博士論文

Development of Interpretable Neural Networks for Document-level Sentiment Analysis
(文書極性分類タスクにおける解釈可能なニューラルネットワークモデルの構築)

伊藤友貴

Abstract

Deep neural networks are powerful for text sentiment analysis; however, in the real world, they cannot be used in situations where explanations are required owing to their black-box property. In response, we propose two basic learning strategies for developing interpretable NNs called Lexicon Initialization Learning (LEXIL) and Joint Sentiment Propagation (JSP) learning. We then practically apply these methods to the development of several interpretable NNs, namely, Sentiment Interpretable Neural Network (SINN), Sentiment Shift Neural Network (SSNN), Gradient Interpretable Neural Network (GINN), and Contextual Sentiment Neural Network (CSNN). Using real textual datasets, we experimentally demonstrated that the developed NNs with our learning strategy had both the high explanation ability and high predictability. In addition, as an application of this study, we develop two types of text-visualization framework called Conceptual Sentiment Cloud Visualization (CSCV). These text-visualization frameworks should be valuable in the industry.

Contents

List of Figures

List of Tables

Part I

Introduction

Chapter 1

Introduction and Background

1.1 Background

This section describes the two crucial problems for text sentiment analysis, namely, the necessity of interpretable neural networks and the necessity of user-friendly text-visualization system.

1.1.1 Necessity of Interpretable Neural Networks Massive web documents such as micro-blogs and customer reviews are useful for public opinion sensing and trend analysis. The sentiment analysis approach (i.e., to automatically predict whether a review is overall positive or negative) has been commonly used in this area. Deep neural networks (DNNs) are some of the best-performing machine learning methods [30]. However, DNNs are often avoided in cases where explanations are required because these networks are generally considered as black-boxes. Thus, developing a high predictable neural network (NN) model that can explain the process of its prediction process in a human-like way is a critical problem. In the development of such NN model, we should consider how humans usually judge the positive or negative polarity of each review. As described in some previous linguistic researches [35, 49], it is well known that humans judge the positive or negative polarity of each review by extracting the following word-level original sentiment, word-level global contextual sentiments, and document-level sentiment, as shown in Fig. 1.

Word-level original sentiment: this sentiment describes the word-level sentiment scores before considering contexts. The sentiment scores in a word sentiment dictionary [17] corresponds to this type of sentiments. This definition is according to the work in [62]. The sentiment that each word in a document originally has (e.g., scores in a word sentiment dictionary [17]). For example, in the following sentences, bull and clean originally have positive meanings. Therefore, the word-level original sentiments of them are positive.

(1) In total, we are in a *bull* market.
(2) This room is not *clean*.

Word-level local contextual sentiment this sentiment describes word-level sentiment after considering word-level original sentiment and local word-level context.

Here, local word-level context means whether the sentiment of each word is shifted or not by polarity shifters [48, 49]. According to [48], "Polarity shifters are content words such as verbs, nouns or adjectives that influence the sentiment polarity of an expression in ways similar to

negation words." (p. 2517). For example, in the following two sentences, "not (negation)" and "failed" are examples of polarity shifters.

(3) This room is not *clean*.

(4) I *failed* to success.

Word-level global contextual sentiments We define word-level global contextual sentiment as word-level sentiment score after considering word-level original sentiment and local and global word-level contexts.

Here, Global word-level context means whether the sentiment of each word is important or not in a global context. For example in the following sentence, "bull market" and "not clean" are especially important points for deciding the overall document-level sentiment. Therefore, in terms of the global word-level context, these words should be important and the others should not be important.

(5) We are in a *bull market*.

(6) This room is *not clean*.

We define the above definition following the previous works for sentiment analysis [16, 66].

Document-level sentiment The prediction results for positive or negative sentiment tags of reviews.

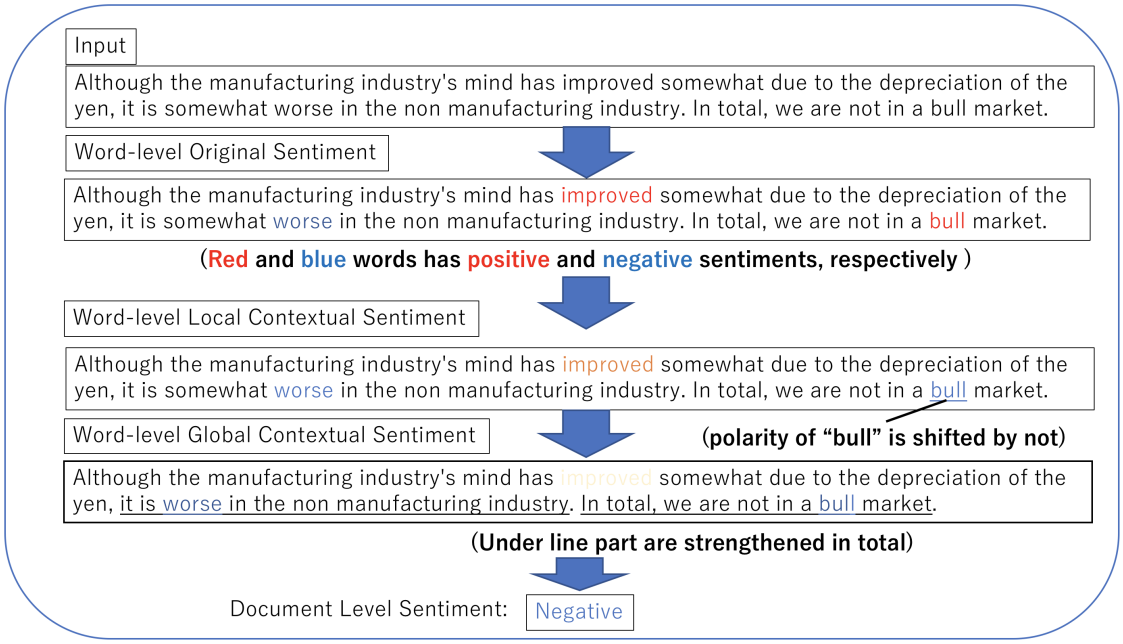

Figure 1: A possible explanation manner for the document-level sentiment analysis

Therefore, to explain the prediction results in a form that humans feel natural and agreeable, we need to use the above four types of sentiments as shown in Figure 1:

1.1.2 Requirement for the flexible framework We believe this explanation scheme should be valuable in real business situations. According to personal communication with four financial professionals, this explanation scheme sufficiently satisfies the requirements of financial document explanations. In financial documents, the recognition of word-level original sentiment and sentiment shift is important. However, the recognition of them is difficult for non-experts because they lack the specific knowledge for financial domain. For example, For example, the word "climb" generally has a neutral sentiment; however, in the financial realm, it often refers to an increase in stock prices and, thus, has a positive word-level original sentiment for investors. As for other example, the meaning of tax increase can be positive for government side, although it can be negative for consumers. Financial professionals can understand the above sentiments, however, they can be difficult for non-experts. In other words, accurate understanding of the word-level original sentiments and local word-level contexts is leading to the agreement of financial professionals. Therefore, explanation using the above three types of sentiments and two types of contexts can satisfy the requirements for the explanation because the presentation of the local word-level contexts solve the difficulty in Sentiment shift recognition difficulty and word-level original sentiment recognition difficulty. We believe the explanation scheme as described in Fig. 1 should satisfy the requirements of other domain document explanations in a similar manner. Therefore, this type of explanation should be valuable in real business situations.

In addition, it should be noted that the required explanation can be changed according to the situations. In some situations, it can be required to explain the document-level sentiment analysis result using only the Word-level original sentiment and Word-level sentiment score (= whether the sentiment of each word is shifted or not by the context). In the other situations, it can be required to analyze the word-level contextual sentiment with the explanation using the Word-level original sentiment, word-level local context, and word-level global context. Moreover, it can be possible that the Word-level original sentiment, word-level local context, word-level local contextual sentiment, and word-level global context, word-level global contextual sentiment, and *concept-level contextual sentiment* are required in the explanation for the document-level sentiment analysis.

Therefore, it is a crucial issue to establish a basic strategy for developing NNs that can explain its predictions using the required scores. However, such strategy is yet to be established, as far as we know. Many studies have been done to address the black-box property of the NNs [2,15,21,27,34,44,54,56,58,64]; however, it is hard to say that these previous works can realize the interpretability in the form that humans can find natural and agreeable because these previous studies alone can not respond to the flexibility for the requirement of the explanation. For example, interpretable NNs with attention mechanism [44, 64] can describe the global important point of each term in a review; however, they cannot describe the other three types of word-level sentiment scores. Interpretable NNs that include word-level original sentiment scores (i.e., original sentiment interpretable NN) [21, 34, 58] can describe the word-level original sentiment scores; however, they cannot describe the word-level global and local contextual sentiment scores. As for other approaches, methods for interpreting NNs can describe the word-level global sentiment scores [2, 15, 27, 54, 56]; however, they cannot describe the other scores.

1.2 Purpose

In response to the above necessity, this thesis aims to establish a basic strategy for developing NNs that can explain its predictions using the required scores. Considering the requirement for flexibility to the requirement in the explanation, the strategy should be basic and flexible, which means it can be applied to several types of interpretable neural networks directly or indirectly. Here, we define that an interpretable NN should represent the NN in which each layer correctly represents the corresponding scores and the scores of the layers in the NN directly conclude to the prediction results of the NN.

In addition, as an application of this study, we aim to develop a user-friendly text visualization framework for the real business content. The success of this application should satisfy the usefulness of this study in real business situations.

1.2.1 Development of interpretable NNs for Sentiment Analysis

Establishment of Basic Learning Theory As discussed in the above, this study first aims to propose a basic learning strategy for developing this type of interpretable NNs. To achieve this aim, we first define an interpretable NN in an abstract way; we then theoretically analyze the required conditions and derive the specific techniques for realizing the interpretability of each layer in a theoretical way. After that, using these specific techniques, we propose two types of basic learning called Lexicon Initialization Learning (LEXIL) and Joint Sentiment Propagation (JSP) learning. These proposed learning techniques can be used in accordance with the architecture of neural networks, flexibly. We then theoretically discuss the validity of these approaches. Our theoretical analysis shows that the proposed approaches can work in a situation where several necessary conditions are satisfied (= in an ideal case).

Development of Interpretable Neural Network We then demonstrate the availability of the proposed LEXIL and JSP learning using real textual datasets. More concretely, we apply the proposed learning strategy to the development of four types of interpretable NNs, namely, sentiment interpretable neural network (SINN) [24], sentiment shift neural network (SSNN) [23], gradient interpretable neural network (GINN) [21], and contextual sentiment neural network (CSNN) [22], in a practical manner. Here, the originally established theory is expected to be unavailable in real situations. Therefore, we propose practical learning techniques for developing interpretable NNs by utilizing the established theory in a practical way.

1.2.2 Application: Development of User-friendly Text Visualization Framework

In addition, as an application of this study, we apply the developed interpretable NNs into the Conceptual Sentiment Cloud Visualization (CSCV), that is the text visualization framework that can visualize the customer reviews of products or shops in a form that users can quickly understand the overview of the reviews. This application property of our research demonstrates the potential of our research for industrial usage.

1.3 Contribution

The main contribution of this thesis is summarized as follows.

- We propose a basic learning strategy for developing interpretable neural networks. This

basic strategy can be utilized in several interpretable neural networks. This application of basic learning theory to several cases is the beneficial point of this thesis because our proposed approach can be considered to be utilized in several cases.

- We experimentally demonstrate that the proposed basic learning theory can be applied to several interpretable neural networks in accordance with the required interpretability. To realize the interpretability of these NNs, we propose their learning strategies in the form of applying the basic learning theory. We experimentally evaluate our approach using real textual datasets. The developed NNs developed with our approach outperformed some DNNs in sentiment analysis tasks, even though their explanation ability was sufficiently high.

- As an application of this study, we develop several types of text visualization frameworks that should be useful in a situation where we want to catch-up with a summary of a large volume of product reviews. This application demonstrates that our research can lead to solving industrial issues.

1.4 Structure of this thesis

This remainder of this thesis is structured as follows.

This remainder of this thesis is structured as follows.

