The comparison and analysis of Skip-gram and CBOW in creating financial sentimental dictionary

Xingjian Zhang[1,3,4,†], **Lu Zhang**[2,†]

[1]Faculty of Science and Technology, University of Macao, Macao, China
[2]Smart Education, Jiangsu Normal University, Xuzhou, Jiangsu, China

[3]dc02748@um.edu.mo
[4]Corresponding author

[†]Xingjian Zhang, Lu Zhang contributed equally to this work and should be considered co-first authors.

Abstract. Textual analysis is increasingly used in various fields due to data availability, computing power, and machine learning techniques. In finance, sentiment analysis is essential for obtaining excess returns, and building domain-specific lexicons using word2vec is a prevalent method. The CBOW and Skip-gram algorithms have different predictive methodologies and performances depending on the task and dataset. This paper reviews financial sentiment analysis using a dictionary method and compares the performance of the two algorithms. CBOW trains faster than Skip-gram when dealing with a small amount of text data, but as the amount of data increases, Skip-gram becomes more efficient. Besides, the Skip-gram captures more synonyms of the selected words than CBOW.

Keywords: word2vec, CBOW, skip-gram, financial sentiment analysis.

1. Introduction

Automatic text analysis has been implemented in increasing numbers of applications spanning a wide range of industries thanks to the expansion in textual data availability, computer capacity, and machine learning techniques.

In the financial sector, though textual and numerical data can provide helpful assistance to decisions, mainly those numerical ones have been widely used in quantity analysis [1]. For instance, price and trade data are explored to find trading patterns, which are believed by technical views that can repeat in the future and thus can be used to predict future stock price change to obtain excess returns. The Hurst exponent, the correlation between Dow Jones daily returns and its historical data, receded from the 1990s, indicating that relying solely on numerical analysis has become less valuable [2]. One reason could be that modern media allows heterogeneous market participants to acquire information. Market participants react differently to information at different speeds, leading to price deviation from its fundamental level. Therefore, financial analysts must consider corporate textual information to obtain excess returns. One of the most essential aspects is sentimental analysis.

Investors sentiment can influence their economic actions significantly. Sentiment analysis of financial text usually starts by counting positive and negative words tagged using a sentimental dictionary. The quality of sentiment analysis relies heavily on these sentimental lexicons. Even the meaning of the same word can be completely different in different contexts. For instance, the Chinese word "庄" express no sentimental tendency in general text, while it expresses strongly negative emotion when used in the financial text, referring to controlling the supply and demand of a particular stock or security to drive up its price and make a profit. Also, every financial market is regulated differently, resulting in different terminologies usage and market-specific writing form [3]. Meanwhile, It has been proved that translating sentimental dictionaries directly from other languages is also dubious and unrealistic [4]. Hence, building domain-specific lexicons is necessary to obtain high-quality results from sentimental analysis.

A prevalent method of building a sentimental dictionary is to expand manually selected seed words using word2vec, which can capture semantic similarity and thus can perform quite well when finding synonyms [4]–[6]. Manual selection of seed words ensures professionalism, whereas using word2vec to expand word lists can improve applicability and efficiency.

CBOW (Continuous Bag-of-Words) and Skip-gram are two distinct machine-learning methods that contribute to making up Word2vec. The two algorithms have different predictive methodologies, varying performance depending on the task and dataset. While CBOW is quicker and more effective with commonly used phrases, Skip-gram is better for infrequent words. [7]. Therefore, users must understand how to select the appropriate algorithm when implementing Word2Vec. Since the sources we aim to process in generating domain-specific dictionaries are different, It is expected that these two algorithms may perform differently in capturing semantic similarity. For instance, the textual data in social media tend to be unstructured and short, while those in financial news and annual reports are relatively structured and contain more content.

Based on the above facts, we consulted relevant literature and found that the tools and progress of emotional word analysis of financial articles and data based on neural networks are mostly concentrated in the English context. In consequence, we will try to realize and optimize the sentiment word analysis for financial articles and data in the Chinese context based on the CBOW and Skip-Gram neural network algorithms, and try to enrich the analysis tools in this field.

In this paper, we review financial sentiment analysis using a dictionary method and showcase a simple comparison of the performance of 2 word2vec algorithms. The remainder of the paper is structured as follows. Section 2 demonstrates relevant research comparing these two algorithms and performing financial textual analysis utilizing a dictionary methodology. Our experiment mainly contains two parts: the first part is based on the analysis of the two algorithms in section two and section three and uses common databases such as encyclopaedia library to reproduce the conclusion; the second part is to use the financial and stock class forum data Xueqiu and Caixin were trained with CBOW and Skip-Gram respectively and made visual analysis. The experimental process and results are provided in section 4. Finally, a conclusion will be drawn based on our study.

2. Related Works

Previous research emphasizes the necessity to create specific dictionaries in different application scenarios and widely use word2vec to expand their original seed word lists.

