training Data Label Restriction (TDR) [-2%,+2%] Used or Not>
_<Sum of Hash of Sentiment Charged Words Used, Modulus 100000>

Figure 1. Pearson and Spearman Correlations to 2-day Return (Day t to t+1) using Test Data (2015-01-01 to 2021-08-19).

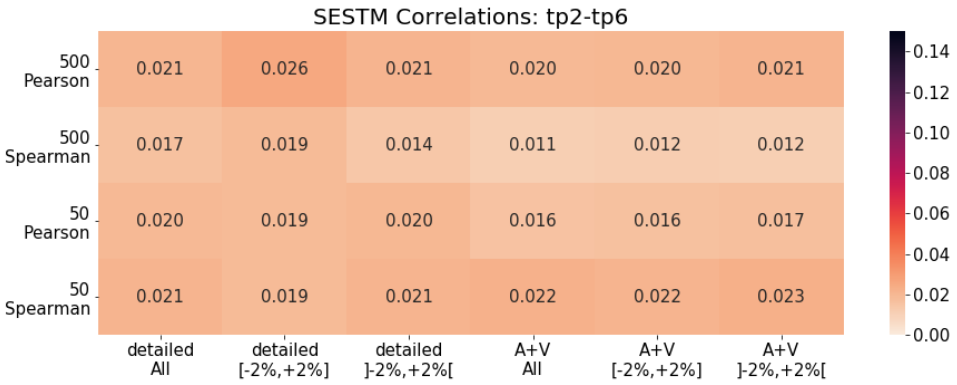

Figure 2. Pearson and Spearman Correlations to Day t+2 to Day t+6 Return using Test Data (2015-01-01 to 2021-08-19).

单字			两字词			三字词			四字及以上	
POS	Explanation		POS	Explanation		POS	Explanation		POS	Explanation
w	标点符号		nw	作品名		s	处所名词		s	处所名词
r	代词		u	助词		r	代词		LOC	地名
ns	地名		m	数量词		LOC	地名		f	方位名词
f	方位名词		t	时间		f	方位名词		ORG	机构名
p	介词		TIME	时间		ORG	机构名		n	普通名词
c	连词		nr	人名		nt	机构名		nz	其他专名
q	量词		PER	人名		n	普通名词		PER	人名
n	普通名词		nz	其他专名		nz	其他专名		nr	人名
nr	人名		n	普通名词		PER	人名		TIME	时间
m	数量词		q	量词		nr	人名		t	时间
u	助词		ORG	机构名		TIME	时间		m	数量词
			f	方位名词		t	时间		nw	作品名
			LOC	地名		m	数量词			
			r	代词		nw	作品名			
			s	处所名词						

Figure 3. Detailed POS Filtering based on Phrase Length

(Number of characters)

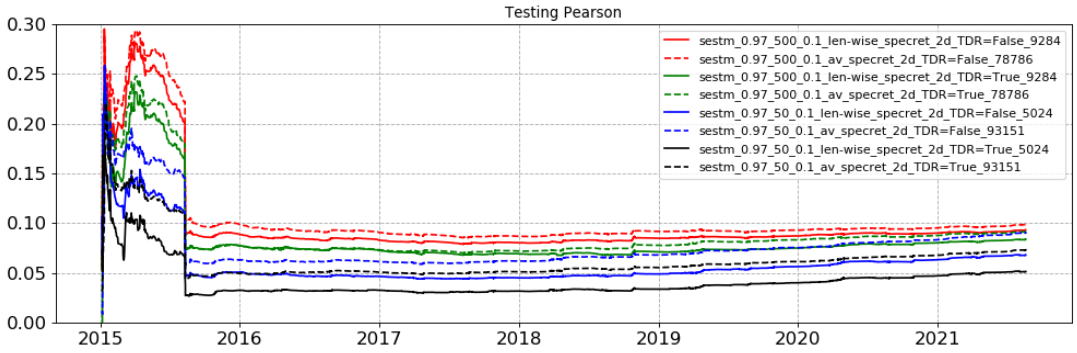

Figure 4. Rolling Pearson Correlation to 2-day Return (Day t to t+1) on Rolling Testing Data (2015-01-01 to maximum 2021-08-19):