In chapter 2, related works for the interpretable neural networks for sentiment analysis and text visualization are reviewed. In Part II, we establish the method for developing an interpretable neural network for sentiment analysis. We first propose a basic learning strategy for developing interpretable neural networks and derives the necessary conditions to realize such neural networks in Chapter 3. In Chapters 6, 4, 5, and 7, we propose methods for practically realizing several interpretable NNs by corresponding the proposed basic learning theory into several cases, and then experimentally evaluate them using real textual datasets. In part III, as an application of this study, we develop a text-Visualization Framework for Catching-up the summary of large customer reviews. Part IV concludes this thesis.

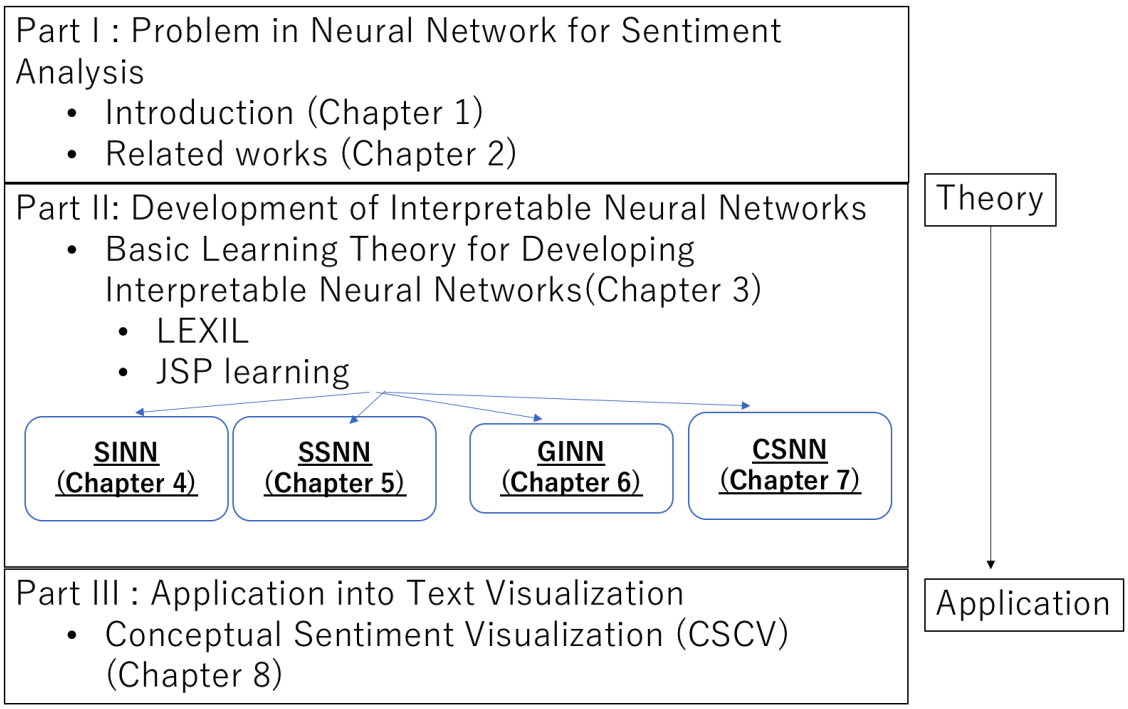

Figure 2: Structure of Thesis

Chapter 2

Related Works

In this chapter, we introduce the related works for sentiment analysis. In section 2.1, we introduce the previous approaches for sentiment analysis, and discuss the limitation of these approaches. In Section 2.2, we introduces relevant topics of research that are related to the interpretablity of the neural networks for sentiment analysis.

2.1 Related works for sentiment analysis

This section reviews the previous works for sentiment analysis. Generally, document-level sentiment analysis methods can be divided into the following two categories, namely, knowledge based approach and machine learning approach.

2.1.1 Knowledge based approach As for traditional methods for document-level sentiment analysis, we can describe the methods using word sentiment dictionary or polarity list [17,37,67]. Using these word sentiment dictionaries or polarity lists in an effective way, we can analyze many types of sentiments such as document-level and word-level. For example, in [59], they propose a method for analyzing the risk of companies using a finance specific sentiment lexicon [37]. Methods using sentiment lexicon are promising; however, this type of approach has the two serious problems.

The first problem is that the creation of a large volume of sentiment lexicons requires a large volume of manpower because these lexicons are generally created manually. As a result, it is haed for these dictionary based approaches to analyze specific domain documents. To solve this problem, many studies have been done for automatic creation of polarity lexicons or sentiment dictionary [3,20,34,43,58]. In [20,34,58], several methods for creating word-sentiment dictionary using the relationship between word frequency and document-level sentiment tags. Moreover, in [19], they apply a revised method for calculating word embeddings considering antonym–synonym distinction [43] to the creation of word polarity list. This method requires only a large text corpus, the synthetic analysis results of it, and a small size of polarity lexicons.

The second problem is that these dictionary based approaches cannot address the sentiment shift such as “not” in “not good” and “fail” in “fail to complete.” Therefore, they often fail to analyze word-level local or global contextual sentiment. To solve this problem, There have been many previous researches to address this problem. Some methods detect sentiment shift or contextual sentiment using the supervised learning methods with annotated contextual sentiment

tags [40, 41, 47]. Some methods address this task using specific knowledge or rules for sentiment shifts [31, 35, 62]. As for other approaches, in [48], the method for for obtaining the sentiment shifters in a boot strappping manner with a few seeds was proposed. These collected sentiment shifters can be useful for analyzing word-level contextual sentiment [48]. In this topic, a Recursive Neural Network [53] is one of the states of the art methods. Recursive Neural Network assigns the sentiment score to each node in a synthetic tree. Therefore, using this Recursive Neural Network, we can detect sentiment shifts and analyze word-level contextual sentiment. Recursive Neural Network can be developed using the sentiment treebank dataset, which is the annotation dataset for the sentiment of each node in a synthetic tree.

2.1.2 Machine learning approach As for other categories for sentiment analysis method, we can describe machine learning approach [9, 16, 18, 30, 57, 66]. This type of approach develop a prediction model using documents and their sentiment tags. Due to the rapid progress of deep neural networks (DNNs), approahces with DNNs [9, 16, 30, 57, 66] outperform the methods in knowledge based approach in most cases, and they are the state-of-the-art-methods.

2.1.3 Problem in previous approaches Knowledge based approaches are useful for analyzing document-level sentiment analysis in an interpretable manner because they can detect word-level original sentiments and sentiment shifts; however, they alone cannot assign the global word-level contexts. In addition, they basically require the synthetic analyzer such as spacy[1], annotated contextual sentiment tags, or other specific knowledge. This causes the strong limitation of these methods because they can not be available for non-grammar documents, miner languages, or domain specific documents. Therefore, analysis methods that require little knowledge or few rules should be required. Moreover, these days, it is experimentally demonstrated that deep neural networks with attention mechanisms [16, 57, 66] outperform these methods in document-level sentiment analysis.

On the other hand, DNNs also have a crucial problem, that is, they are black-box functions. In this sense, interpretable methods with high prediction ability or methods for interpreting DNNs should be required.

2.2 Related works for the interpretability in DNNs

As for research of addressing the black-box property of DNNs, we can describe two types of approaches, namely, interpretation of Neural Networks [2, 15, 27, 46, 52, 56] and development of interpretable Neural Networks [10, 16, 34, 44, 58, 64, 66].

2.2.1 Interpretation of Neural Network As for useful techniques in the black-box property of the DNNs, the methods for interpreting deep neural networks (DNNs) can be described [2, 15, 27, 46, 52, 56]. Several methods [2, 15, 27, 52, 56] calculated the gradient score of each input feature in the prediction and visualized an important feature in their predictions. The LRP method is one of the state-of-the-art methods. Other methods [46] analyze the important feature in a prediction by analyzing the relationship between the input features and output value. These methods are useful for understanding the important feature or point in the prediction process. However, these methods alone can not explain the prediction process in more complex forms,

[1] https://spacy.io/

that is, they alone can not explain the process of prediction in a form like Figure 1.

2.2.2 Interpretable Neural Network As for other approaches, the methods for developing interpretable NNs are also described [10, 16, 34, 44, 58, 64, 66]. For example, interpretable NNs with the attention mechanism [10, 16, 44, 64, 66] can visualize the global important point in the prediction process. As for other approaches, we can describe the interpretable NNs that include the layers that represent original word-level sentiment [34, 58] can be described.

However, there is a serious problem in these methods. First,the former approaches with attention mechanisms can not produce the three types of sentiment scores and word-level local contexts. In addition, in recent researches, it was described that attention scores assigned by these methods were not always agreed to the humans' feeling [25, 51]. In fact, in the experiment of chapters 4–7, the global important point scores produced by some NNs with the attention mechanism were not agreed to the humans' feeling. Moreover the crucial problems are also included in the latter approaches with the interpretable NNs that include the original word-level sentiment layers. They can not produce the local and global word-level contexts and word-level local and global contextual sentiments. Therefore, they can not satisfy the required interpretability.

2.3 Summary

In this chapter, we review the previous approaches for document-level sentiment analysis and those for addressing black-box property of DNNs. From the results of reviewing, we can see that one of the promising directions for developing the document-level sentiment analysis methods with both interpretability and predictability is addressing the black-box property of DNNs. Unfortunately, satisfactory methods are not established in this topic. Therefore, the development of methods for addressing the interpretability of DNNs that can satisfy the required interpretability is one of the important topics in this sentiment analysis area.

Part II

Development of Interpretable Neural Networks

Chapter 3

Basic Learning Theory for Developing Interpretable Neural Networks

3.1 Introduction

As discussed in Chapter 2, there are many works for the interpretability of neural networks; however, there have been little works for the theoretical strategy for making the hidden layers interpretable, that is, making the hidden layers represent the corresponding sentiment scores as shown in Figure 3. In response, this chapter aims to derive the basic learning theory for developing interpretable NNs that includes WOSL, SSL, LWCSL, GIL, and GWCSL and outputs the document-level sentiment as a prediction result (Figure 3). We consider the above setting because we believe that word-level original sentiment, sentiment shift, global importance, and local and global word-level contextual sentiment scores are basic and crucial for considering the sentiments in reviews following some previous works [35, 49].

We define a set of the above types of neural network models as Base Interpretable Neural Networks (BINNs), and this chapter aims to derive the learning theory for realizing the interpretability in BINNs.

To achieve this aim, we first define the BINNs more concretely and then consider some assumptions for the relation among word-level sentiment scores to design a problem setting. We then provide some necessary conditions for BINN and propose three types of novel learning strategies called Lexical Initialization Learning (LEXIL) and Joint Sentiment Propagation Learning (JSP learning) for realizing the BINN. These methods utilize one or two specific techniques among the following three techniques:

- Lexical Initialization: the initialization strategy for WOSL with a prepared sentiment dictionary,
- SSL regularization: the regularization strategy for SSL.

LEIXIL utilizes only *Lexical Initialization*, JSP learning utilizes only *Lexical Initialization* and *SSL regularizaion*, and These techniques can be flexibly available in accordance with cases or NN architectures.

We theoretically analyze the effects of these approaches and analyze whether the proposed learning strategies are theoretically valuable or not for realizing the interpretability in BINN.

The remain of this chapter is constructed as follows. Section 3.2 defines the problem setting and Section 3.3 discuss the assumptions, conditions, and ideas for achieving the interpretability of BINN. Section 3.4 describes the proposed learning strategies for realizing the interpretable BINN. Section 3.5 theoretically analyze the property of our approach, and Section 3.6 concludes this chapter.

3.2 Definition of Basic Interpretable Neural Network

To consider the learning theory, concretely, we define the layers of a BINN (= a certain element of BINNs) in the following way.

Notation

We first define several symbols. Let $\Omega^{tr} = \{(\mathbf{Q}_n, d^{\mathbf{Q}_n})\}_{n=1}^{N}$ be a training dataset where N is the training data size, $\mathbf{Q}_n$ is a review, and $d^{\mathbf{Q}_n}$ is its sentiment tag (1 is positive and 0 is negative). Assume that each review $\mathbf{Q}_n$ has L sentences and each sentence contains T words. $w_{it}^{\mathbf{Q}_n}$ represents the tth word in the ith sentence. After the SINN has been developed, it can analyze word-level contextual sentiment with explaining its analysis result, as shown in Figure 5. Let $\{w_i\}_{i=1}^{v}$ be the terms that appear in a text corpus, v be the vocabulary size, and $I(w_i)$ be the vocabulary index of word w_i where $I(w_i) = i$. Let $\boldsymbol{w}_i^{em} \in \mathbb{R}^e$ be an embedding representation of word w_i where $\|\boldsymbol{w}_i^{em}\|_2 = 1$, and the embedding matrix $\boldsymbol{W}^{em} \in \mathbb{R}^{v \times e}$ be $[\boldsymbol{w}_1^{emT}, \cdots, \boldsymbol{w}_v^{emT}]^T$ where e is the dimension size of the word embeddings. $\boldsymbol{W}^{em}$ is constant and obtained using the skip-gram method [39] and the text corpus in a training dataset.