For one thing, they have proved that domain-specific dictionaries should be regarded as the cornerstone of research. Loughran and McDonald (2016) [8] developed financial keyword lists manually. They found that their dictionary outperformed the general-purpose Harvard IV-4 sentimental dictionary when analysing the U.S. Securities and Exchange Commission portal from 1994 to 2008. A later study found that the dictionary that performed well in analysing governmental portals did not apply to social media content. Yao [9] distinguished between formal and informal lexicons using annual reports and discussions of investors, respectively.

Furthermore, creating more specific lexicons can serve to evaluate the influence of sentimental factors on other variables in specific economic events. Wang and Huang (2018)[6] used Word2vec to develop a sentimental lexicon specific to financial technology to evaluate the impact of negative news on trading volume and price of P2P products. With the assistance of word2vec, Du [4] generated a

Chinese financial sentimental dictionary using a large amount of financial news. They also built a list of political words. These words are commonly used in Chinese text and tend to be recognized positively in general analysis. However, they have generally weaker implications for returns, which embody a "media bias" in political entities or nouns. In this process, we are more concerned about the effectiveness and efficiency of these two algorithms in various corpus with different structures and contents.

In linguistics, language is regarded as a set of lexicons and a syntactic system [10]. The information a language conveys involves not only a set of language rules, like grammar and syntax, but also effectiveness, which is usually associated with the denotation and connotation of a given text. Therefore, to represent textual data as features that can be quickly processed by a computer, semantic modelling or the numerical representation of natural language should consider both aspects. The efficiency of this process in terms of time and cost should also be considered, including minimizing storage space occupation, and maximizing training speed.

Numerous studies have explored the differences between CBOW and Skip-gram models for generating word embeddings in processing English text data. New models have also been developed by modifying these original models. Irsoy et al. [11] argues that CBOW can perform equally well as Skip-gram when a bug in the CBOW gradient update is fixed. Onishi et al. [12] proposes a method to combine both models to achieve faster learning speed and a more accurate distribution representation of words. Xiong et al. [13] suggests new models based on optimization and regression methods to enhance the performance of both CBOW and Skip-gram. However, more research is needed that compares these two models using the Chinese language in NLP.

3. Segmentation and Classification Approaches

The subject of this paper is the comparison between the skip-gram model and the CBOW model on the performance of the word segmentation of Chinese vocabulary based on financial texts. The first section introduces how to segment the text in this research, and the second introduces the text classification models.

3.1. Text segmentation method

The basis of text analysis is to segment the text, and the basis of word segmentation is the thesaurus that comes with the word segmentation tool. The jieba word segmentation is a commonly used word segmentation method in Chinese texts. However, the general thesaurus with the jieba word segmentation cannot identify the proprietary vocabularies in financial texts. Because the thesaurus of the jieba module in Python is open source, all users can add words that are not included in the thesaurus. We add some economic and financial-related proprietary vocabularies to the thesaurus of the jieba module to improve word segmentation accuracy. In the stage of text segmentation, we import the jieba module and rely on it to find the maximum word frequency combination based on word segmentation. Thus, the effect of word segmentation is significantly improved.

3.2. Text classification model

Since all words cannot be typed directly into the neural network, they need to be encoded, and their vectors can be fed into the neural network. Currently, the common word vector expression techniques include one-hot and distributed representations. On the one hand, all word vectors in the one-hot method are represented by 1 and 0, and the process of this method is relatively simple. On the other hand, distributed word vectors represent words using multiple elements, which can exist in arbitrary numbers, and the dimensions of a vector can be predetermined. Unlike one-hot methods, distributed word embeddings can help classification models learn textual features. The distance between vectors indicates the relationship's closeness, which helps the model understand the connection between words [14].

There are two training models for distributed word embeddings. One is the Continuous Bag Of Words (CBOW) model, which predicts the central word based on the context; the other is the skip-gram model, which predicts the context based on the central word. The difference between CBOW and skip-gram training architectures is shown in Figure 1. In the CBOW model, each training sample consists of multiple input features and an output; in the skip-gram model, each consists of an input and an output.

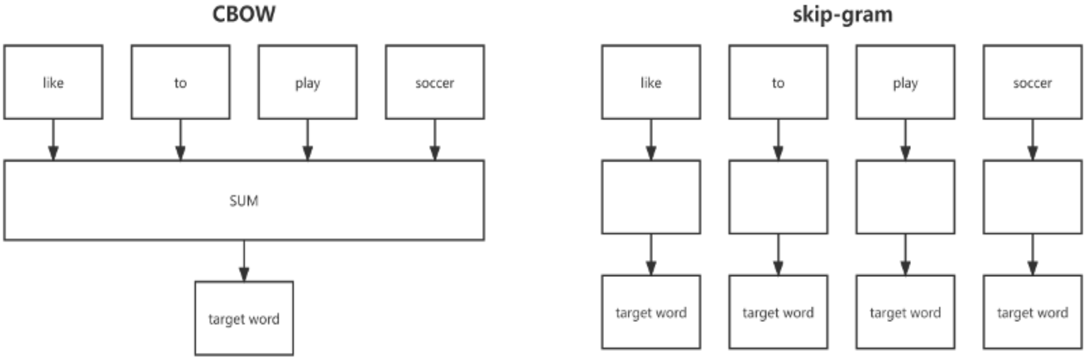

Figure 1. Shows the difference between CBOW and skip-gram training architectures.

