FinSense: An Assistant System for Financial Journalists and Investors

Yi-Ting Liou
Department of Computer Science and Information Engineering, National Taiwan University, Taiwan
ytliou@nlg.csie.ntu.edu.tw

Chung-Chi Chen
Department of Computer Science and Information Engineering, National Taiwan University, Taiwan
cjchen@nlg.csie.ntu.edu.tw

Tsun-Hsien Tang
Department of Computer Science and Information Engineering, National Taiwan University, Taiwan
thtang@nlg.csie.ntu.edu.tw

Hen-Hsen Huang
Department of Computer Science, National Chengchi University, Taiwan
MOST Joint Research Center for AI Technology and All Vista Healthcare, Taiwan
hhhuang@nccu.edu.tw

Hsin-Hsi Chen
Department of Computer Science and Information Engineering, National Taiwan University, Taiwan
MOST Joint Research Center for AI Technology and All Vista Healthcare, Taiwan
hhchen@ntu.edu.tw

ABSTRACT

This paper demonstrates FinSense, a system that improves the working efficiency of financial information processing. Given the draft of a financial news story, FinSense extracts the explicit-mentioned stocks and further infers the implicit stocks, providing insightful information for decision making. We propose a novel graph convolutional network model that performs implicit financial instrument inference toward the in-domain data. In addition, FinSense generates candidate headlines for the draft, reducing a significant amount of time in journalism production. The proposed system also provides assistance to investors to sort out the information in the financial news articles.

CCS CONCEPTS

• **Computing methodologies** → **Natural language generation**.

KEYWORDS

Assistant system, tag recommendation, implicit information extraction

ACM Reference Format:

Yi-Ting Liou, Chung-Chi Chen, Tsun-Hsien Tang, Hen-Hsen Huang, and Hsin-Hsi Chen. 2021. FinSense: An Assistant System for Financial Journalists and Investors. In *Proceedings of the Fourteenth ACM International Conference on Web Search and Data Mining (WSDM '21), March 8–12, 2021, Virtual Event, Israel.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3437963.3441704

WSDM '21, March 8–12, 2021, Virtual Event, Israel

ACM ISBN 978-1-4503-8297-7/21/03.
https://doi.org/10.1145/3437963.3441704

News Article:
A federal judge gave his blessing to the US$26.5 billion merger between *T-Mobile* and *Sprint* on Feb. 11, several months after the deal got final antitrust approval from the U.S. government. Sprint surges 68.75%. T-Mobile rise 7.36%.

Related Stock:
Sprint, *T-Mobile*, **SoftBank**

Headline:
The Judge Nodded! *Sprint* Rise 70% in After-hour Trading.

Figure 1: An example of financial news.

1 INTRODUCTION

Implicit financial instrument inference is important for both financial journalists and investors when they write/read a financial news article. For example, the news of Google Drive may influence the stock price of Dropbox. However, most of the previous works [7, 15] only focus on using explicit information and neglect such implicit information. In this paper, we aim to increase the capability of machines toward the textual financial data, and demonstrate an assistant system, named FinSense, to deal with this issue.

Figure 1 shows a financial news article, where two companies, Sprint and T-Mobile, are explicitly mentioned. The stock price of another company SoftBank, which is not mentioned in the article, may be influenced since Softbank owns shares of Sprint. Most of the previous works [1, 3] analyze the impact toward explicit-mentioned targets, but omit implicit targets such as Softbank in this example. In this paper, we propose a model based on both BERT [6] and graph convolutional network (GCN) [11]. Experimental results show that our approach successfully enhances BERT with an increase of 8.00% of micro-averaged F1-score.

The working process of a financial journalist is shown as follows.

(1) Write down a news article.

(2) Select the explicit-mentioned stocks and the implicit ones in the article.
(3) Write down the headline of the news article.

FinSense selects the related stocks, including explicit-mentioned stocks and implicit stocks, efficiently from a news article. Financial journalists can edit the news article in the FinSense. After they complete the article, FinSense recommends the related stocks for consideration. Financial journalists can further select the stocks recommended by FinSense.