Structure of BINN

This section defines the layers in BINN, respectively. The WOSL, SSL, LWCSL, GIL, and GWCSL are difined in the following way.

WOSL Given a review $\mathbf{Q} = \{\{w_{it}^{\mathbf{Q}}\}_{t=1}^{T}\}_{i=1}^{L}$, this layer converts the words $\{\{w_{it}^{\mathbf{Q}}\}_{t=1}^{T}\}_{i=1}^{L}$ to word-level original sentiment representations $\{\{p_{it}^{\mathbf{Q}}\}_{t=1}^{n}\}_{i=1}^{L}$ in a word sentiment dictionary form as

$$p_{it}^{\mathbf{Q}} = w_{I(w_{it}^{\mathbf{Q}})}^{p} \tag{1}$$

where $\boldsymbol{W}^p \in \mathbb{R}^v$ represents the original sentiment scores of words, and w_i^p is the ith element of $\boldsymbol{W}^p$. The w_i^p value corresponds to the original sentiment score of the word w_i.

SSL This layer represents their word-level sentiment shift scores $s_{it}^{\mathbf{Q}}$ using terms and their Surrounding terms as follows:

$$s_{it}^{\mathbf{Q}} := SSL(\boldsymbol{e}_{it}^{\mathbf{Q}}, \{\boldsymbol{e}_{it}^{\mathbf{Q}}\}_{t=1}^{T}). \tag{2}$$

where $\boldsymbol{e}_{it}^{\mathbf{Q}}$ is the embedding representation of word $w_{it}^{\mathbf{Q}}$, $SSL(\cdot) \in [-1, 1]$ and $s_{it}^{\mathbf{Q}}$ denotes whether the sentiment of $w_{it}^{\mathbf{Q}}$ is shifted ($s_{it}^{\mathbf{Q}} < 0$) or not ($s_{it}^{\mathbf{Q}} \geq 0$). Here, $SSL(w_{it}^{\mathbf{Q}}, \{w_{it}^{\mathbf{Q}}\}_{t=1}^{T})$ is a function calculated by term $w_{it}^{\mathbf{Q}}$ and sentence $\{w_{it}^{\mathbf{Q}}\}_{t=1}^{T})$ in a review $\mathbf{Q}$.

WLCSL This layer converts the values in WOSL and SSL into the word-level local contextual sentiment representations $c_{it}^{\mathbf{Q}}$:

$$c_{it}^{\mathbf{Q}} := p_{it}^{\mathbf{Q}} \cdot s_{it}^{\mathbf{Q}}. \tag{3}$$

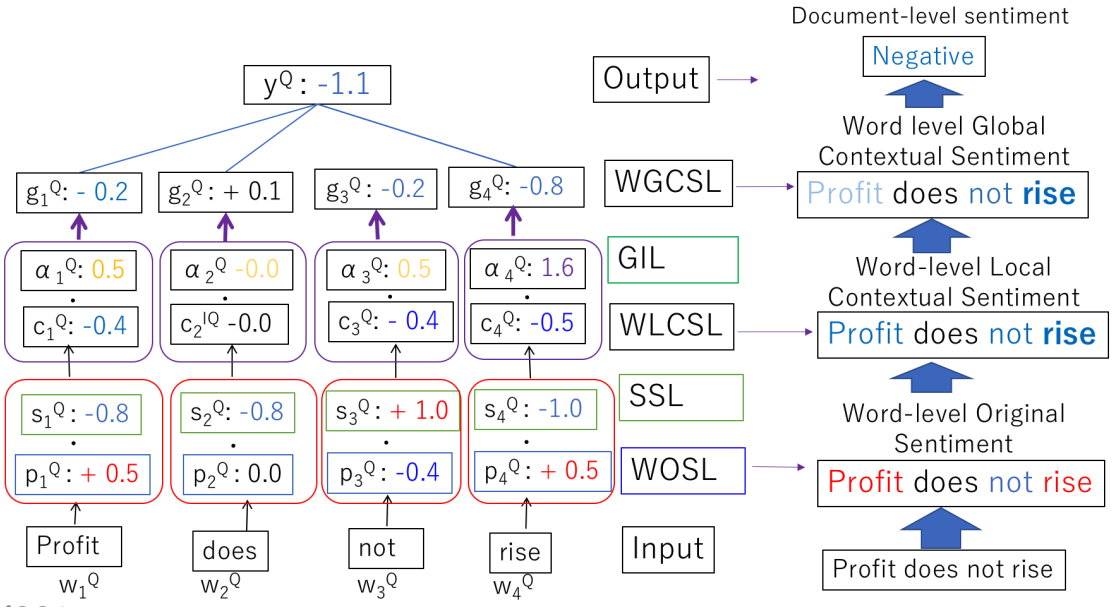

Figure 3: The architecture of the BINN

GIL This layer represents their word-level sentiment shift scores $s_{it}^{\mathbf{Q}}$ using terms and their Surrounding terms as follows:

$$\alpha_{it}^{\mathbf{Q}} := GIL(\boldsymbol{e}_{it}^{\mathbf{Q}}, \{\{\boldsymbol{e}_{it}^{\mathbf{Q}}\}_{t=1}^{T}\}_{i=1}^{L}). \tag{4}$$

where $GIL(\cdot) \in [0, \infty]$ and $\alpha_{it}^{\mathbf{Q}} (> 0)$ represents the scores for global important. Here, if the value of $\alpha_{it}^{\mathbf{Q}}$ is large, then, term $w_{it}^{\mathbf{Q}}$ is important.

WGCSL This layer converts the values in WLCSL and GIL into the word-level global contextual sentiment scores $\{\{g_{it}^{\mathbf{Q}}\}_{t=1}^{T}\}_{i=1}^{L}$:

$$g_{it}^{\mathbf{Q}} := c_{it}^{\mathbf{Q}} \cdot \alpha_{it}^{\mathbf{Q}}. \tag{5}$$

Output Finally, the document-level sentiment score of this review $\mathbf{Q}$ is output as follows:

$$y^{\mathbf{Q}} := \sum_{i=1}^{L} \sum_{t=1}^{T} c_{it}^{\mathbf{Q}} \tag{6}$$

where $y^{\mathbf{Q}} > 0$ means that a review $\mathbf{Q}$ is positive and $y^{\mathbf{Q}} < 0$ means that a review $\mathbf{Q}$ is negative.

As for this BINN , this chapter consider the learning theory for realizing the interpretability,

3.3 Main Assumption and Problem Setting

This section mainly discuss the required conditions and ideas for realizing the interpretability of the layers in BINN.

Main Assumption In developing BINN, the realization of the interpretability in each layer of BINN (i.e., the situation where each layer represents the corresponding score) is crucial. This study plans to learn BINN using the backpropagation method using the sigmoid cross-entropy between $y^{\mathbf{Q}}$ and $d^{\mathbf{Q}}$ $(= L^{\mathbf{Q}}_{doc})$, basically, and a training Ω^{tr}. However, learning using $L^{\mathbf{Q}}_{doc}$ alone cannot realize such interpretability; therefore, a specific learning strategy is required to achieve our aim. To address this issue, we first assume the following Assumption 3.3.1 according to some previous linguistic researches [35, 49].

Assumption 3.3.1. *Let S^* be a set of terms which have strong original sentiment. For each $w^{\mathbf{Q}}_{it} \in \mathbf{Q}$, if $w^{\mathbf{Q}}_{it} \in S^*$, then the following equation is satisfied.*

$$\begin{cases} d^{\mathbf{Q}} = 1 & (R^*(w^{\mathbf{Q}}_t) \cdot PN^*(w^{\mathbf{Q}}_t) \cdot G^*(w^{\mathbf{Q}}_t) = 1) \\ d^{\mathbf{Q}} = 0 & (R^*(w^{\mathbf{Q}}_t) \cdot PN^*(w^{\mathbf{Q}}_t) \cdot G^*(w^{\mathbf{Q}}_t) = -1) \end{cases} \tag{7}$$

where

$$R^*(w^{\mathbf{Q}}_t) := \begin{cases} -1 & (sentiment\ of\ w^{\mathbf{Q}}_t\ is\ shifted) \\ 1 & (otherwise) \end{cases},$$

$$PN^*(w^{\mathbf{Q}}_t) := \begin{cases} 1 & (original\ sentiment\ of\ w^{\mathbf{Q}}_t is\ positive) \\ -1 & (otherwise) \end{cases},\ and$$

$$G^*(w^{\mathbf{Q}}_t) := \begin{cases} 1 & (term\ w^{\mathbf{Q}}_t\ is\ important\ in\ the\ entire\ review\ \mathbf{Q}) \\ 0 & (otherwise) \end{cases},$$

In the above Assumption 3.3.1, $PN^*(w^{\mathbf{Q}}_t)$, $PN^*(w^{\mathbf{Q}}_t) \cdot R^*(w^{\mathbf{Q}}_t)$, and $PN^*(w^{\mathbf{Q}}_t) \cdot R^*(w^{\mathbf{Q}}_t) \cdot G^*(w^{\mathbf{Q}}_t)$ correspond to word-level original sentiment, word-level local contextual sentiment, and word-level global contextual sentiment, respectively.

Interpretabiliry for WGCSL and WLCSL

We first discuss the interpretability in WGCSL and WLCSL. If Assumption 3.3.1 is satisfied, then, the following Proposition 3.3.2 is established.

Proposition 3.3.2. *If $w^{\mathbf{Q}}_t \in S^*$ and $w^{\mathbf{Q}}_t$ appears sufficient times in a training dataset Ω^{tr}, then, the following equation is satisfied after sufficient time of update using the backpropagation method with $L^{\mathbf{Q}}_{doc}$:*

$$\begin{cases} g^{\mathbf{Q}}_{it} > 0 & (d^{\mathbf{Q}} = 1) \\ g^{\mathbf{Q}}_{it} < 0 & (d^{\mathbf{Q}} = 0) \end{cases} \tag{8}$$

where

$$L^{\mathbf{Q}}_{doc} = CE(\mathrm{sigmoid}(y^{\mathbf{Q}}), d^{\mathbf{Q}}) \tag{9}$$

and $CE(a, b)$ is the cross-entropy between a and b.

Therefore, from Eq (8) in Proposition 3.3.2 and Eq (7) in Assumption 3.3.1, the following Corollary3.3.1 is established:

Corollary 3.3.1.

$$\begin{cases} g^{\mathbf{Q}}_{it} > 0 & (PN^*(w^{\mathbf{Q}}_t) \cdot R^*(w^{\mathbf{Q}}_t) \cdot G^*(w^{\mathbf{Q}}_t) = 1) \\ g^{\mathbf{Q}}_{it} < 0 & (PN^*(w^{\mathbf{Q}}_t) \cdot R^*(w^{\mathbf{Q}}_t) \cdot G^*(w^{\mathbf{Q}}_t) = -1) \end{cases} \tag{10}$$

This Corollary3.3.1 explains that the learning using $L_{doc}^{\mathbf{Q}}$ as a loss value is effective for realizing the interpretability in WGCSL. Therefore, we can realize the interpretability of WGCSL with the learning using $L_{doc}^{\mathbf{Q}}$.

Moreover, the following proposition 3.3.3 is established:

PROPOSITION 3.3.3. *If $w_{it}^{\mathbf{Q}} \in S^*$ and $w_{it}^{\mathbf{Q}}$ appears sufficient times in a training dataset Ω^{tr}, then, the following equation is satisfied after sufficient time of update using the backpropagation method with $L_{doc}^{\mathbf{Q}}$:*

$$\begin{cases} c_{it}^{\mathbf{Q}} > 0 & (PN^*(w_{it}^{\mathbf{Q}}) \cdot R^*(w_{it}^{\mathbf{Q}}) = 1) \\ c_{it}^{\mathbf{Q}} < 0 & (PN^*(w_{it}^{\mathbf{Q}}) \cdot R^*(w_{it}^{\mathbf{Q}}) = -1) \end{cases} \tag{11}$$

after the learning.