The study by Mikolov et al. [7] introduces the CBOW model and Skip-gram model, which consist of three layers: input, projection, and output, and the training goal is for the machine to learn to predict nearby words well. The structural diagrams of the CBOW model and the skip-gram model are shown in Figure 2. Both models consist of three layers: input, projection, and output. CBOW model: the context of the word w(t) is known, that is,w(t-2), w(t-1), w(t+1), w(t+2) are known, and w(t) is unknown, to predict the word w(t). skip-gram model: w(t) is known, to predict its context, w(t-2), w(t-1), w(t+1), w(t+2).

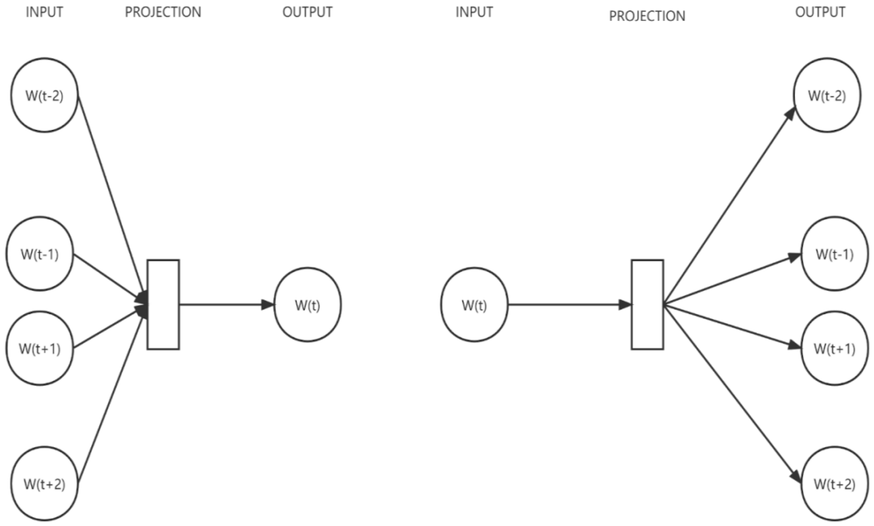

Figure 2. Shows the CBOW model(left) and The skip-gram model(right).

$$\sum_{t=1}^{T} \sum_{-c\leq j\leq c\ j\neq 0} \log P\left(W_t | W_{t+j}\right) \tag{1}$$

$$\sum_{t=1}^{T} \sum_{-c\leq j\leq c\ j\neq 0} \log P\left(W_{t+j} | W_t\right) \tag{2}$$

For a given sequence of training words W1, W2, …, Wt, the CBOW model is asked to maximize the log probability, while the Skip-Gram model is asked to maximize the log probability, and T is the size of text and c is the size of the training context.

In the skip-gram model, assuming that w is our central word, c is the word to be predicted, and θ is the model's parameter. θ consists of two matrices, u and v; u is a matrix of the context, v is a matrix of the center word, the size of both is V’×n, where V' is the size of the thesaurus, and n is the trained dimensions of the word vectors.

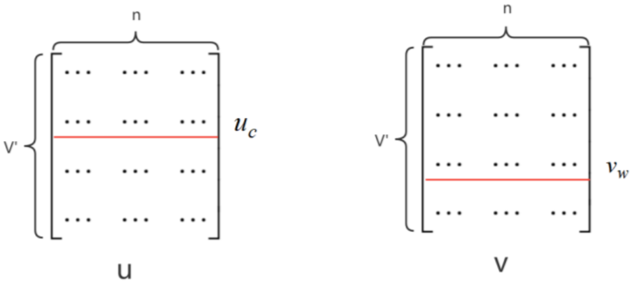

Figure 3. Shows the matrix u and the matrix v, parameters u and v are defined above.

$$P(c|w,\theta) = \frac{e^{u_c v_w}}{\sum_{c' \in V} e^{u_{c'} v_w}}$$

Its conditional probability can be expressed as follows, where u_c is the vector of the c-th row of the u matrix, and v_w is the vector of the w-th row of the v matrix, which can be regarded as the context vector of word c and the central word vector of word w respectively. c' is a word in the thesaurus other than the current context.

4. Experiment

The Experiment is organized as follows: first, we compare CBOW and Skip-Gram using a general corpus; then, we conduct the comparison on two different financial textual sources.

4.1. Dataset selection and construction

For our general corpus selection, we compiled Baidu Encyclopedia, Chinese Wikipedia, People's Daily, Xinhua News Agency, Sohu.com, Toutiao, and other mainstream media platforms from the past 1-2 years to form a raw text of almost 250,000 words. For our corpus specific to financial content, the first corpus is financial articles specific to daily opening and closing reports produced by caixin.com, which is generally regarded as a reputable source of financial and business news and analysis in China. The second corpus comprises discussions from xueqiu.com, a Chinese social media platform focused on finance and investment. It features a community aspect where users can share investment ideas and insights.