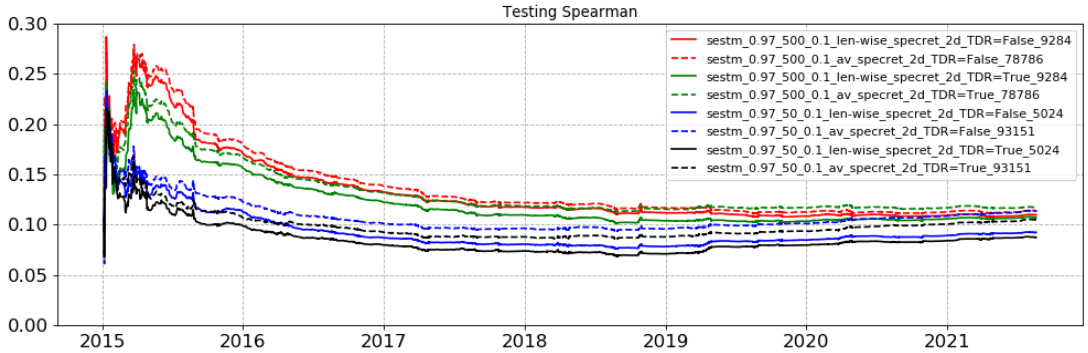

Figure 5. Rolling Spearman Correlation to 2-day Return (Day t to t+1) on Rolling Testing Data (2015-01-01 to maximum 2021-08-19):

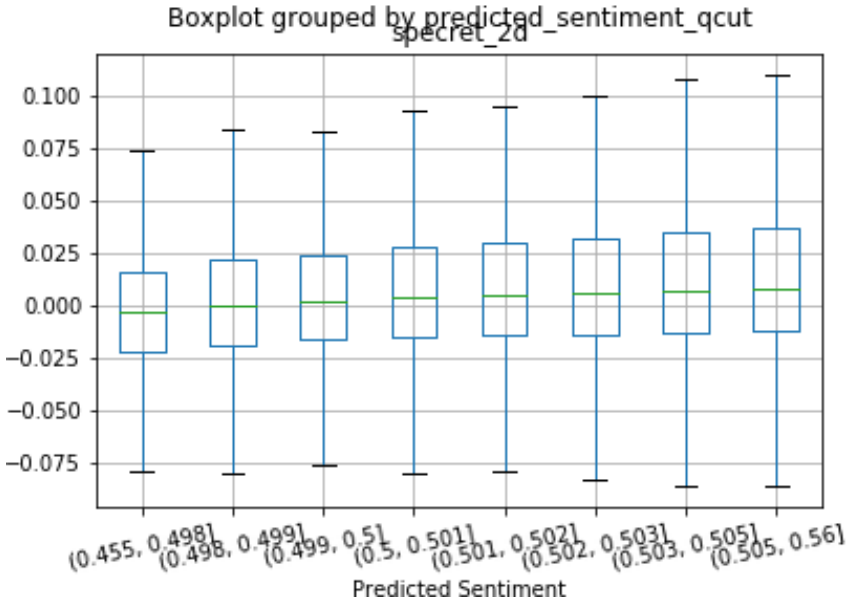

Figure 6. Best Testing Result Boxplot.
QCUT(8) by Predicted Sentiment, Fliers Not Shown. Identifier:
sestm_0.97_500_0.1_av_specret_2d_TDR=True_78786 (green dashed line)

predicted_sentiment_qcut	specret	specret_2d	specret_tp2-tp6	sestm_0.97_500_0.1_av_specret_2d_TDR=True_78786
(0.455, 0.498]	-0.001109	-0.001836	-0.000965	0.495226
(0.498, 0.499]	0.001683	0.002776	-0.000260	0.498469
(0.499, 0.5]	0.003145	0.005509	0.000008	0.499808
(0.5, 0.501]	0.005186	0.008683	0.000888	0.500736
(0.501, 0.502]	0.006693	0.010693	0.001540	0.501657
(0.502, 0.503]	0.007048	0.011476	0.001533	0.502612
(0.503, 0.505]	0.009098	0.014449	0.001945	0.503820
(0.505, 0.56]	0.010254	0.016160	0.001429	0.506762