Problem for the realization of the interpretability in WOSL and SSL

As discussed in the above, it can be established that Learning with $L_{doc}^{\mathbf{Q}}$ can realize the interpretability in WCSL for terms in S^*; however, it cannot realize the interpretability in WOSL and SSL due to the following problem 3.3.4:

PROBLEM 3.3.4. *If the polarity of $c_t^{\mathbf{Q}}$ is accurately negative, the following two cases are possible: (1) $p_t^{\mathbf{Q}} > 0$ and $s_t^{\mathbf{Q}} < 0$, or (2) $p_t^{\mathbf{Q}} < 0$ and $s_t^{\mathbf{Q}} > 0$, and the accurate case cannot be chosen automatically in general learning with $L_{doc}^{\mathbf{Q}}$.*

Main Idea for the realization of interpretability in WOSL and SSL

Lexicon Initialization Let $\Phi(S^*)$ be a subset of S^*. Moreover, let us denote Conditions 3.3.5 and 3.3.5 as follows:

CONDITION 3.3.5. *$\|\boldsymbol{e}_{it}^{\mathbf{Q}} - \boldsymbol{w}_j^{em}\| < \delta$ where δ is sufficiently small, then,*

$$\|s_{it}^{\mathbf{Q}} - s_{it}^{\mathbf{Q}(w_{it}^{\mathbf{Q}}, w_j)}\|_2 < T'\delta$$

where $T' > 0$ and $\mathbf{Q}(w_{it}^{\mathbf{Q}}, w_j)$ represents the review where word $w_{it}^{\mathbf{Q}}$ is replaced by w_j in $\mathbf{Q}$, is established.

CONDITION 3.3.6. *$\|\boldsymbol{e}_{it}^{\mathbf{Q}} - \boldsymbol{w}_j^{em}\| < \delta$ where δ is sufficiently small, then,*

$$\|\alpha_{it}^{\mathbf{Q}} - \alpha_{it}^{\mathbf{Q}(w_{it}^{\mathbf{Q}}, w_j)}\|_2 < T''\delta$$

where $T'' > 0$ is established.

We assume that this problem can be solved by initially limiting the polarity of $p_t^{\mathbf{Q}}$ to the accurate case for each word in $\Phi(S^*)$ if the Condition 3.3.5 is satisfied for $SSL(\cdot)$. This is because this limitation leads to the accurate choice from the above two cases, and can lead to the learning of $s_t^{\mathbf{Q}}$ within the appropriate case in this situation. If Condition 3.3.5 is satisfied, then, the effect of this limitation first works for only words in $\Phi(S^*)$; however, it is assumed that this effect is propagated to each term $w' \in S^* \backslash \Phi(S^*)$ if the meaning of term w' is similar to any of word in $\Phi(S^*)$, thorough learning, afterward, due to Condition 3.3.5. As a result, if $|\Phi(S^*)|$ is sufficiently large, then, the effect is assumed to be propagated to all the terms in S^*.

Effect of Lexical Initialization into the interpretability in GIL. In addition, if the above Lexicon initialization is used, then, after the learning with $L^{\mathbf{Q}}_{doc}$ has finished, $\alpha^{\mathbf{Q}}_{it}$ is expected to become large in a case where $w^{\mathbf{Q}}_{it} \in \Omega(S^d)$ and any of the similar terms to $w^{\mathbf{Q}}_{it}$ has a strong sentiment (Proposition 3.5.4). This manner is known to be natural for humans [64].

SSL regularization Moreover, it is assumed that the following $L^{*\mathbf{Q}}_{shift}$ is expected to improve the interpretability in WOSL and SSL because this $L^{*\mathbf{Q}}_{shift}$ regularize the values of SSL in a form that $s^{\mathbf{Q}}_{it}$ learns to be positive (negative) if the polarities of $s^{\mathbf{Q}}_{it}$ and $p^{\mathbf{Q}}_{t}$ are agree (different) and $w^{\mathbf{Q}}_{it} \in \Phi(S^*)$.

$$L^{*\mathbf{Q}}_{shift} := \sum_{i,t \in \{i,t | w^{\mathbf{Q}}_{it} \in (\Phi(S^*) \cap \mathbf{Q})\}} SCE(s^{\mathbf{Q}}_{it}, l_{ssl}(PN^*(w^{\mathbf{Q}}_{it}))) \tag{12}$$

where $SCE(a, b)$ is the sigmoid cross entropy between a and b and
$l_{ssl}(a) = \begin{cases} 1 & (a > 0 \wedge d^{\mathbf{Q}} = 1) \vee (a < 0 \wedge d^{\mathbf{Q}} = 0) \\ 0 & (a > 0 \wedge d^{\mathbf{Q}} = 0) \vee (a < 0 \wedge d^{\mathbf{Q}} = 1) \end{cases}$.

This regularization is agreeable for Assumption 3.3.1 because if $w^{\mathbf{Q}}_{it} \in S^*$, then, $R^*(w^{\mathbf{Q}}_{it})$ is negative when $l_{ssl}(PN^*(w^{\mathbf{Q}}_{it})) = 0$, and that is positive in the opposite case. Therefore, the joint learning with the above $L^{*\mathbf{Q}}_{shift}$ and $L^{\mathbf{Q}}_{doc}$ should be promising for realizing the interpetability of layers in BINN.

3.4 Learning Strategy

This section describes the proposed Lexical Initialization learning (LEXIL) and JSP learning, which are the learning strategy for developing a BINN.

3.4.1 LEXIL: Lexical Initialization Learning

This section describes the learning strategy of the BINN. Motivated by the discussion in Section 3.3, we propose a learning strategy as shown in Algorithm 1. We call this learning strategy as Lexical Initialization Learning (LEXIL).

Training In LEXIL, BINN is learned using the following $Lo^{\mathbf{Q}}$ as a loss function:

$$L^{\mathbf{Q}}_{doc} := SCE(\sum_{i=1}^{L} \sum_{t=1}^{T} c^{\mathbf{Q}}_{it}, d^{\mathbf{Q}}) \tag{13}$$

where $SCE(a, b)$ means the sigmoid cross-entropy between a and b. Through the learning with this $L^{\mathbf{Q}}_{doc}$, the values in the WLCSL and WGCSL learn to represent the word-level local and global contextual sentiment scores, respectively, for terms in S^* because Propositions 3.3.3 and Corollary 3.3.1 are established.

Lexical Initialization Motivated by the assumption in Section 3.3, LEXIL initializes the values in $\boldsymbol{W}^p$ in the following way using a part of S^*, $\Phi(S^*)$ (process 2 in Algorithm 1):

$$w^p_i \leftarrow \begin{cases} PN^*(w_i) & (w_i \in \Phi(S^*)) \\ 0 & (\text{otherwise}) \end{cases} \tag{14}$$

Algorithm 1 LEXIL: Lexical Initialization Learning

1: **for** $i \leftarrow 1$ to v **do**

2: $w_i^p \leftarrow \begin{cases} PN^*(w_i) & (w_i \in \Phi(S^*)) \\ 0 & (\text{otherwise}) \end{cases}$;

3: Learn BINN using the gradient values by $L_{doc}^{\mathbf{Q}}$;

Let $\Omega(\Phi(S^*))$ be a set of word w_j that satisfies the $\min_{w_i \in \Phi(S^*)} \|\boldsymbol{w}_i^{em} - \boldsymbol{w}_j^{em}\|_2 < \delta$ where δ is sufficiently small. The lexical initialization is expected to improve the interpretability in SSL, WOSL, and GIL as follows.

A) SSL By the effect of lexical initializaion, SSL is expected to learn the sentiment shift for words in $\Phi(S^*)$ and $\Omega(\Phi(S^*))$ through LEXIL. (Propositions 3.5.5 and 3.5.6).

B) WOSL As a result, WOCL learns word-level original sentiment for words in $\Omega(\Phi(S^*))$ through LEXIL, because the appropriate cases were decided for them (Proposition 3.5.7).

C) GIL GIL learns to represent global word-level context through LEXIL because $\alpha_{it}^{\mathbf{Q}}$ is expected to become large in a case where $w_{it}^{\mathbf{Q}} \in \Omega(\Phi(S^*))$ and any of the similar terms to $w_{it}^{\mathbf{Q}}$ has a strong sentiment (Proposition 3.5.4). This manner is known to be natural for humans [64].

Through LEXIL, WOSL, SSL, and GIL learns to represent their corresponding scores. After the learning has finished, BINN can analyze document-level sentiment through extracting the word-level original sentiment, sentiment shift, and word-level local contextual sentiment, and word-level global contextual sentiment from WOSL, SSL, WLCSL, and WGCSL for words in S^*, respectively in a following situation (= ideal situation):

- Conditions 3.3.5 and Conditions 3.3.6 are satisfied for SSL and GIL,
- the size of $\Phi(S^*)$ is large enough to satisfy $S^* \in \Omega(\Phi(S^*))$, and
- Assumption 3.3.1 is satisfied (for all the terms in S^* and $\Phi(S^*)$).

The above three requirements are important for the success of the realization fo the interpretability (See Section 3.5 for the details).

3.4.2 Joint Sentiment Propagation (JSP) Learning In addition, we propose Joint Sentiment Propagation (JSP) Learning (as descrined in Algorithm 2) as the improved LEXIL. Motivated by the assumption for the SSL regularization in Section 3.3, JSP learning uses the following $L_{joint}^{\mathbf{Q}}$ as a loss:

$$L_{joint}^{*\mathbf{Q}} := L_{doc}^{\mathbf{Q}} + \lambda \cdot L_{shift}^{*\mathbf{Q}}$$

where $L_{doc}^{\mathbf{Q}}$ is the sigmoid cross entropy between $d^{\mathbf{Q}}$ and $y_t^{\mathbf{Q}}$ and λ is the hyper-parameter value.

$L_{doc}^{\mathbf{Q}}$ corresponds to the loss for document-level sentiment and $L_{shift}^{*\mathbf{Q}}$ corresponds to the loss for regularizing the SSL. Usage of $L_{shift}^{*\mathbf{Q}}$ is specific in the above and we call this usage of $L_{shift}^{\mathbf{Q}}$ as *SSL regularization* in this thesis.

It can be possible that LEXIL can not realize the interpretability in the SSL. This JSP learning has the potential for working in that situation because JSP utilizes joint learning for the document-level sentiment analysis and SSL regularization. The SSL regularization is expected to support the realization of the interpretability in SSL.

Algorithm 2 Joint Sentiment Propagation Learning

1: **for** $i \leftarrow 1$ to v **do**

2: $w_i^p \leftarrow \begin{cases} PN(w_i) & (w_i \in \Phi(S^*)) \\ 0 & (\text{otherwise}) \end{cases}$;

3: Learn BINN using the gradient values by $L_{joint}^{*\mathbf{Q}}$.;

JSP learning is also expected to realize the interpretability of BINN in a situation where LEXIL works (can be theoretically analyzed using the same manner as in LEXIL).

3.5 Theoretical Analysis

3.5.1 Overview This section briefly describes theoretical analysis result in LEXIL. We briefly describes theoretical analysis result in LEXIL. Before the explanation, we define several symbols. See Section 3.5.2 for details and proofs. Let us define $R(\cdot)$, $PN(\cdot)$, and Condition 3.5.1 as follows.

$$R(w_{it}^{\mathbf{Q}}) := \begin{cases} -1 & (sentiment\ of\ w_{it}^{\mathbf{Q}}\ is\ shifted) \\ 1 & (otherwise) \end{cases}.$$

$$PN(w_{it}^{\mathbf{Q}}) := \begin{cases} 1 & (\text{sign}(d^{\mathbf{Q}} - 0.5) \neq R(w_{it}^{\mathbf{Q}})) \\ -1 & (\text{sign}(d^{\mathbf{Q}} - 0.5) = R(w_{it}^{\mathbf{Q}})) \end{cases}.$$

Condition 3.5.1. $w_i^p \begin{cases} > 0 & (OS(w_i^p) > 0) \\ < 0 & (OS(w_i^p) < 0) \end{cases}$ *is established where* $OS(w_j^p) := E[PN(w_{it}^{\mathbf{Q}}) | w_{it}^{\mathbf{Q}} = w_j^p, \mathbf{Q} \in \Omega^{tr}]$

and Ω^{tr} *is a set of reviews in a training dataset.*

Here, $PN(w_{it}^{\mathbf{Q}}) = 1$ denotes the case where the sentiment of $w_{it}^{\mathbf{Q}}$ is shifted in a negative review or the sentiment of $w_{it}^{\mathbf{Q}}$ is not shifted in a positive review, and $PN(w_{it}^{\mathbf{Q}}) = -1$ denotes the opposite case. In LEXIL, following three propositions are satisfied.