We select three sources of corpus because each corpus's text length, vocabulary, and structure vary, making it logical to create different sentiment dictionaries. Furthermore, since we assume that CBOW and skip-gram have different performances when capturing semantic similarity in various textual data, we test the performance of both algorithms in each corpus.

4.2. Construction of CBOW and Skip-gram model

First, we tried to build a skip-gram Chinese word vector neural network without the ready-made framework.

We constructed the initialized weight matrices VxD and DxV, the activation function reLu, and the normalized exponential function (softmax). We also completed the forward propagation and backpropagation equations. At the same time, we set the generated word vector dimension to 300. These structures are combined to form the primary type of skip-gram Chinese word vector network. However, we found that the Chinese word vector neural network (skip-gram) set up in this way has many disadvantages: 1. Due to the lack of CPU/GPU acceleration, the running time is too long; 2. Since the network is built using the numpy package, the data structure is not optimized, resulting in excessive memory usage at runtime. After testing, in the case of 16GB memory, it supports inputting up to 40,000 Chinese words; 3. The leading model network is not fitted with an optimizer after an iterative update, which has an unpredictable impact on the regression of the results. After the above construction and experimental discussion. We improve the whole structure of the model with Python's torch package. PyTorch has an abstract dataset class that can be indexed via functions. At the same time, PyTorch uses torch.tensor instead, which significantly improves the data processing efficiency during model training; in addition, to improve operating efficiency, we use cuda for hardware acceleration. For the setting of model parameters, although the increase in training times will improve the training effect, the time cost will also increase significantly. After weighing the two, we chose the typical size of the epoch to be ten and the learning rate to be 10^-3; If the length of each batch is set to 32, the length of the word vector is also 32. In the structure of word vectors, we use sparse vectors instead of dense vectors, which occupy less memory. Set the word vector dimension to 100 dimensions. We introduce the negative sampling algorithm and set the negative sampling number to 64. The neural network needs to use softmax to calculate the probability of each word in the corpus.

Furthermore, the dictionary's size determines that the network's weight will also be considerable, and updating the weight is very time-consuming, so we introduce negative sampling to reduce the amount of calculation. For example, when the order of magnitude of the lexicon is 10^5, the magnitude of positive samples plus negative samples can reach about 10^10, most of which are negative samples, so we randomly select a few groups from the negative samples instead of using all negative samples for gradient descent. This dramatically reduces the computational time complexity.

When building skip-gram and CBOW neural networks, we controlled the exact input text, and preprocessing method, used the same optimizer and training parameters, and used the same result processing and presentation, ensuring that only the network model itself is different.

4.3. Implement and evaluation of two models using general corpus

First, this experiment uses the general corpus to train the skip-gram and CBOW models. It uses Encyclopedia/Current Affairs News as the test data to compare the word segmentation performance of the two models. This step is expected to verify the word segmentation effect of the skip-gram model and the CBOW model under normal conditions.

After training, the word vector is projected onto a two-dimensional coordinate system for observation. Because the word vectors are too many to be observed, we selected three words about cities - "上海", "深圳", "纽约" to show the classification performance of the model. The results are shown in Figures 4, 5, and 6 below. It can be seen that for the two groups of models, the surrounding words "深圳" and "纽约" are not related, while only the country "古巴" and the region "半岛" are closely related to "上海." From this, it can be roughly seen that the word segmentation effect of CBOW and skip-gram is relatively poor for the related vocabulary of countries and regions.

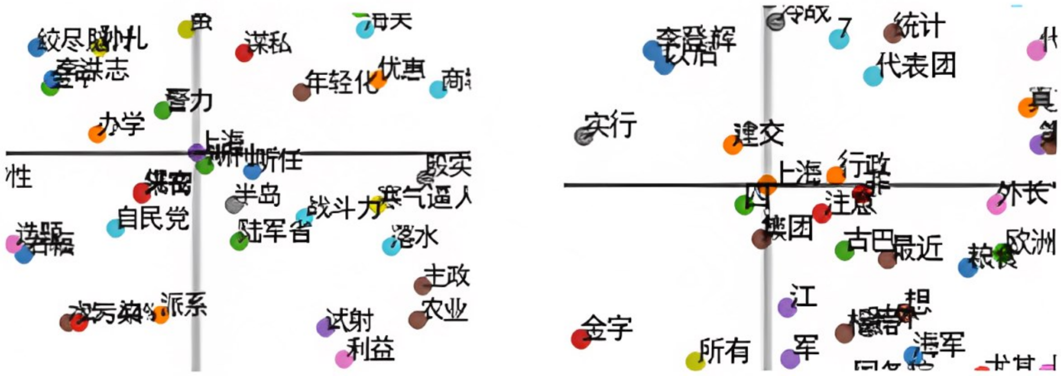

Figure 4. Shows the word segmentation effect of CBOW model and skip-gram model for “上海”.