We also generate the headline for financial journalists based on our previous work [4]. Although the generated headline may not be perfect, it can still provide a possible direction for financial journalists. The proposed system can not only be applied to the working process of the financial journalists, but also provide an assistant for the investors. Since few news vendors provide the related stocks with a news article, most investors need to infer the related stocks, especially the implicit ones, by themselves. From this point of view, FinSense recommends implicit stocks to investors and assists their decision-making process.

2 METHODS

In this paper, we experiment on the news from June 22, 2013 to June 20, 2018. Each instance consists of one news article and multiple tags for the related stocks, including both tags for explicit-mentioned stocks and implicit stocks. The tags are labeled by the professional financial journalist.

2.1 Problem Formulation

Feature Matrix. Given the set of candidate tags $\mathcal{T} = \{t_1, \ldots, t_{|\mathcal{T}|}\}$, and the set of attributes $\mathcal{F} = \{f_1, \ldots, f_{|\mathcal{F}|}\}$, we generate a binary feature matrix $\mathbf{X} \in \{0,1\}^{|\mathcal{T}|\times|\mathcal{F}|}$, where $X_{ij} = 1$ if the tag t_i has the attribute f_j, otherwise $X_{ij} = 0$. In this system, a tag is one of the listed or over-the-counter stocks in Taiwan. The attributes of the tag mean products or industry type correspond to the company trading the stock. If the company manufactures the product or belongs to the industry type, then we call the company has that attribute.

Definition of Explicit and Implicit Tags. Given a news article $\mathbf{c} = (c_1, \ldots, c_{|\mathbf{c}|})$ with $|\mathbf{c}|$ words and the set of corresponding related tags $\mathcal{T}^{\mathbf{c}} \subseteq \mathcal{T}$, we denote the set of explicit tags as $\mathcal{T}^{\mathbf{c}} \cap \{c_i\}_{i=1}^{|\mathbf{c}|}$ and the set of implicit tags as $\mathcal{I}^{\mathbf{c}} = \mathcal{T}^{\mathbf{c}} - \{c_i\}_{i=1}^{|\mathbf{c}|}$. The elements in explicit tag set are mentioned directly in the news article, while the ones in implicit tag set are not mentioned in the news article but associated with it. Our goal is to mine the underlying related tags on the news.

2.2 Model

The model is composed of BERT [6] and GCN [11]. Here, we adopt the pre-trained Chinese model, *bert-base-chinese*. We divide the training process into two parts. The first part is to train a GCN for getting the tag embeddings, and the second part is to find implicit tags by using news article and tag embeddings.

Tag Embeddings. To obtain the tag embeddings $\mathbf{T} \in \mathbb{R}^{|\mathcal{T}|\times d}$, we train a variational graph auto-encoder (VGAE) [12] based on GCN by utilizing the tag information. The tag information contains the tag attributes mentioned in Section 2.1 and the tag frequency in the dataset. We create a matrix $\mathbf{A}$, where the non-diagonal elements of $\mathbf{A}$ are the number of co-occurrence between tags, and the diagonal elements of $\mathbf{A}$ are the frequency of tags in the training data. Subsequently, we pass the information into the GCN. The following equation defines a two-layer GCN with learnable matrices $\mathbf{W}^0$ and $\mathbf{W}^1$:

$$\mathrm{GCN}(\mathbf{X}, \mathbf{A}) = \mathrm{ReLU}(\widetilde{\mathbf{A}}\mathrm{ReLU}(\widetilde{\mathbf{A}}\mathbf{X}\mathbf{W}^0)\mathbf{W}^1), \tag{1}$$

where the symmetric normalized Laplacian matrix $\widetilde{\mathbf{A}} = \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}$. By applying the different learnable weight matrices $\mathbf{W}^{\boldsymbol{\mu}}$ and $\mathbf{W}^{\boldsymbol{\sigma}}$ on the output of $\mathrm{GCN}(\mathbf{X}, \mathbf{A})$, we obtain the mean vectors $[\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_{|\mathcal{T}|}]^\top$ and the logarithm of the variance vectors $[\log\boldsymbol{\sigma}_1, \ldots, \log\boldsymbol{\sigma}_{|\mathcal{T}|}]^\top$.