Proposition 3.5.2. $\begin{cases} \frac{\partial L_{doc}^{\mathbf{Q}}}{\partial c_{it}^{\mathbf{Q}}} < 0 & (d^{\mathbf{Q}} = 1) \\ \frac{\partial L doc^{\mathbf{Q}}}{\partial c_{it}^{\mathbf{Q}}} > 0 & (d^{\mathbf{Q}} = 0) \end{cases}$ *is satisfied.*

Proposition 3.5.3. *If the Condition 3.5.1 , is satisfied for every word* $w_i \in S^d$*, then, for every* $w_{it} \in \Omega(S^d)$*,*

$$\begin{cases} E[w_{I(w_{it})}^p] > 0 & (OS(w_{I(w_{it})}^p > 0) \\ E[w_{I(w_{it})}^p] < 0 & (OS(w_{I(w_{it})}^p) < 0) \end{cases} \quad and \tag{15}$$

$$\begin{cases} E[s_{it}^{\mathbf{Q}}] > 0 & (R(w_{it}^{\mathbf{Q}}) > 0) \\ E[s_{it}^{\mathbf{Q}}] < 0 & (R(w_{it}^{\mathbf{Q}}) < 0) \end{cases} \tag{16}$$

are satisfied after sufficient iterations through LEXIL.

Proposition 3.5.4. *After the sufficient iterations,*

$$E[\alpha_{it}^{\mathbf{Q}} | w_{it}^{\mathbf{Q}} \in \Omega^*(S^d)] > E[\alpha_{it}^{\mathbf{Q}} | w_{it}^{\mathbf{Q}} \notin \Omega^*(S^d)]. \tag{17}$$

Proposition 3.5.4 is established because

$$\frac{\partial L^{\mathbf{Q}}}{\partial \alpha_{it}^{\mathbf{Q}}} = \Delta_o^{\mathbf{Q}} \cdot s_{it}^{\mathbf{Q}} \cdot p_{it}^{\mathbf{Q}} \tag{18}$$

and where

$$\alpha_{it}^{\mathbf{Q}} > 0,$$

$$\Delta_o^{\mathbf{Q}} := \begin{cases} \text{sigmoid}(\sum_{i=1}^{L}\sum_{t=1}^{T} c_{it}^{\mathbf{Q}}) & (d^{\mathbf{Q}} = 0) \\ \text{sigmoid}(\sum_{i=1}^{L}\sum_{t=1}^{T} c_{it}^{\mathbf{Q}}) - 1 & (d^{\mathbf{Q}} = 1) \end{cases},$$

$$\alpha_{it}^{\mathbf{Q}} \simeq \alpha_{it}^{\mathbf{Q}(\mathbf{w}_{\mathbf{it}}^{\mathbf{Q}}, \mathbf{w_j})} \tag{19}$$

if $\|\boldsymbol{e}_{it}^{\mathbf{Q}} - \boldsymbol{w}_j^{em}\|$ is sufficiently small, and by the Lexical Initialization,

$$E[p_{it}^{\mathbf{Q}} | w_{it}^{\mathbf{Q}} \in \Omega(S^d)] >> E[p_{it}^{\mathbf{Q}} | w_{it}^{\mathbf{Q}} \notin \Omega(S^d)].$$

is established in the early iterations.

They indicate that WCSL, WOSL, LWCL, and GWCL learn to represent the corresponding scores in an ideal case. Moreover, this analysis suggests that the quality of the word sentiment dictionary is important for the success of propagation, where $|S^d|$ should not be too small and each word in S^d must satisfy Condition 3.5.1. Proposition 3.5.3 can be explained from the following propositions.

Proposition 3.5.5. *If Condition 3.5.1 is satisfied for word $w_{it}^{\mathbf{Q}}$, then, Eq (16) is satisfied for $w_{it}^{\mathbf{Q}}$.*

This can be proved by analyzing that if $d^{\mathbf{Q}} = 1$ and $w_{it}^{\mathbf{Q}} > 0$, or $d^{\mathbf{Q}} = 0$ and $w_{it}^{\mathbf{Q}} < 0$, then, $\frac{\partial Lo^{\mathbf{Q}}}{\partial s_{it}^{\mathbf{Q}}} < 0$, If $d^{\mathbf{Q}} = 1$ and $w_{p,i} < 0$, or $d^{\mathbf{Q}} = 0$ and $w_{p,i} > 0$, then, and in the opposite case, $\frac{\partial Lo^{\mathbf{Q}}}{\partial s_{it}^{\mathbf{Q}}} > 0$.

As a result, every word $w_{it}^{\mathbf{Q}} \in S^d$ is expected to satisfies Eq (16) because it satisfies Condition 3.5.1.

Proposition 3.5.6. *If w_i satisfies Condition 3.5.1 and Eq (16), then Eq (16) is satisfied for w_j where $\|\boldsymbol{e}_{it}^{\mathbf{Q}} - \boldsymbol{w}_j^{em}\|_2 < \delta$ where $\delta > 0$ is sufficiently small.*

This can be explained considering that let $(w_{it}^{\mathbf{Q}}, w_j)$ be a review in which a word ($w_{it}^{\mathbf{Q}}$ is replaces to word w_j, then, $\|s_{it}^{\mathbf{Q}} - s_{it}^{\mathbf{Q}(w_{it}^{\mathbf{Q}}, w_j)}\|_2 < T'\delta$ where $T' > 0$ is established, and that similar words often appears in a similar pattern.

As a result, every word in $\Omega(S^d)$ satisfies Eq (16) because similar words often appears in a similar pattern.

Proposition 3.5.7. *If Eq (16) is satisfied for w_i, then, Eq (15) is satisfied for w_i and becomes to satisfy Condition 3.5.1.*

This can be explained by analyzing that if $s_{xt}^{\mathbf{Q}} < 0 (R(w_{xt}^{\mathbf{Q}}) = -1)$ and $s_{xt}^{\mathbf{Q}} > 0 (R(w_{xt}^{\mathbf{Q}}) = 1)$, then,

$$\begin{cases} \frac{\partial Lo^{\mathbf{Q}}}{\partial w^p_{I(w_{xt}^{\mathbf{Q}})}} < 0 (PN(w_{xt}^{\mathbf{Q}}) = 1) \\ \frac{\partial Lo^{\mathbf{Q}}}{\partial w^p_{I(w_{xt}^{\mathbf{Q}})}} > 0 (PN(w_{xt}^{\mathbf{Q}}) = -1) \end{cases} \text{ is established.}$$

3.5.2 Proofs We briefly introduce the proofs or explanations of the propositions.

Proof of Proposition 3.5.2 *Proof*

$$\frac{\partial Lo^{\mathbf{Q}}}{\partial c_{it}^{\mathbf{Q}}} = \Delta_o^{\mathbf{Q}} = \begin{cases} > 0 & (d^{\mathbf{Q}} = 0) \\ < 0 & (d^{\mathbf{Q}} = 1) \end{cases} \tag{20}$$

because $0 < sigmoid(\cdot) < 1$. Therefore, the proposition is established.

Proof of Proposition 3.5.5 *Proof*

$$\frac{\partial Lo^{\mathbf{Q}}}{\partial s_{it}^{\mathbf{Q}}} = \frac{\partial Lo^{\mathbf{Q}}}{\partial c_{it}^{\mathbf{Q}}} \frac{\partial c_{it,i}^{\mathbf{Q}}}{\partial s_{it}^{\mathbf{Q}}} = \frac{\partial Lo^{\mathbf{Q}}}{\partial c_{it}^{\mathbf{Q}}} p_{it}^{\mathbf{Q}} \alpha_{it}^{\mathbf{Q}} \tag{21}$$

Here, $alpha_{it}^{\mathbf{Q}} > 0$, Condition 3.5.1 is established for $w_{it}^{\mathbf{Q}}$, and word $w_{it}^{\mathbf{I}} = w_i$,and Proposition 3.5.2 is established. Therefore, this proposition is established.

Explanation of Proposition 3.5.6 First, if $\|\boldsymbol{e}_{it}^{\mathbf{Q}} - \boldsymbol{w}_j^{em}\| < \delta$ where δ is sufficiently small, then,

$$\|s_{it}^{\mathbf{Q}} - s_{it}^{\mathbf{Q}(w_{it}^{\mathbf{Q}}, w_j)}\|_2 < T'\delta$$

where $T' > 0$ is established. Therefore, this proposition is satisfied if the following assumption: "similar words often appears in a similar pattern." is established.

Proof of Proposition 3.5.7 First, the following equation is established.

$$\begin{aligned} \frac{\partial Lo^{\mathbf{Q}}}{\partial w_j^p} &= \sum_{i=1}^{L} \sum_{t=1}^{T} \frac{\partial Lo^{\mathbf{Q}}}{\partial p_{it}^{\mathbf{Q}}} \delta(w_{it}^{\mathbf{Q}}, w_j) \\ &= \sum_{i=1}^{L} \sum_{t=1}^{T} \frac{\partial Lo^{\mathbf{Q}}}{\partial c_{it}^{\mathbf{Q}}} \frac{\partial c_{it}^{\mathbf{Q}}}{\partial w_{p,j}} \delta(w_{it}^{\mathbf{Q}}, w_i) \\ &= \sum_{i=1}^{L} \sum_{t=1}^{T} \frac{\partial Lo^{\mathbf{Q}}}{\partial c_{it}^{\mathbf{Q}}} \alpha_{it}^{\mathbf{Q}} s_{it}^{\mathbf{Q}} \delta(w_{it}^{\mathbf{Q}}, w_i). \\ &= \sum_{i=1}^{L} \sum_{t=1}^{T} \Delta_o^{\mathbf{Q}} \alpha_{it}^{\mathbf{Q}} s_{it}^{\mathbf{Q}} \delta(w_{it}^{\mathbf{Q}}, w_i) \end{aligned}$$

where

$$\delta(a, b) = \begin{cases} 1 & (a = b) \\ 0 & (a \neq b) \end{cases}.$$

Here, $\alpha_{it}^{\mathbf{Q}} > 0$ is established, $s_{it}^{\mathbf{Q}}$ satisfies the Eq (16), and $\Delta_o^{\mathbf{Q}}$ satisfies Eq(20); therefore, Eq (15) is also satisfied for word $w_{it}^{\mathbf{Q}}$. In this situation, $w_{it}^{\mathbf{Q}}$ satisfy the both Eq (16) and Eq (15); therefore, this word satisfies Condition 3.5.1. Therefore, this proposition is established.

3.6 Conclusion

This chapter derives the two types of basic learning strategy called LEIXIL and JSP learning that can realize the interpretablity of BINN in an ideal situation. To achieve the interpretability of layers in $BINN$, LEIXIL utilizes *Lexical Initialization*, and JSP learning utilizes *Lexical Initialization* and SSL *regularizaion*. These techniques are theoretically effective for improving the interpretability in SSL and WOSL. In addition, we theoretically analyze that proposed LEXIL and JSP learning can realize the interpretability of BINN in a case where some requirements are satisfied (= in an ideal case.) First, we can choice the LEXIL for the learning strategy, and then we can utilize the JSP learning in situations where the LEXIL fails. This theoretical derivation and analysis of the proposed basic learning strategy is the first work for systemically analyzing the realization of the interpretability in NNs, as far as we know.

Unfortunately, the LEXIL and JSP learning are thoroughly basic learning strategy and can not be used originally in practical cases because in practical, we can not use $\Phi(S^*)$ and $PN^*(\cdot)$; however, they can be utilized by revising them a little. We describe the practical method for the usage of LEXIL and JSP learning in the following chapters in this part.