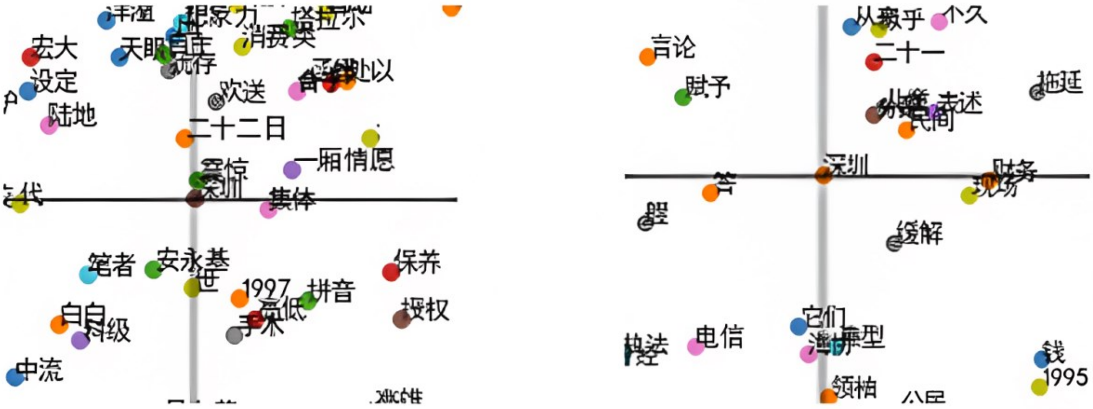

Figure 5. Shows word segmentation effect of CBOW model and skip-gram model for ”深圳”.

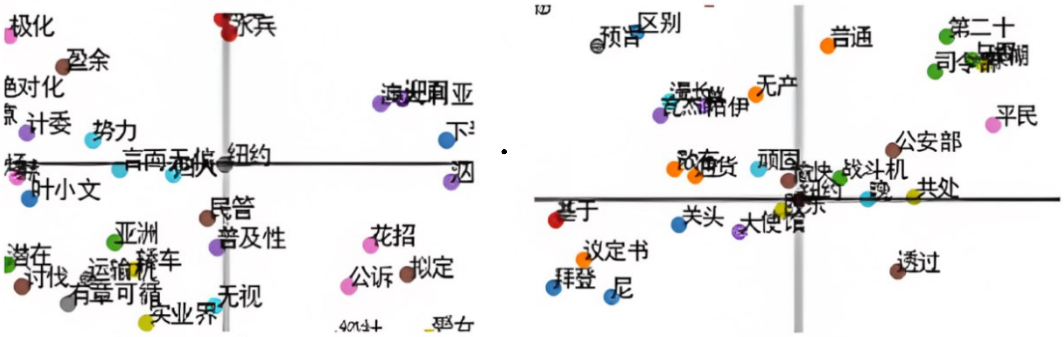

Figure 6. Shows word segmentation effect of CBOW model and skip-gram model for ”纽约”.

At the same time, this experiment also compared the total training time for the CBOW model and the skip-gram model for general texts in Table 1. The CBOW model uses the surrounding words to predict the central word. The adjustment of the surrounding words is uniform, so the number of predictions of the CBOW model is almost the same as the number of words in the entire text; however, the skip-gram model predicts the surrounding words according to the central word and constantly adjusts the word vector of the central word to traverse the complete text. Hence, the skip-gram model needs more predictions than the CBOW model. So as shown in Table 1, the skip-gram model's training time far exceeds the CBOW model's training time.

Table 1. Shows the total training time for CBOW model and skip-gram model for general texts.

Model	Time
CBOW	0:05:10.671676
skip-gram	0:21:33.794041

4.4. Comparison conducted with 2 different sources of financial text

In the second step, this experiment uses a financial-specific corpus to train the skip-gram and CBOW models. To compare their performance with different textual characteristics, we scraped 5779 discussions of the top 30 stocks in the A shares market from the Chinese social media platform, Xueqiu.com. This platform is focused on finance and investing and provides users with real-time discussion forums. These discussions are usually unstructured and contain frequent words not used in formal texts, such as in-depth financial news and listed companies' annual reports. We also scraped 100 financial articles from Caixin.com, a Chinese financial news website that covers a wide range of topics, including economics, finance, and business news. These articles consist of opening and closing summaries over three months. These articles are highly structured and contain few infrequent words compared to the first dataset scraped from social media.

Just like the method used above, based on the two data sets of Xueqiu and Caixin, we also selected common financial vocabulary "亏损", "涨", "盈利", "下跌" to judge its word segmentation effect in CBOW and skip-gram. Specifically, for Figure 7, there are no prominent related words around the central word of CBOW, and there are related financial words such as "央行" and "强股" around the central word of skip-gram; for Figure 8, there are only "经理" and "沪指" are relatively close but far away, and there are more related words around the skip-gram center word, and "盘中", "回落", "小幅", "政策" and so on have strong correlations The word distance of is very close; It is not difficult to see the relatively similar situation by analysing Figures 9, 10, 11, and 12.