Figure 7. Mean, QCUT(8) by Predicted Sentiment

predicted_sentiment_qcut	specret	specret_2d	specret_tp2-tp6	sestm_0.97_500_0.1_av_specret_2d_TDR=True_78786
(0.455, 0.498]	-0.002216	-0.003481	-0.004457	0.495872
(0.498, 0.499]	-0.000517	-0.000173	-0.003159	0.498516
(0.499, 0.5]	0.000360	0.001577	-0.003258	0.499873
(0.5, 0.501]	0.001575	0.003763	-0.002407	0.500746
(0.501, 0.502]	0.002212	0.004959	-0.001783	0.501664
(0.502, 0.503]	0.002325	0.005573	-0.003035	0.502602
(0.503, 0.505]	0.003399	0.007100	-0.002379	0.503788
(0.505, 0.56]	0.003924	0.007897	-0.003012	0.506035

Figure 8. Median, QCUT(8) by Predicted Sentiment

6.2. Dictionary Based Method

Figure 9 to 12 summarizes the results of the Pearson and Spearman correlations to the 2-day and [t+2, t+6] return. The testing is done for two versions, on full data and on test data only. The full data testing is used to give a more general test results as more data is included in the testing (note that no training is needed for this model). The test data testing provides test performance which can be fairly compared with other models which requires testing as the test data will be commonly used for testing.

The results shows that the predicted sentiment has much higher correlations to 2-day return than to [t+2, t+6] return. This is intuitive as the information is impounded into the price of the asset after a period (a good guess will be 2-3 days). The results also show that some scores work better generally, for example, Scores 4-6 and 10-12. However, the differences between the correlations are close.

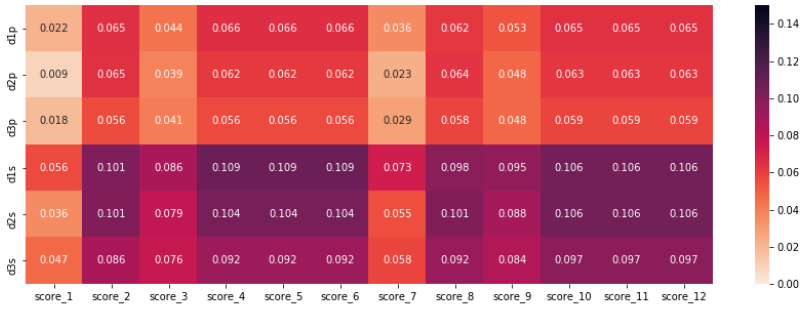

Figure 9. Pearson and Spearman Correlations to 2-day Return (Day t to t+1) Using Full Data (2010-01-04 to 2021-08-19):

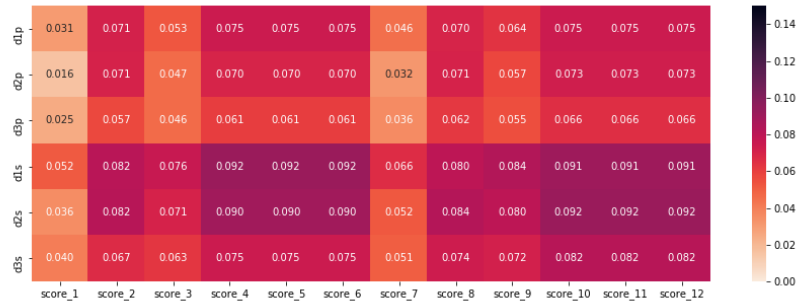

Figure 10. Pearson and Spearman Correlations to 2-day Return (Day t to t+1) Using Test Data (2015-01-01 to 2021-08-19)