Chapter 4

Sentiment Interpretable Neural Network (SINN)

This chapter introduces the sentiment interpretable neural network (SINN) [24] as a specific example of BINN and specific example of the application of our basic theory. It should be noted that original LEXIL or JSP learning can not be used in a real situation because S^* (defined in Assumption 3.3.1) is not available. Therefore, in developing a SINN, we utilize LEXIL or JSP learning in a form that we utilize S^d and $PS(\cdot)$ instead of $\Phi(S^*)$ and $PN(\cdot)$ where S^d is a set of words in a word sentiment dictionary and $PS(w)$ is the score of word w provided by the word sentiment dictionary. We describe this type of converted LEXIL and JSP learning as Practical LEXIL (PLEXIL) and Practical JSP learning (PJSP learning). In such a way, we can develop a SINN in a practical way. The success of the SINN development using PLEXIL and PJSP learning means that the proposed LEXIL and JSP learning (proposed in Chapter 3) can be utilized in actually developing interpretable NNs.

We first introduce the SINN in Section 4.1 and then explain the detailed SINN structure and the learning strategy of SINN in Section 4.2. In this learning strategy, we use LEXIL or JSP learning in a practical manner. We then experimentally evaluate our approach using real textual datasets In Sections 4.4 and 4.5, and then conclude this chapter in Section 4.6.

4.1 Overview

As a specific example of BINN, this chapter considers the Sentiment Interpretable Neural Network (SINN) [24]. This SINN includes the Word-level Original Sentiment Layer (WOSL), Local word-level context Layer (LWCL), Global word-level context Layer (GWCL) and Word-level Contextual Sentiment Layer (WCSL), as shown in Figure 5. Each layer in this neural network represents the corresponding word-level scores.

Therefore, this type of neural network can explain the prediction results for the word-level contextual sentiment analysis (WCSA) as shown in using the following three types of scores are required in the explanation as shown in Figure 4:

1) Word-level original sentiment represents the sentiment of each word where it originally has (e.g., scores in a word sentiment dictionary [17]).

2) Local word-level context represents whether each term in a review is shifted or not by the contexts of multiple words or phrases (e.g., "up" in "did not go up." and "bullish" in "manipulate

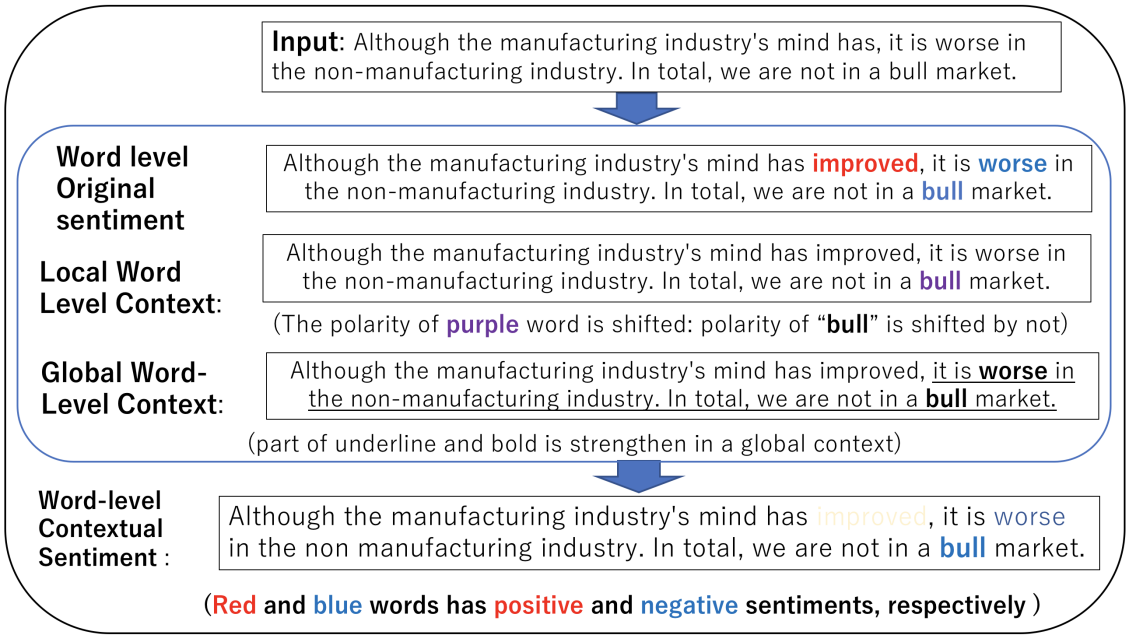

Figure 4: Explanation Image from SINN

bullish opinion on the stock market").

3) Global word-level context represents the important part of an entire review.

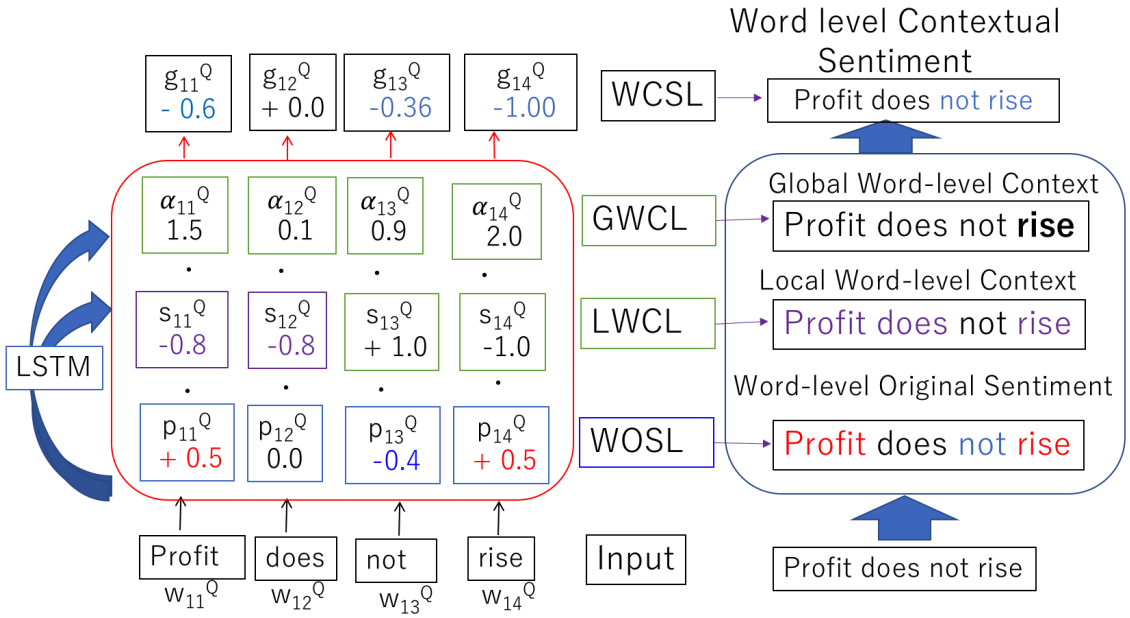

Figure 5: The architecture of the SINN

Thus, this SINN is valuable in a case where the WCSA with the explanation using the above word-level original sentiment, local word-level context, and global word-level context are required.

Here, WCSA is the task for assigning word-level sentiment score to each term in a review by considering the contextual influence to it from the other terms. For example, "good" originally has a positive meaning. However, in the phrase "not good", this word is shifted by "not" and its sentiment becomes negative. This WCSA is known to be valuable for mining reviews or opinions [62] because pinpointing positive or negative expressions as shown in the sentences below is often required in the industry.

(1) In total, we are in a $bull^+$ market.

(2) This room is not $clean^-$.

(3) Products in this shop are too $expensive^-$.

By pinpointing positive or negative expressions, we can identify the detailed positive or negative attitude of consumers. For example, from the third review listed above, we see that the problem for this shop is caused by price, therefore, the price should be improved.

4.2 Architecture of SINN

This section explains the structure of SINN [24] that includes the WOSL, SSL, WLCSL, GIL and WGCSL, as shown in Figure 5 more concretely.

Notation We first define several symbols. Let $\{(\mathbf{Q}_n, d^{\mathbf{Q}_n})\}_{n=1}^N$ be a training dataset where N is the training data size, $\mathbf{Q}_n$ is a review, and $d^{\mathbf{Q}_n}$ is its sentiment tag (1 is positive and 0 is negative). Let $\{w_i\}_{i=1}^v$ be the terms that appear in a text corpus, v be the vocabulary size, and $I(w_i)$ be the vocabulary index of word w_i where $I(w_i) = i$. Let $\boldsymbol{w}_i^{em} \in \mathbb{R}^e$ be an embedding representation of word w_i where $\|\boldsymbol{w}_i^{em}\|_2 = 1$, and the embedding matrix $\boldsymbol{W}^{em} \in \mathbb{R}^{v \times e}$ be $[\boldsymbol{w}_1^{emT}, \cdots, \boldsymbol{w}_v^{emT}]^T$ where e is the dimension size of the word embeddings. $\boldsymbol{W}^{em}$ is constant and obtained using the skip-gram method [39] and the text corpus in a training dataset.

WOSL Given a review $\mathbf{Q} = \{\{w_{it}^{\mathbf{Q}}\}_{t=1}^T\}_{i=1}^L$, this layer converts the words $\{\{w_{it}^{\mathbf{Q}}\}_{t=1}^T\}_{i=1}^L$ to word-level original sentiment representations $\{\{p_{it}^{\mathbf{Q}}\}_{t=1}^n\}_{i=1}^L$ as

$$p_{it}^{\mathbf{Q}} = w^p_{I(w_{it}^{\mathbf{Q}})} \tag{22}$$

where $\boldsymbol{W}^p \in \mathbb{R}^v$ represents the original sentiment scores of words, and w_i^p is the ith element of $\boldsymbol{W}^p$. The w_i^p value corresponds to the original sentiment score of the word w_i.

LWCL converts words $\{\{w_{it}^{\mathbf{Q}}\}_{t=1}^T\}_{i=1}^L$ to their embeddings $\{\{\boldsymbol{e}_{it}^{\mathbf{Q}}\}_{t=1}^L\}_{i=1}^L$ using $\boldsymbol{W}^{em}$, and converts them to context representations $\{\{\overrightarrow{\boldsymbol{h}}_{it}^{\mathbf{Q}}\}_{t=1}^T\}_{i=1}^L$ and $\{\{\overleftarrow{\boldsymbol{h}}_{it}^{\mathbf{Q}}\}_{t=1}^T\}_{i=1}^L$ using forward and backward LSTMs, $\overrightarrow{\text{LSTM}}$ and $\overleftarrow{\text{LSTM}}$:

$$\overrightarrow{\boldsymbol{h}}_{it}^{\mathbf{Q}} = \overrightarrow{\text{LSTM}}(\boldsymbol{e}_{it}^{\mathbf{Q}}), \overleftarrow{\boldsymbol{h}}_{it}^{\mathbf{Q}} = \overleftarrow{\text{LSTM}}(\boldsymbol{e}_{it}^{\mathbf{Q}}).$$

Then, it converts them to right and left oriented sentiment shift representations, $\overrightarrow{s}_{it}^{\mathbf{Q}}$ and $\overleftarrow{s}_{it}^{\mathbf{Q}}$:

$$\overleftarrow{s}^{\mathbf{Q}}_{it} := \tanh(\boldsymbol{v}^{left^T}\overleftarrow{\boldsymbol{h}}^{\mathbf{Q}}_{it}),\ \overrightarrow{s}^{\mathbf{Q}}_{it} := \tanh(\boldsymbol{v}^{right^T}\overrightarrow{\boldsymbol{h}}^{\mathbf{Q}}_{it}).$$

Here, $\boldsymbol{v}^{right}$ and $\boldsymbol{v}^{left} \in \mathbb{R}^e$ are parameter values. $\overrightarrow{s}^{\mathbf{Q}}_{it}$ and $\overleftarrow{s}^{\mathbf{Q}}_{it}$ denote whether the sentiment of $w^{\mathbf{Q}}_{it}$ is shifted or not by the left-side and right-side terms of $w^{\mathbf{Q}}_{it}$: $\{w^{\mathbf{Q}}_{it'}\}^{t-1}_{t'=1}$ and $\{w^{\mathbf{Q}}_{it'}\}^{T}_{t'=t+1}$, respectively. Finally, $\overrightarrow{s}^{\mathbf{Q}}_{it}$ and $\overleftarrow{s}^{\mathbf{Q}}_{it}$ are converted into word-level sentiment shift scores $s^{\mathbf{Q}}_{it}$:

$$s^{\mathbf{Q}}_{it} := \overrightarrow{s}^{\mathbf{Q}}_{it} \cdot \overleftarrow{s}^{\mathbf{Q}}_{it}. \tag{23}$$

where $s^{\mathbf{Q}}_{it}$ denotes whether the sentiment of $w^{\mathbf{Q}}_{it}$ is shifted ($s^{\mathbf{Q}}_{it} < 0$) or not ($s^{\mathbf{Q}}_{it} \geq 0$).