$$[\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_{|\mathcal{T}|}]^\top = \mathrm{GCN}(\mathbf{X}, \mathbf{A})\mathbf{W}^{\boldsymbol{\mu}}, \tag{2}$$

$$[\log\boldsymbol{\sigma}_1, \ldots, \log\boldsymbol{\sigma}_{|\mathcal{T}|}]^\top = \mathrm{GCN}(\mathbf{X}, \mathbf{A})\mathbf{W}^{\boldsymbol{\sigma}}. \tag{3}$$

Meanwhile, we set the tag embeddings equal to the mean vectors, i.e., $\mathbf{T} = [\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_{|\mathcal{T}|}]^\top$. Referring to the vanilla VAE [10] model, the VGAE model consists of the inference model $q(\mathbf{Z}|\mathbf{X}, \mathbf{A})$ and the generation model $p(\mathbf{A}|\mathbf{Z})$.

$$q(\mathbf{Z}|\mathbf{X}, \mathbf{A}) = \prod_{i=1}^{|\mathcal{T}|} \mathcal{N}(\mathbf{z}_i|\boldsymbol{\mu}_i, \mathrm{diag}(\boldsymbol{\sigma}_i^2)), \tag{4}$$

$$p(\mathbf{A}|\mathbf{Z}) = \prod_{i=1}^{|\mathcal{T}|}\prod_{j=1}^{|\mathcal{T}|} p(A_{ij} > \delta_{ij}|\mathbf{z}_i, \mathbf{z}_j), \tag{5}$$

where $p(A_{ij} > \delta_{ij}|\mathbf{z}_i, \mathbf{z}_j) = \sigma(\mathbf{z}_i^\top \mathbf{z}_j)$ and $\sigma(.)$ is the sigmoid activation function. The threshold δ_{ij} is used to determine whether the tag t_i and t_j are related. In this paper, we set the threshold $\delta_{ij} = \frac{A_{ii}\times A_{jj}}{N}$, where N denotes the number of training samples. Also, we assume the prior probablity $p(\mathbf{Z}) = \prod_{i=1}^{|\mathcal{T}|}\mathcal{N}(0, \mathbf{I})$. The goal in this section is to minimize the cost function L_{VGAE}.

$$L_{\mathrm{VGAE}} = -\mathbb{E}_{q(\mathbf{Z}|\mathbf{X},\mathbf{A})}[\log p(\mathbf{A}|\mathbf{Z})] + \mathrm{KL}(q(\mathbf{Z}|\mathbf{X}, \mathbf{A})||p(\mathbf{Z})). \tag{6}$$

News Embedding. Before passing a news article $\mathbf{c}$ through the BERT model, we add special tokens [CLS] and [SEP] to the front and end of the news article separately. Afterward, this reprentation is fed to BERT. We take the first output of the last layer of BERT as the news embedding $\mathbf{n} \in \mathbb{R}^d$.

$$\mathbf{n} = \mathrm{BERT}[0]([\mathrm{CLS}], c_1, \ldots, c_{|\mathbf{c}|}, [\mathrm{SEP}]). \tag{7}$$

Implicit Tag Prediction. After getting the news and tag embeddings, we obtain the hidden vector $\mathbf{h} = [\mathrm{head}_1; \ldots; \mathrm{head}_K]$ by using the multi-head attention, where head_i is represented as Equation (8). In this paper, we set the number of attention heads $K = 12$.

$$\mathrm{head}_i = (\mathbf{T}\mathbf{W}_i^v)^\top \mathrm{softmax}(\frac{\mathbf{T}\mathbf{W}_i^q\mathbf{n}}{\sqrt{d}}), \tag{8}$$

where $\mathbf{W}_i^v$ and $\mathbf{W}_i^q$ are learnable matrices.