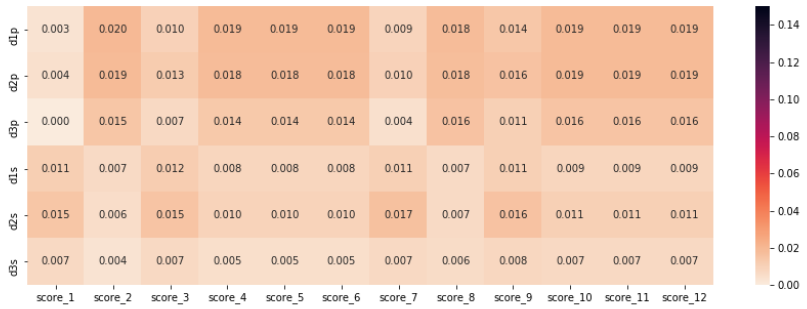

Figure 11. Pearson and Spearman Correlations to Day t+2 to Day t+6 Return Full Data (2010-01-04 to 2021-08-19)

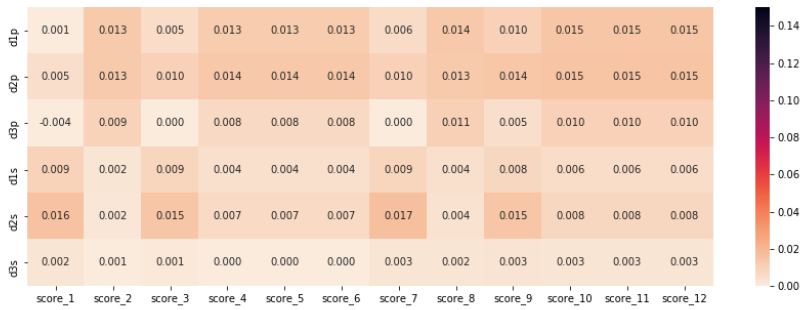

Figure 12. Pearson and Spearman Correlations to Day t+2 to Day t+6 Return Testing Data (2015-01-01 to 2021-08-19)

6.3. Word2Vec

Word2Vec is used mainly in the pre-processing stage. Two methods of Word2Vec are implemented here, including training 100-dimentional vectors with the trained data, and the 300-dimensional pre-trained vectors using available sources.

For the 100-dimensional vectors, we use these vectors to represent the filtered words. The vectors within the same article are summed up elementwise to obtain article vectors, which replaces the h_i in SESTM, to be the "predictors" of the model. In the process, further word filtering is also performed by removing positive sentiment words with statistically different vectors (with Euclidean distance to the average vector above a certain number of standard deviation), similar for negative sentiment words.

6.3.1. Vector Difference

A simple model of vector different is implemented as a benchmark. The sentiment is defined as: "The Euclidean distance to Positive Sentiment average vector minus that to Negative Sentiment average vector". The result shows a 0.1478 training Spearman correlation and a 0.0887 test Spearman correlation to 2-day specific return.

6.3.2. Word2Vec Words Pruning

The following is a 2D projection of the word vectors (shown in Figure 13), Datapoints labelled with words represent words (phrases) that are further pruned. Green dots are positive words and red dots are negative words.

6.3.3. New MLE Suggestion

As the predictors no longer follow a multinomial distribution, the MLE should be replaced. A suggested testing method will be considering $O_{\pm}$ as the maximally positive and maximally negative vectors, i.e., the vectors for the maximally positive and negative article.

Other articles with fewer extreme sentiments are assumed to have linear combinations of the two "topic vectors". Hence, we find the linear combination with the least Euclidean distance to the article vector.