GWCL This layer converts terms $\{\{w^{\mathbf{Q}}_{it}\}^T_{t=1}\}^L_{i=1}$ into the global word-level context scores $\{\{\alpha_{it}\}^T_{t=1}\}^L_{i=1}$. First, using a revised self-attention mechanism [57], the word-level attention scores are represented as

$$\beta^{\mathbf{Q}}_{it} := \sum_{t'=1}^{T} \frac{e^{\tanh(\overrightarrow{\boldsymbol{h}}^{\mathbf{Q}^T}_{it}\overrightarrow{\boldsymbol{h}}^{\mathbf{Q}}_{it'} + \overleftarrow{\boldsymbol{h}}^{\mathbf{Q}^T}_{it}\overleftarrow{\boldsymbol{h}}^{\mathbf{Q}}_{it'})}}{\sum_{t=1}^{T} e^{\tanh(\overrightarrow{\boldsymbol{h}}^{\mathbf{Q}^T}_{it}\overrightarrow{\boldsymbol{h}}^{\mathbf{Q}}_{it'} + \overleftarrow{\boldsymbol{h}}^{\mathbf{Q}^T}_{it}\overleftarrow{\boldsymbol{h}}^{\mathbf{Q}}_{it'})}}. \tag{24}$$

Using the sentence-level attention mechanism [66], the sentence-level attention scores are represented as

$$\beta^{\mathbf{Q}}_{i} = \frac{e^{AttRNN(\{\boldsymbol{e}^{\mathbf{Q}}_{it}\}^T_{t=1})^T \boldsymbol{v}^s}}{\sum_{i=1}^{L} e^{AttRNN(\{\boldsymbol{e}^{\mathbf{Q}}_{it}\}^T_{t=1})^T \boldsymbol{v}^s}} \tag{25}$$

where $AttRNN(\cdot)$ is a sentence level context vector produced by the word-level Attention RNN [66]. Using these two attention scores, the global word-level context scores are represented by following

$$\alpha^{\mathbf{Q}}_{it} := \beta^{\mathbf{Q}}_{it} \cdot \beta^{\mathbf{Q}}_{t} \tag{26}$$

WCSL represents the word-level contextual sentiment scores $\{\{g^{\mathbf{Q}}_{it}\}^T_{t=1}\}^L_{i=1}$ using the WOSL, LWCL and GWCL:

$$g^{\mathbf{Q}}_{it} := p^{\mathbf{Q}}_{it} \cdot s^{\mathbf{Q}}_{it} \cdot \alpha^{\mathbf{Q}}_{it}. \tag{27}$$

Document-level sentiment The document-level sentiment is predicted by SINN as follows:

$$y^{\mathbf{Q}} := \sum_{i=1}^{L}\sum_{t=1}^{T} g_{it}. \tag{28}$$

4.3 Learning Strategy

This section describes the practical versions of LEXIL and JSP learning, which are available for the SINN.

4.3.1 Practical Lexical Initialization Learning (PLEXIL) We first describes the PLEXIL that is the practical version of LEXIL. Overall process is described in Algorithm 3.

Problem in the application of LEXIL SINN is included in BINNs. Therefore, we can develop using SINN, basically. However, we can not use original LEXIL because S^* is not available in a real case. In addition, considering the practicality, We should not use any contextual word or phrase-level tags, or specific knowledge for word-level contexts.

Considering the above descriptions, we utilize S^d and $PS(\cdot)$ instead of $\Phi(S^*)$ and $PN(\cdot)$ in the assumption that words in S^d should satisfy Assumption 3.3.1, that is, words in S^d should have string sentiment and the sentiment score of word sentiment dictionary should be accurate.

PLEXIL Motivated by the above idea, we develop a SINN utilizing LEXIL as described in Algorithm 3 where $PS(w_i)$ is the sentiment score for word w_i given by the word sentiment dictionary, and S^d is a set of words included in the dictionary. We call this revised LEXIL as PLEXIL. It should be noted that the size of S^d should not be large. As shown in the experimental evaluation (Section 4.4), SINN can be developed with at most fifty terms.

Algorithm 3 PLEXIL: Lexical Initialization Learning

1: **for** $i \leftarrow 1$ to v **do**

2: Initialize w_i^p as $w_i^p \leftarrow \begin{cases} PS(w_i) & (w_i \in S^d) \\ 0 & (\text{otherwise}) \end{cases}$;

3: Learn parameters using the gradient values by $L_{doc}^{\mathbf{Q}}$;

Let $\Omega(S^d)$ be a set of word w_j that satisfies the $\min_{w_i \in S^d} \|\boldsymbol{w}_i^{em} - \boldsymbol{w}_j^{em}\|_2 < \delta$ where δ is sufficiently small. If $|S^d|$ is sufficiently large and $S^d \in S^*$, then, The lexical initialization is expected to improve the interpretability in LWCL, WOSL, and GWCL as follows.

A) LWCL By the effect of lexical initializaion, LWCL is expected to learn the sentiment shift for words in S^d and $\Omega(S^d)$ through PLEXIL. (Propositions 3.5.5 and 3.5.6).

B) WOSL As a result, WOCL learns word-level original sentiment for words in $\Omega(S^d)$ through PLEXIL, because the appropriate cases were decided for them (Proposition 3.5.7).

C) GWCL GWCL learns to represent global word-level context through PLEXIL because $\alpha_{it}^{\mathbf{Q}}$ is expected to become large in a case where $w_{it}^{\mathbf{Q}} \in \Omega(S^d)$ and any of the similar terms to $w_{it}^{\mathbf{Q}}$ has a strong sentiment (Proposition 3.5.4). This manner is known to be natural for humans [64].

Through PLEXIL, the number of words where WOSL, WLCSL, and GWCL can represent their corresponding sentiments ($= |\Omega(S^d)|$) becomes large gradually. After the learning has finished, we can extract word-level contextual sentiment scores from *WCSL* through extracting the word-level original sentiment, local word-level context, and global word-level context scores from WOSL, LWCL, and GWCL.

4.3.2 PJSP learning This section describes the proposed PJSP learning, which is the practical verion of the JSP learning. The overall process is described in Algorithm 4. PJSP learning includes the Lexicon Initialization (process 2 in Algorithm 3) and Joint Learning with SSL Regularization (process 3 in Algorithm 3), which accelerate the interpretability in LWCL in the SINN.

Lexicon Initialization Motivated by the assumption in Section 3.3, we initialize the values in $\boldsymbol{W}^p$ using the prepared small word sentiment dictionary as follows (process 2 in Algorithm 3):

$$w_i^p \leftarrow \begin{cases} PS(w_i) & (w_i \in S^d) \\ 0 & (\text{otherwise}) \end{cases} \tag{29}$$

where $PS(w_i)$ is the sentiment score for word w_i given by the word sentiment dictionary, and S^d is a set of words included in the dictionary. In the above, $PS(\cdot)$ is used instead of $PN(\cdot)$ in the assumption that words in a sentiment dictionary has a strong sentiment and accurate because $PN(\cdot)$ is not availlable in real situations.

Joint Learning with SSL regularization For words in $\Omega(S^d)$, Lexicon initialization and learning with $L_{doc}^{\mathbf{Q}}$ alone can realize the interpretability in the WOSL, LWCL, and CWSL in an ideal case. However, in real situations, they alone often fail to realize such interpretability. To solve this problem, we use the joint learning for the document-level sentiment and SSL regularization in the development. Concretely, A SINN is learned using the following $L_{joint}^{\mathbf{Q}}$ as a loss:

$$\begin{aligned} L_{shift}^{\mathbf{Q}} &:= \sum_{t \in \{t | w_t^{\mathbf{Q}} \in (S^d \cap \mathbf{Q})\}} SCE(s_t^{\mathbf{Q}}, l_{ssl}(PS(w_t^{\mathbf{Q}})) \\ L_{joint}^{\mathbf{Q}} &:= L_{doc}^{\mathbf{Q}} + \lambda \cdot L_{shift}^{\mathbf{Q}} \end{aligned}$$

where $L_{doc}^{\mathbf{Q}}$ is the sigmoid cross entropy between $d^{\mathbf{Q}}$ and $y_t^{\mathbf{Q}}$ and λ is the hyper-parameter value. $L_{doc}^{\mathbf{Q}}$ corresponds to the loss for document-level sentiment and $L_{shift}^{\mathbf{Q}}$ corresponds to the loss for regularizing the LWCL. Usage of $L_{shift}^{\mathbf{Q}}$ is specific in the above and we call this usage of $L_{shift}^{\mathbf{Q}}$ as *SSL regularization* in this thesis.

We use $L_{shift}^{\mathbf{Q}}$ motivated by the assumption in Section 3.3. $L_{shift}^{\mathbf{Q}}$ uses $PS(\cdot)$ and S^d, whereas $L_{shift}^{*\mathbf{Q}}$ in Eq (12) uses $PN^*(\cdot)$ and $\Phi(S^*)$. We use $PS(\cdot)$ and S^d because $PN^*(\cdot)$ and S^* are not available in the real cases. They are used under the assumption that the sentiment scores in a sentiment dictionary should be accurate and $S^d \in S^*$ should be satisfied.

In this learning, $y^{\mathbf{Q}}$ learns to be document-level sentiment using $L_{doc}^{\mathbf{Q}}$, whereas it leads to the interpretability in WCSL. Moreover, learning with $L_{shift}^{\mathbf{Q}}$ is expected to support the interpretability in LWCL, and it leads to the improvement of the interpretability in WOSL.

Algorithm 4 PJSP Learning

1: **for** $i \leftarrow 1$ to v **do**
2: $\quad w_i^p \leftarrow \begin{cases} PS(w_i) & (w_i \in S^d) \\ 0 & (\text{otherwise}) \end{cases}$;
3: Learn parameters using the gradient values by L^{joint}.;

Through PJSP learning, each layer in the SINN learns to represent the corresponding scores, gradually. This propagation succeeds in an ideal case where (1) the size of S^d is large enough to satisfy $S^* \in \Omega(\Phi(S^d))$, and (2) $S^d \in S^*$ is satisfied.

4.4 Experimental Intepretability Evaluation

This section experimentally evaluates the proposed method in terms of the interpretability in A) WOSL, B) LWCL, and C) GWCL using real textual datasets.

4.4.1 Text Corpus We used the following four textual corpora including reviews and their sentiment tags for evaluation.

1) EcoReview I. This dataset included comments for the current economic trend and their sentiment tags [1]. This dataset was collected by workers closely related to the regional economy. We used oldest 20,000 positive comments and oldest 20,000 negative comments as the training dataset, oldest 2,000 positive and oldest 2,000 negative comments of the remaining comments as the validation dataset, and newest 4,000 positive and newest 4,000 negative comments of the remaining comments as the test dataset.

2) EcoReview II. This dataset involves Japanse comments[2] on the future economic trend between 2002 and 2017 and their sentiment tags. This dataset included 26,000 positive comments and 26,000 negative comments. We used oldest 35,000 positive comments and oldest 35,000 negative comments as the training dataset, oldest 2,000 positive and oldest 2,000 negative comments of the the remaining comments as the validation dataset, and newest 4,000 positive and newest 4,000 negative comments of the the remaining comments as the test dataset. The vocabulary size v was 11,130.

3) Yahoo reviews. This dataset is composed of comments on stocks and their long (positive) or short (negative) attitude tags, extracted from financial micro-blogs.[2] between September 2015. We used the oldest 40,000 posts (30,612 positive posts and 9,388 negative posts) as the training dataset, the oldest 5,000 posts from the remaining posts (3,387 positive posts and 1,613 negative posts) as the validation dataset, and the newest 10,000 posts (7,538 positive posts and 2,462 negative posts) as the test dataset. The vocabulary size v was 33,08.