Finally, we experiment with two models, BERT-GCN and BERT-GCN-concat. For the BERT-GCN-concat model, we concatenate the hidden vector and the news embedding, and then fed the result into the fully-connected layer to make the prediction $\hat{\mathbf{y}} = \sigma(\mathbf{W}^n[\mathbf{h}; \mathbf{n}])$. As for the BERT-GCN model, the prediction is represented as $\hat{\mathbf{y}} =$

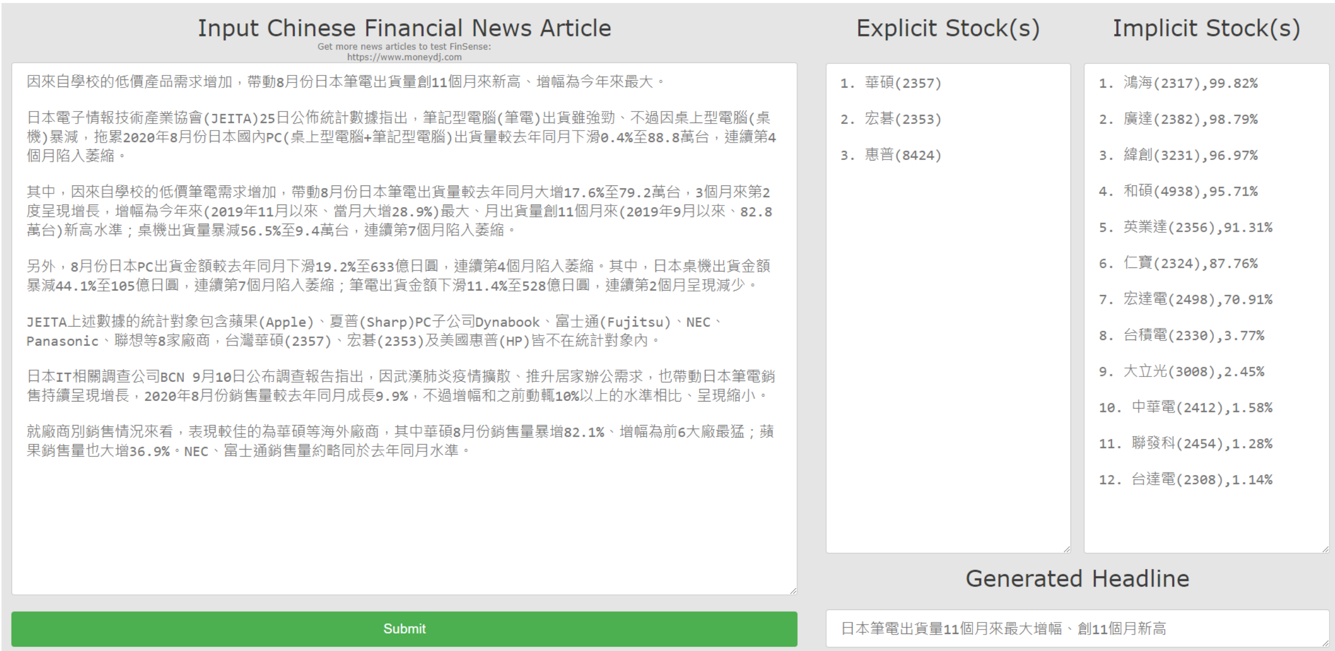

Figure 2: Screenshot of FinSense.

Table 1: Statistics of the news dataset

	Train	Dev	Test
start date	2013/06/22	2017/03/24	2017/11/13
end date	2017/03/23	2017/11/10	2018/06/20
# samples	24,640	3,076	3,088
avg. implicit tags	5.379	6.558	6.820
std. implicit tags	4.473	4.870	4.998
max. implicit tags	29	28	32
# candidate tags ($\lvert\mathcal{T}\rvert$)		693	
# attributes ($\lvert\mathcal{F}\rvert$)		633	