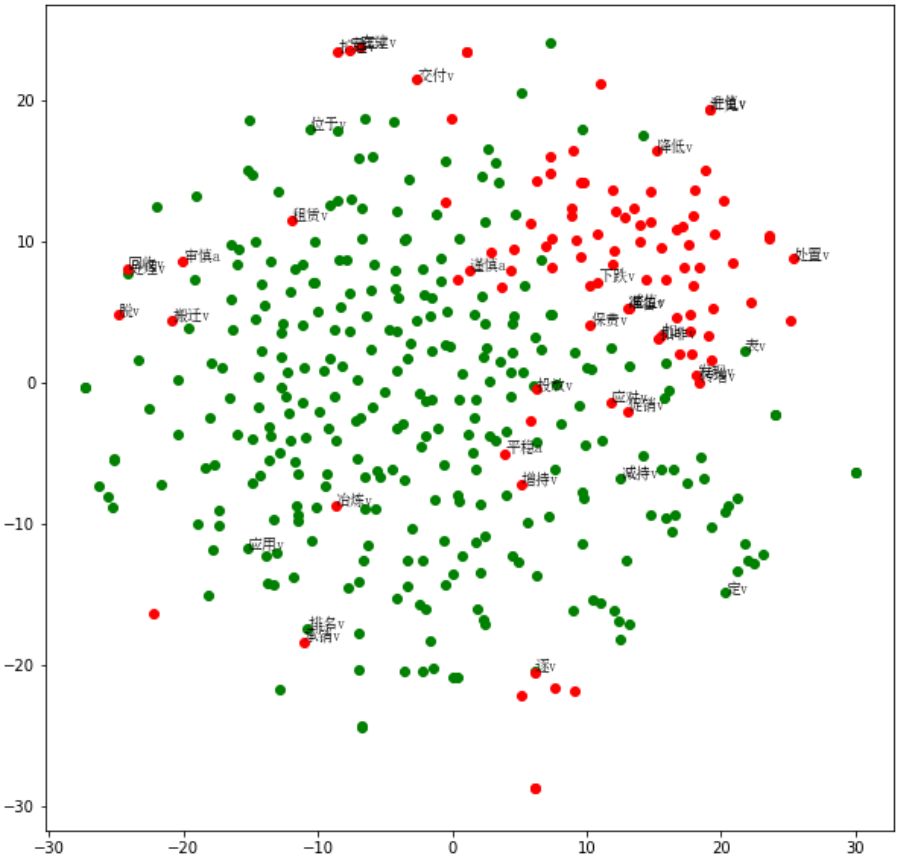

Figure 13. Positive and Negative Sentiment Words (Phrase Shown: Removed), 100D Vectors Projected to a Low-dimensional Plane

Suggested MLE which minimizes the difference between linearly estimated topic vector and the article-level vector:

$$argmin_{p\in[0,1]}\left\|(p\overrightarrow{O_+}+(1-p)\overrightarrow{O_-})-\overrightarrow{h_l}\right\|_2$$

$$\text{where } \overrightarrow{h_l}=\frac{1}{s_i}\sum_{i=1}^{n}\overrightarrow{v_n}$$

The test Result shows a Spearman correlation of 0.0922.

Some parts of this model are still in progress and hence empirical results are not obtained yet, including to use pre-trained 300D vectors and using supervised learning models on the predictors (the elements of the vectors). Those parts will be the next steps of the continuation of this project.

6.4. Portfolio Analysis

Portfolios are constructed based on the sentiment signals generated by the models. In our analysis, we adopt 30 days, 60 days, and 90 days signals for comparison. These signals are computed using the n-day average sentiment (of all analyst reports in the respective periods) of the given stock.

The stocks are divided to 10 groups – group 1 contains the top 10% stock with the highest sentiment, group 2 contains the next 10% stock with high sentiments, etc. Group 10 contains the 10% stock with the lowest sentiment. We name these 10 portfolios as Portfolios 1 to 10. We have another hedged portfolio (Portfolio 0) which takes the long position of portfolio 1 and the short position of portfolio 10.

In our analysis, the 11 portfolios are computed in a monthly basis. Strategies going into a long position of Portfolio X every month are analysed, based on the cumulative returns, risk, and other metrics. For example, one of the strategies is to go into a long position of Portfolio 1 for one month every month.