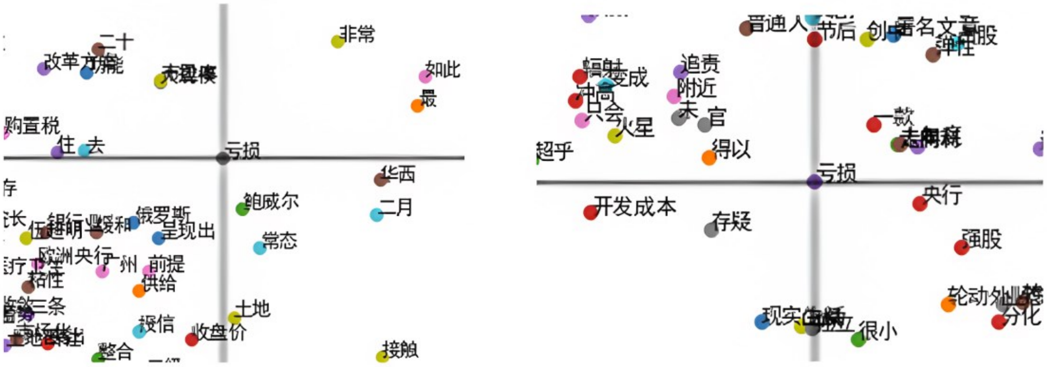

Figure 7. Shows word segmentation effect of CBOW model and skip-gram model for ”亏损”(caixin).

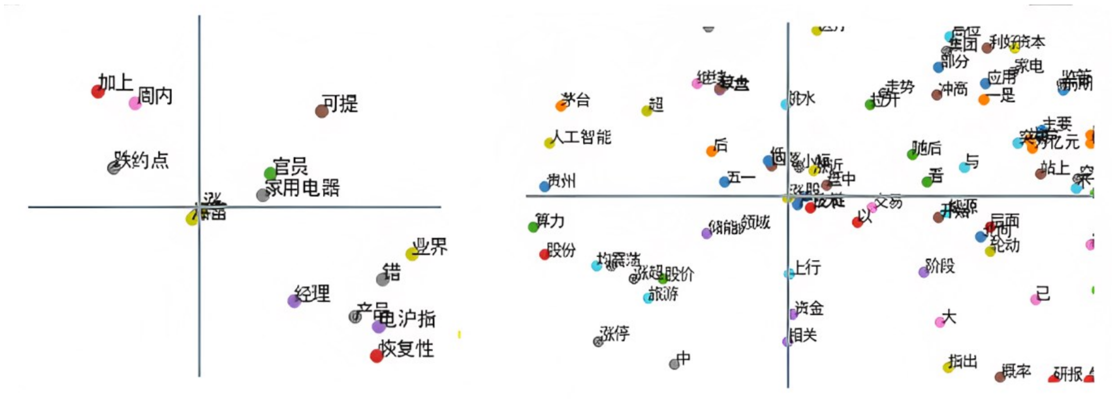

Figure 8. Shows word segmentation effect of CBOW model and skip-gram model for ”涨”(caixin).

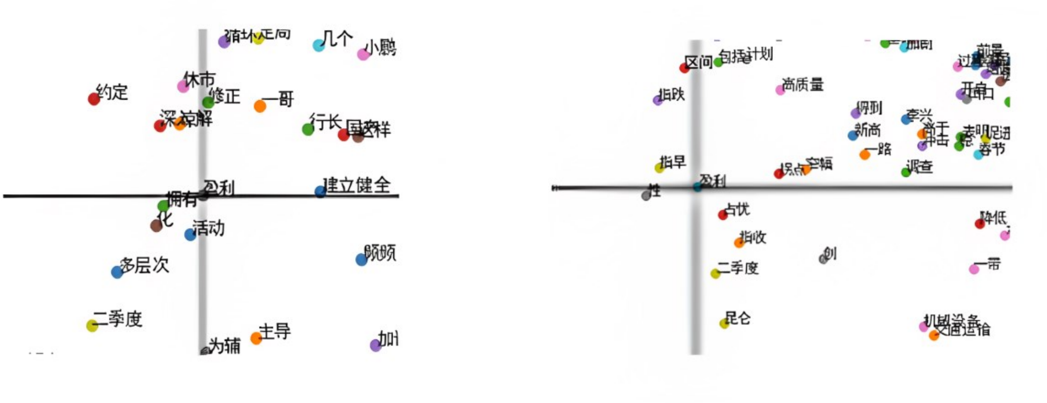

Figure 9. Shows word segmentation effect of CBOW model and skip-gram model for ”盈利”(caixin).