$\sigma(\mathbf{W}^h\mathbf{h})$. We treat the implicit prediction task as the multi-label classification task, and apply the binary cross-entropy loss L on the prediction vector $\hat{\mathbf{y}}$ and the actual vector $\mathbf{y} = [y_1, \ldots, y_{|\mathcal{T}|}]^\top$, where $y_i = \mathbb{1}_{\mathcal{I}^c}(t_i)$, and $\mathbb{1}_{\mathcal{I}^c}(.)$ is the indicator function of implicit tag set $\mathcal{I}^c$.

$$L = -\frac{1}{|\mathcal{T}|}\sum_{i=1}^{|\mathcal{T}|} y_i \log\hat{y}_i + (1-y_i)\log(1-\hat{y}_i). \tag{9}$$

3 EXPERIMENTAL RESULTS

Our dataset is collected from MoneyDJ[1], a financial newsvendor in Taiwan, and is split by time. The training set contains 24,640 instances from June 22, 2013 to March 23, 2017. The development set contains 3,076 instances from March 24, 2017 to November 10, 2017. The test set contains 3,088 instances from November 13, 2017 to June 20, 2018. Table 1 provides more detailed information about the dataset.

Table 2: Experimental results. (%)

Model	Dev	Test
BERT	61.33	52.95
BERT-GCN	68.00	59.91
BERT-GCN-concat	**68.59**	**60.95**

Our baseline model is BERT without using tag embeddings generated from the GCN model. We evaluate the experimental results by micro-averaged F1-score. Table 2 shows the experimental results. Our model achieves 60.95% of micro-averaged F1-score, and outperforms the BERT model. The results evidence that adding the graph information between the candidate tags and training with the GCN architecture is useful for the proposed task.

4 DEMONSTRATION

Figure 2 shows the screenshot of FinSense. In FinSense[2], users can input Chinese financial news article on the left side. After pushing the "Submit" button, users will get both explicit-mentioned stocks and implicit stocks related to the input news article. Note that, the number in parentheses corresponds to the ticker of the stock. The percentage along with the implicit stocks is the probability of the implicit stock. The generated headline is also proposed on the right side.

Because FinSense is trained with financial news articles from MoneyDJ, the input news article should be written in Chinese, and the recommended stocks are selected from the pool in the Taiwan

[1]https://www.moneydj.com/

[2]http://nlg6.csie.ntu.edu.tw/FinSense

stock market. This demonstration can not only provide editing assistance to the financial journalists, but also help the investors discover the implicitly-related financial instruments of the input news article.

5 REAL-WORLD APPLICATION

Predicting the stock movement is one of the focused challenges of investors. In order to show the influence of adopting implicit stock in the stock movement prediction task, we use the Hybrid Attention Network (HAN) [7] as the base model and compare the performance of only using the news explicitly mentioning the target stock with that of using the news explicitly and implicitly related to the target stock. That is, we have the following two settings:

- **Setting 1 (*S*1)**: Only explicit news is considered.
- **Setting 2 (*S*2)**: We integrate manually labeled implicit news with explicit news to determine whether information from related news aids the prediction task.

We adopt the cumulative abnormal return, which is often used to evaluate the impact of news on a stock price, as our prediction target. Given a target stock s within an n-day event window, the cumulative abnormal return (CAR) is defined as

$$CAR_{s,n} = \sum_{m=1}^{n} (R_{s,m} - \hat{R}_m) \tag{10}$$

where $R_{s,m}$ denotes the return of a specific stock on day m and $\hat{R}_m$ represents the return on an index such as the Dow Jones or the S&P 500 during the same period. We adopt CAR_3 as previous works [2, 5, 14].

Furthermore, we treat stock movement prediction as a binary classification problem. RISE and FALL classes are defined based on the polarity of cumulative abnormal return. Formally,

$$y = \mathbb{1}\left(CAR_{s,n} > 0\right) \tag{11}$$

We use accuracy (ACC) and Matthews Correlation Coefficient (MCC) as previous works [7, 15] for evaluating the results, and compared the performances of the HAN models with the following baseline models:

- Random Guess: A basic predictor randomly guesses the movement.
- BoW + Random Forest [8]: Bag-of-words (BoW) is a classical text representation. We consider the top 20,000 common words in our corpus. Therefore, in this baseline, the news collections for 10 days are encoded into a 20,000-dimensional one-hot feature vector for each training sample. Then we adopt a random forest classifier with 100 trees in the forest.
- FastText + Random Forest [9, 13]: FastText is a strong baseline method in neural network-based text representation. For the model input, we first average the daily news representations to construct a news collection representation for each day and concatenate them as input instances. We adopt a random forest classifier with 200 trees in the forest.