Figure 14 shows the cumulative return of the of a particular portfolio (Portfolio 0, SESTM, 0.97_500_0.1_av_specret_2d_TDR=True_7876, 30/60/90-day). The result shows that the return is gradually increasing, and the 60-day signal performs the best. It may not be true for some other cases tested,

Figure 15 shows the annualized return, annualized risk, and Sharpe ratio of the portfolio. The results shows that different specification results in different resulting metrics. For example, portfolios using 50 sentiment words have lower annualized risk and higher Sharpe ratio in general.

The 11 portfolios using the same model and specifications are compared. Figure 16 is an example. The visualization shows that, at the ending month, the sequence of the portfolios, from highest to lowest final account values, is roughly similar to the sequence 1 to 10. This implies that portfolios using higher predicted sentiment has generated more account value, further implying that the trading strategy is working when the account value (hence returns) are compared.

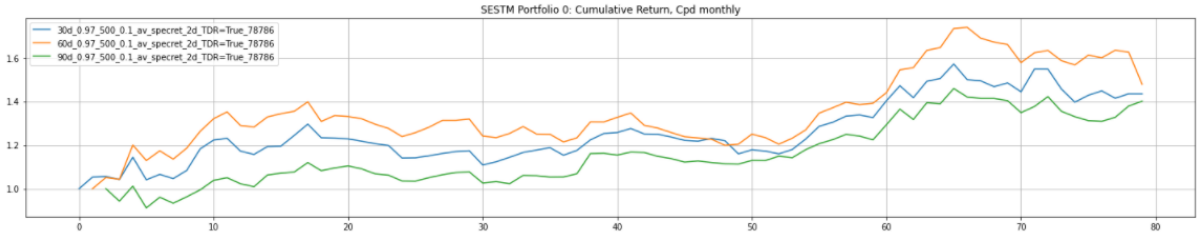

Figure 14. SESTM Portfolio 0. Cumulative Return Compounded Monthly

30/60/90-day Signal, 0.97_500_0.1_av_specret_2d_TDR=True_7876

Figure 15. SESTM Portfolio 0.