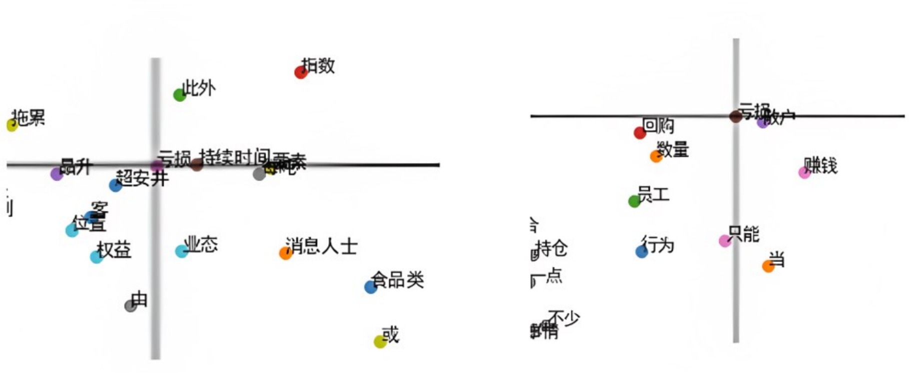

Figure 10. Shows word segmentation effect of CBOW model and skip-gram model for ”亏损”(xueqiu).

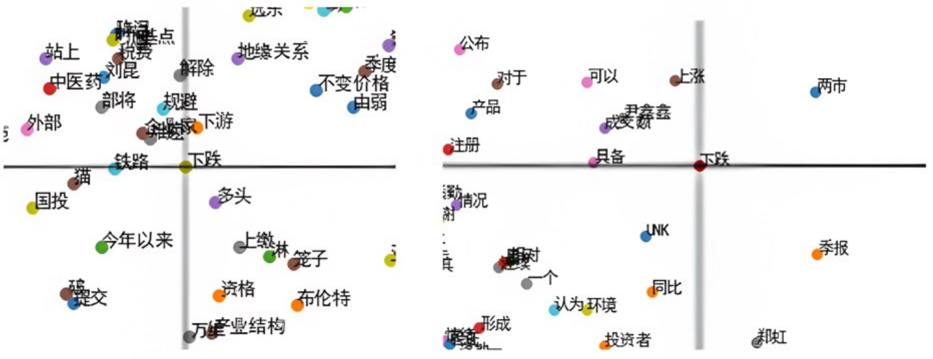

Figure 11. Shows word segmentation effect of CBOW model and skip-gram model for ”下跌”(xueqiu).

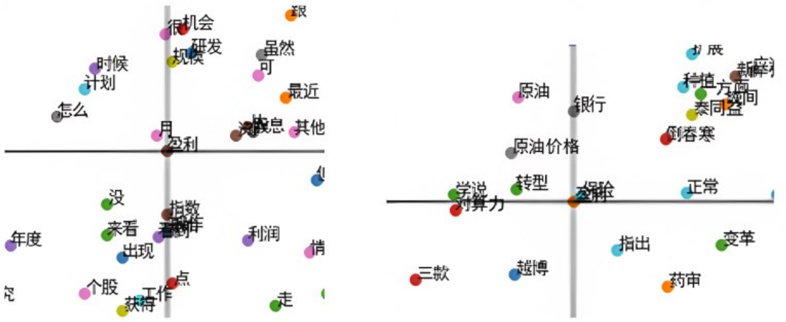

Figure 12. Shows word segmentation effect of CBOW model and skip-gram model for ”盈利”(xueqiu).

In this step, this experiment also compares the total training time for the CBOW model and the skip-gram model for financial texts in Table 2. Both Caixin and Xueqiu are financial texts. The difference is that as a corpus, the text volume of Xueqiu is much larger than Caixin. When Caixin is used as the corpus, the training time of the CBOW model is much shorter than that of the skip-gram model. However, when Xueqiu is used as the corpus, the training time of the CBOW model is much longer than that of the skip-gram model. When the corpus is small, the CBOW model has a faster training speed; but when the corpus is large, the skip-gram model has better training efficiency.

Table 2. Shows the total training time for CBOW model and skip-gram model for financial texts

Corpus	Model	Time
Caixin	CBOW	0:03:15.219547
	skip-gram	0:11:30.102163
Xueqiu	CBOW	8:32:06.945374
	skip-gram	3:55:59.245841

5. Result and Conclusion

According to the results presented in the previous section, the first noticeable thing is the training time. When dealing with small quantities of text data, CBOW can be faster to train than Skip-gram because CBOW requires less training data to learn. However, as the amount of data increases, Skip-gram becomes more efficient due to its ability to handle rare words and more diverse contexts.

In the case of the Caixin corpus, it is evident that the Skip-gram model captures more synonyms of the selected words than CBOW. For example, when counting the synonyms of "上涨" in the vector space, Skip-gram presents "涨超", "涨停", "上行", "站上", and "冲高". These words describe a similar situation to "上涨" in formal financial text. It also captures antonyms of the selected words usually used under the same language rules and context to describe the opposite situation. Furthermore, we also find some names of stocks and their corresponding sector in these verb words, pointing to the subjects of these verbs, which presents the possibility of using Skip-gram to mine financial news text and make predictions.