Our experimental results show that with the implicit news, the performance of the HAN model can be significantly improved. It supports the usefulness of the proposed system, FinSense.

Table 3: Experimental results. The asterisk (*) denotes models that significantly outperform the HAN$_{S1}$ at $p < 0.05$ (using McNemar's test).

Model	Acc. (%)	MCC
Random Guess	50.77	0.0147
BoW + Random Forest	50.97	0.0159
FastText + Random Forest	52.83	0.0485
HAN$_{S1}$	54.37	0.0788
HAN$_{S2}$	56.77*	0.1300*

6 CONCLUSION

In this paper, we present a novel task, implicit stock detection, and further propose a model, BERT-GCN, to deal with it. The experimental results show that our model performs better than vanilla BERT. The real-world application scenarios are also described and implemented by our FinSense system, assistant of financial journalists and investors. In the future, we plan to apply the proposed model to improve the performance of the algorithm trading system. Besides, the cross-market inference and the inference based on other financial documents are also worthwhile to explore in the future.

ACKNOWLEDGMENTS

This research was partially supported by Ministry of Science and Technology, Taiwan, under grants MOST 109-2218-E-009-014, MOST 109-2634-F-002-040, and MOST 109-2634-F-002-034, and by Academia Sinica, Taiwan, under grant AS-TP-107-M05.

REFERENCES

[1] Ananth Balashankar, Sunandan Chakraborty, Samuel Fraiberger, and Lakshminarayanan Subramanian. 2019. Identifying Predictive Causal Factors from News Streams. In *EMNLP-IJCNLP*. 2338–2348.
[2] Ching-Yun Chang, Yue Zhang, Zhiyang Teng, Zahn Bozanic, and Bin Ke. 2016. Measuring the Information Content of Financial News. In *COLING*. 3216–3225.
[3] Chung-Chi Chen, Hen-Hsen Huang, and Hsin-Hsi Chen. 2019. Crowd view: converting investors' opinions into indicators. In *IJCAI*.
[4] Jui Chu, Chung-Chi Chen, Hen-Hsen Huang, and Hsin-Hsi Chen. 2020. Learning to Generate Correct Numeric Values in News Headlines. In *Companion Proceedings of the The Web Conference 2020*.
[5] Angela K Davis, Jeremy M Piger, and Lisa M Sedor. 2012. Beyond the numbers: Measuring the information content of earnings press release language. *Contemporary Accounting Research* 29, 3 (2012), 845–868.
[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL*. 4171–4186.
[7] Ziniu Hu, Weiqing Liu, Jiang Bian, Xuanzhe Liu, and Tie-Yan Liu. 2018. Listening to Chaotic Whispers: A Deep Learning Framework for News-oriented Stock Trend Prediction. In *WSDM*. 261–269.
[8] Thorsten Joachims. 1998. Text categorization with Support Vector Machines: Learning with many relevant features. In *Machine Learning: ECML-98*. 137–142.
[9] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of Tricks for Efficient Text Classification. In *EACL*. 427–431.
[10] Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* (2013).
[11] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).
[12] Thomas N Kipf and Max Welling. 2016. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308* (2016).
[13] V. S. Pagolu, K. N. Reddy, G. Panda, and B. Majhi. 2016. Sentiment analysis of Twitter data for predicting stock market movements. In *SCOPES*. 1345–1350.
[14] Paul C Tetlock. 2007. Giving content to investor sentiment: The role of media in the stock market. *The Journal of finance* 62, 3 (2007), 1139–1168.
[15] Yumo Xu and Shay B. Cohen. 2018. Stock Movement Prediction from Tweets and Historical Prices. In *ACL*. 1970–1979.