Annualized Return, Annualized Risk, Sharpe Ratio

30/60/90-day Signal, 0.97_500_0.1_av_specret_2d_TDR=True_7876

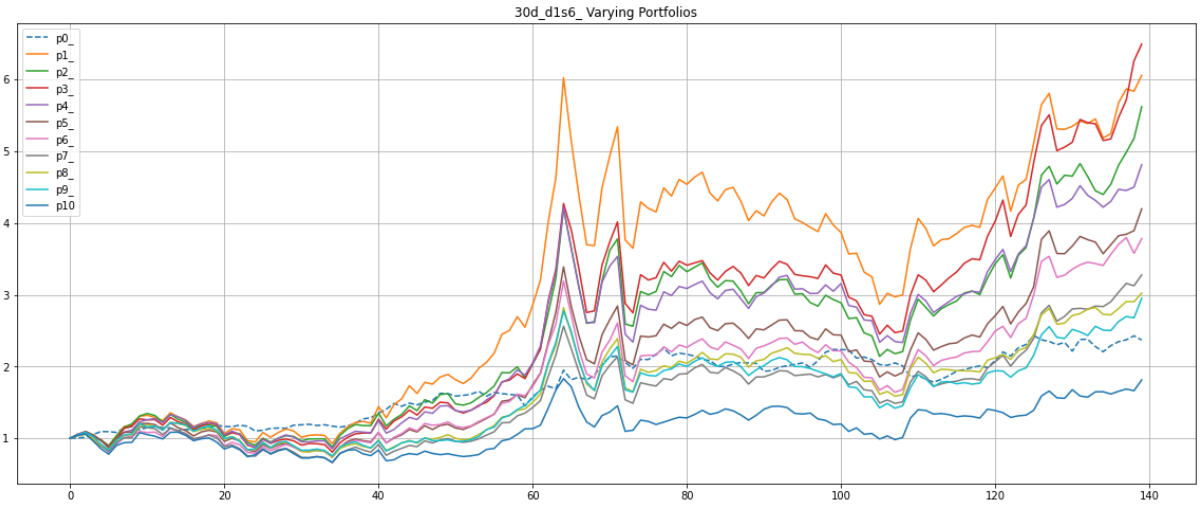

Figure 16. All Portfolios. 30-day, Dictionary(1)-based, Score 6

Account Value

7. Discussions

Some models in this study requires training and some models does not. Hence, the performance of different models may not be directly compared. For example, the dictionary-based method has both full data testing results and test data testing results. The full data approach tends to be more trustworthy as more data is used. The test data is the one that can be more fairly compared with other models require training. However, it is important to note that testing results should not be used for model selection. As a result, it is not very appropriate to compare different Scores in the dictionary-based approach, and at the same time declare that the best score of them is better/worse than the testing results of other models. A better approach will be to use the Score computed from the "training set" to select the best Scoring methodology (for the dictionary-based method), and use that methodology (for example, Score 6) to compute the testing results of the dictionary-based method as a whole.

For the portfolio analysis, notice that the final account value of different portfolios (in Figure 16) roughly follows the sequence "1" to "10". This implies that the trading strategy is performing well, meaning that the performance of portfolios using better signals from the model (for example, dictionary-based method in Figure 16) has better return (hence, having larger account value).

8. Limitations and Conclusion

There are some limitations in this study.

1. This study focuses on the bag-of-words approach to deal with the phrases cut by Jieba. The intra-sentence POS relations are not investigated nor modelled. For example, some negation words such as "not" inverts the sentiment of the later words in some cases. Moreover, some parts of the article should be more important than others. Intuitively, the conclusion or summary/abstract parts of the article should have paid more emphasis on.
2. Not all kinds of supervised learning models are included in this study, so this study does not provide an all-rounded comparison to the readers. For example, tree-based models, and neural networks have not (yet) been investigated. It is possible that some of those models have better performances than those implemented in this study at the current stage.

Despite the limitations aforementioned, a conclusion can be made according to the empirical results of this study.

Models implemented in this study can generally obtain test Spearman correlations close to 0.1. This provides evidence to support the statement that sentiment extraction of Chinese analyst reports is useful to predict returns of Chinese stocks. The SESTM model has slightly better test performance than the dictionary-based method. The test Spearman correlation of the SESTM model can sometimes exceed 0.1, while those of dictionary-based method are slightly lower than 0.1 in general. The Word2Vec model applied in this study requires further tuning and trials of implementation to obtain better results that are comparable to the other two models. Finally, the portfolio analysis shows that the Sharpe ratio using dictionary-based method and SESTM model are close to 1, which reaches an informal standard of acceptable (profit-generating under acceptable risk-level) trading strategy.

9. Future Work

The Word2Vec models requires further investigation and implementation. It can be used either together with the SESTM model, and other supervised learning models (using the vector elements as predictors and the 2-day specific return as the target variable. Moreover, other models such as tree models, neural networks, and BERT could be subsequently investigated.

References

[1] Ke, Z., Kelly, B. T. and Xiu, D., 2020. Predicting Returns with Text Data. University of Chicago, Becker Friedman Institute for Economics Working Paper No. 2019-69, Yale ICF Working Paper No. 2019-10, Chicago Booth Research Paper No. 20-37, Available at SSRN: https://ssrn.com/abstract=3389884 or http://dx.doi.org/10.2139/ssrn.3389884

[2] Mikolov, T., Chen, K., Corrado, G. and Dean, J., 2013. Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.

Appendix

1. Meanings of the POS tags

标签	含义	标签	含义	标签	含义	标签	含义
n	普通名词	f	方位名词	s	处所名词	t	时间
nr	人名	ns	地名	nt	机构名	nw	作品名
nz	其他专名	v	普通动词	vd	动副词	vn	名动词
a	形容词	ad	副形词	an	名形词	d	副词
m	数量词	q	量词	r	代词	p	介词
c	连词	u	助词	xc	其他虚词	w	标点符号
PER	人名	LOC	地名	ORG	机构名	TIME	时间

(Disclaimer: Minor amendments are made after the official submission to the HKUST UROP office.)