In the case of the Xueqiu corpus, it is difficult to determine which model is better due to their unremarkable performance. Possible explanations include that the data may need more structure for word2vec to recognize features and capture similarities. Increasing the training data quantity could be a solution, but the existing text volume is already up to two million words.

In this situation, a pre-trained model like BERT might be a better solution than Word2vec. The main idea of these pre-trained language models is to train a neural network on a large corpus of text data to learn the statistical patterns of natural language and then fine-tune the model on a specific downstream task with a smaller labeled dataset. This pre-training allows these models to generate high-quality embeddings even when working with smaller datasets, which is often the case with social media data.

These abilities make pre-trained models a promising solution for natural language processing tasks like unstructured social media discussions.

Though pre-trained language models perform powerfully and deal with large datasets, WordVec can be trained on smaller amounts of data and produce good-quality embeddings. It is a user-friendly and computationally efficient alternative to the pre-trained language model for individual users.

References

[1] F. Z. Xing, E. Cambria, and R. E. Welsch, “Natural language based financial forecasting: a survey,” *Artif. Intell. Rev.*, vol. 50, no. 1, pp. 49–73, Jun. 2018, doi: 10.1007/s10462-017-9588-9.

[2] Guozhi Wang, “Notice of Retraction: Time-dependent Hurst exponent in financial time series in China financial market,” in *2010 2nd International Conference on Advanced Computer Control*, Shenyang: IEEE, Mar. 2010, pp. 87–89. doi: 10.1109/ICACC.2010.5486883.

[3] A. Huang, W. Wu, and T. Yu, “Textual analysis for China’s financial markets: a review and discussion,” *CFRI*, vol. 10, no. 1, pp. 1–15, Sep. 2019, doi: 10.1108/CFRI-08-2019-0134.

[4] Z. Du, A. G. Huang, R. R. Wermers, and W. Wu, “Language and Domain Specificity: A Chinese Financial Sentiment Dictionary,” *SSRN Journal*, 2020, doi: 10.2139/ssrn.3759258.

[5] N. Oliveira, P. Cortez, and N. Areal, “The impact of microblogging data for stock market prediction: Using Twitter to predict returns, volatility, trading volume and survey sentiment indices,” *Expert Systems with Applications*, vol. 73, pp. 125–144, May 2017, doi: 10.1016/j.eswa.2016.12.036.

[6] Jingyi Wang and Yiping Huang, "Characterization of Financial Technology Media Sentiment and Its Impact on Online Lending Market," Economics (Quarterly), vol. 17, no. 4, pp. 1623–1650, 2018, doi: 10.13821/j.cnki.ceq.2018.03.15.

[7] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, “Distributed Representations of Words and Phrases and their Compositionality.” arXiv, Oct. 16, 2013. doi: 10.48550/arXiv.1310.4546.

[8] T. Loughran and B. Mcdonald, “Textual Analysis in Accounting and Finance: A Survey: TEXTUAL ANALYSIS IN ACCOUNTING AND FINANCE,” *Journal of Accounting Research*, vol. 54, no. 4, pp. 1187–1230, Sep. 2016, doi: 10.1111/1475-679X.12123.

[9] Yao Jiaquan, Feng Xu, Wang Zanjun, Ji Rongrong, and Zhang Wei, "Tone, Sentiment, and Market Impact: Based on a Financial Sentiment Dictionary," Journal of Management Science, vol. 24, no. 5, pp. 26–46, 2021, doi: 10.19920/j.cnki.jmsc.2021.05.002.

[10] K. Bühler, "Sprachtheorie," in *Sprachwissenschaft*, L. Hoffmann, Ed., DE GRUYTER, 2010, pp. 84–104. doi: 10.1515/9783110226300.1.84.

[11] O. Irsoy, A. Benton, and K. Stratos, "kōan: A Corrected CBOW Implementation," *ArXiv*, 2020, Accessed: May 23, 2023. [Online]. Available: https://www.semanticscholar.org/paper/k%C5%8Dan%3A-A-Corrected-CBOW-Implementation-Irsoy-Benton/7f410c00d08abcfd81a0d5516972b047462871e0

[12] T. Onishi and H. Shiina, "Distributed Representation Computation Using CBOW Model and Skip–gram Model," *2020 9th International Congress on Advanced Applied Informatics (IIAI-AAI)*, pp. 845–846, Sep. 2020, doi: 10.1109/IIAI-AAI50415.2020.00179.

[13] Z. Xiong, Q. Shen, Y. Xiong, YijieWang, and W. Li, "New Generation Model of Word Vector Representation Based on CBOW or Skip-Gram," *Computers, Materials & Continua*, vol. 60, no. 1, pp. 259–273, 2019, doi: 10.32604/cmc.2019.05155.

[14] X. Zhang, J. Zhao, and Y. LeCun, "Character-level Convolutional Networks for Text Classification." arXiv, Apr. 03, 2016. doi: 10.48550/arXiv.1509.01626.