4) Sentiment 140. This dataset contains 800,000 positive tweets and 800,000 negative tweets [3]. We used the first 650,000 positive tweets and 650,000 negative tweets as the training dataset, the next 50,000 positive tweets and 50,000 negative tweets as the validation dataset, and the remain 100,000 positive tweets and 100,000 negative tweets as the test dataset.

EcoRevs and Yahoo review are Japanese datasets, and Sentiment 140 is English. We used them to verify whether the SINN can be applied irrespective of the language or domain. We divided each dataset into training, validation, and test datasets, as outlined in Table 1.

4.4.2 Evaluation Metrics After developing the SINN with the training and validation datasets, we evaluated the interpretability in A) WOSL, B) LWCL, and C) GWCL as follows.

A) Evaluation for WOSL For this evaluation, we used the economic, Yahoo, and LEX word polarity list[4], which include words along with their positive or negative polarities. The economic and Yahoo word polarity lists include Japanese economic terms, and the LEX word polarity list includes English terms. If we used the EcoRev I or II, Yahoo reviews, and Sentiment 140 in training, then, we utilized the economic, Yahoo, and LEX word polarity list, respectively.

[1] https://www5.cao.go.jp/keizai3/watcher-e/index-e.html
[2] http://textream.yahoo.co.jp
[3] https://www.kaggle.com/kazanova/sentiment140
[4] http://quanteda.io/reference/data_dictionary_LSD2015.html

We used only the terms that appeared in the training dataset and not used in *LEXIL*. Table 1 summarizes the number of words used in this evaluation. We evaluated the interpretability of the WOSL based on the agreement between the polarities of word w_i (= answer) and w_i^p (= prediction) and used the macro F_1 score for the evaluation basis.

B) Evaluation for LWCL We prepared the Economy, Yahoo, and message annotated datasets for this evaluation. The Economy annotated dataset scontains 2,200 reviews (1,100 positive and 1,100 negative) from the test dataset of EcoRev I. The Yahoo annotated dataset includes 1,520 reviews (760 positive and 760 negative) from the test dataset of Yahoo reviews. The message annotated dataset has 10,258 reviews obtained from the test datasets in the SemEval tasks [41,47]. In these datasets, part of the terms in the reviews had word-level sentiment shift tags indicating whether the sentiments of the terms were shifted (1: shifted) or not (0: non-shifted) as follows.

(1) In total, we are in a $bull^{(0)}$ market.
(2) This room is not $clean^{(1)}$.
(3) Products in this shop are too $expensive^{(1)}$.

Using these tags, we evaluated the interpretability of the LWCL according to the agreement between the sentiment shift tag of $w_{it}^{\mathbf{Q}}$ and the polarity of $s_{it}^{\mathbf{Q}}$ (shifted: $s_{it}^{\mathbf{Q}} < 0$ and non-shifted: $s_{it}^{\mathbf{Q}} > 0$). We used the macro F_1 score for the evaluation basis.

C) Evaluation for GWCL We used the global important point tags included in the Economy and Yahoo annotated datasets for this evaluation, which indicate whether each term in a review is important (1) or not (0) for deciding the document-level polarity of the review as follows.

(1) $We^{(0)}$ $are^{(0)}$ $in^{(0)}$ $a^{(0)}$ $bull^{(1)}$ $market^{(1)}$.
(2) $This^{(0)}$ $room^{(0)}$ $is^{(0)}$ $not^{(1)}$ $clean^{(1)}$.

Using these tags, we evaluated the interpretability of the GWSL based on the correlation between $\{\alpha_t^{\mathbf{Q}}\}_{t=1}^n$ and the word-level global important points. We used the Pearson correlation for this evaluation.

In the evaluations for the LWCL and GWCL, we used the Economy, Yahoo, and message annotated datasets when we developed SINN with the EcoReviews, Yahoo reviews, and Sentiment 140, respectively. We only employed tags of terms that were not used in LEXIL and appeared in the training dataset. Table 1 summarizes the numbers of tags used.

4.4.3 Dataset Creation Details The annotated datasets and word polarity lists were created in the following way.

(A) Word polarity list

The economic word polarity list and the Yahoo word polarity list were developed in the following ways. In developing polarity word list for the EcoReview I, annotators picked up the following words from the top 2500 words most appeared in the EcoReview I.

- Words relate to trend (e.g., up and down).
- Words related to purchase (e.g., buy and sell).
- Words relate to evaluation (e.g., good and bad).

- Words related to abnormal (e.g., strange).
- Words related to events which influence the Economy in Japan (e.g., Olympic (positive), Abenomics (positive), Bubble (positive), heavy snowfall (negative)).

In developing the polarity word list for the Yahoo review, annotators picked up the following words from the top 5000 words most appeared in the Yahoo review.

- Words relate to trend (e.g., up and down).
- Words relate to evaluation such as good and bad.
- Words related to stock trading such as Buy (positive), Hold (positive), Sell (negative) and Run (negative.)
- Words related to events which influence the company and its stock price such as dirty (negative), insider (negative), and arrest (negative.)

Two annotators, who were individual investors, assigned these tags. The Cohen's Kappa score was 0.961 and 0.961 in the Economic dataset and Yahoo dataset. We used terms which were assigned the positive tags by both annotators as positive words and used terms which were assigned the negative tags by both annotators as negative words.

(B) Sentiment shift tags

Economic and Yahoo annotated dataset Sentiment shift tags of the economic and Yahoo annotated dataset were annotated manually by two individual professionals. Annotators assigned "sentiment shift tag (GPE)" to each word that appeared in a comment if the word satisfies the following conditions:

- the word was included in the polarity word list (i.e., the word was positive or negative word), and
- the word shifted by the following two types of words:

  function-word negation the words that were negated by function-word such as "not" and "never," or "too."

  contend-word negation the words that were negated by the contend-word such as "fall" and "drop."

  Here, annotators did not assign "sentiment shift tag (GPE)" to the words that were shifted by 'Paradox words such as "however," "though," "nevertheless," and "but," and we excluded these words from the evaluation in this annotation.

The decisions of whether "positive" word or "negative" word were determined using the word polarity list. The Cohen's Kappa score was 0.78 in Economic annotation dataset, and 0.75 in Yahoo annotation dataset.

They used the brat ([5] for this annotation task).

[5] http://brat.nlplab.org/

We only used the tags of words which were included in the polarity word list and did not use the tags of words which were not included in the polarity word list.

There were 1157 shifted and 5168 not-shifted terms, and 390 shifted and 2,591 not-shifted terms, in the Economic annotated dataset, and the Yahoo annotated dataset, respectively.

Message annotated datasets Sentiment shift tags of the message annotated datasets were developed in the following way. We first collected all the SMS and Tweets that were included in the test dataset of SemEval 2013 task-2 [41] or SemEval 2014 task-9 [47] and can be downloaded on January 20, 2019. In total, we collected 10258 messages. After that, we developed sentiment shift tags using these messages, the phrase-level sentiment tags (positive or negative) provided by the SemEval tasks, and the LEX dictionary [67] in the following way: We assigned sentiment shift tags to only terms which were assigned phrase-level sentiment tags and were included in the LEX dictionary. If the phrase-level sentiment of a term and the original word-level sentiment provided by the LEX dictionary were different, then the term was tagged as "shifted": in other cases, the term was tagged as "not shifted."

(C) Contextual word-level sentiment tags

If a term was tagged as positive in the polarity list, and was not tagged as "shifted," or it was tagged as negative and was tagged as "not shifted," it was tagged as contextual negative. In the opposite cases, it was tagged as contextual positive.

4.4.4 SINN Development Setting We developed the SINN using each training and validation datasets in the following settings.

PLEXIL and PJSP Learning In the part of Lexical Initialization, we used part of Japanese financial word sentiment dictionary (JFWS dict) and the Vader word sentiment dictionary (Vader dict) [17]. These dictionaries contain words with sentiment scores. After excluding words with zero sentiment scores from these dictionaries, we extracted 200 words that appeared mostly in each training dataset from them and used their sentiment scores in PLEXIL or PJSP Learning. The percentage of sentences covered by the above 200 terms was 3.4%, 4.1%, 0.7% and 7.5% in EcoRev I, EcoRev II, Yahoo, and Sentiment 140, respectively.

Others We calculated the word embedding matrix $\boldsymbol{W}^{em}$ with the skip-gram method (window size = 5) based on each textual corpus. We set the dimension of the hidden and embedding vectors to 200 and epoch to 50 with early stopping. We used the mean score of the five trials for evaluation. We used stratified sampling [69] to analyze imbalanced data, and the Adam optimizer [8], and the dropout [55] method (rate = 0.5).

4.4.5 Comparison for the Learning Strategy As explained in Section 3.5, the Lexical Initialization is expected to be important for realizing the interpretability of the SINN. In addition, the SSL regularization in PJSP is expected to have an effect to the interpretability in SINN. To investigate its effect, we compared the results of three types of SINNs, namely, $SINN^{Base}$, $SINN^{LEXIL}$, and $SINN^{JSPL}$. The structure of $SINN^{Base}$ is the same as that of the SINN; however, it is different from SINN in that the values of $\boldsymbol{W}^{p}$ were initialized according to $U(-1, 1)$ where $U(a, b)$ is a uniform distribution between a and b, that is, $SINN^{Base}$ is developed without the Lexical Initialization. $SINN^{LEXIL}$ represents the SINN developed with

LEXIL. $SINN^{JSPL}$ represents the SINN developed with PJSP learning.

4.4.6 Comparison Method To evaluate whether the interpretabilities of the WOSL, LWCL, and GWCL are sufficiently high, we compared the evaluation results of these layers and that of the corresponding comparison methods, respectively.

A) Interpretability in WOSL To evaluate the interpretability of WOSL, we compared the results of the SINN and following word-level original sentiment analysis methods: PMI [40], logistic fixed weight model (LFW) [58], sentiment-oriented NN (SONN) [34], and GINN [21]. PMI is a statistical analysis method, while the others are interpretable NN based methods.

B) Interpretability in LWCL To evaluate the interpretability of LWCL, we compared the results of the SINN with that of the baseline, NegRNN methods, and Recursive Neural Tensor Network (RNTN). In the baseline, we predicted $w_{it}^{\mathbf{Q}}$ as "shifted" if the document-level sentiment tag of $\mathbf{Q}$ predicted by the RNN and sentiment tag of the word $w_{it}^{\mathbf{Q}}$ assigned by the PMI were different and as "not shifted" in other cases. In NegRNN, we first developed the polarity shifting training data using the weighed frequency odds method [36], and then developed the RNN that predicts polarity shifts [11], and used this for prediction.

In RNTN [53], we detected the sentiment shift of each term by comparing the mean polarity score of the upper nodes of each node and the polarity score of each node. If polarities of these scores are different, then, its sentiment was judged to be shifted, and in the opposite case, it was decided as not shifted. We used the RNTN model developed from Sentiment Tree Bank [53], and this RNTN can be applied to only English texts; therefore, the results of RNTN model were provided to only Sentiment 140 annotated dataset. RNTN

C) Interpretability in GWCL To evaluate the interpretability of GWCL, we compared the evaluation result of SINN with that of the methods using the attention-based NNs: ATT [66], HN-ATT [66], SNNN [16], and LBSA [64]. We used the attention score of each model as the global word-level context score.

4.4.7 Result and Discussion

Overall Result Tables 2 indicate the results. The SINN significantly outperformed the other methods in most cases ($p < 0.05$ in five trials), demonstrating the high interpretability of the SINN. Moreover, the results of the SINNs and $SINN^{Base}$ demonstrate that the Lexical Initialization was important for realizing the interpretability of the SINN, as expected.

Ablation Analysis To analyze the results when PLEXIL used fewer words, we additionally evaluated the SINNs developed with 0, 50, 100, or 200 words in PLEXIL: $SINN^{LEXIL}$ (0), $SINN^{LEXIL}$ (50), $SINN^{LEXIL}$(100) or $SINN^{LEXIL}$ (200). Tables 3 summarize the result, showing that the interpretability of layers in SINN is sufficiently high even when we used only fifty terms in the LEXIL. This result demonstrates the practicality of our method because the result shows that we require only (1) a large number of reviews with their positive or negative sentiment tags, and (2) a small word sentiment dictionary composed of a few hundred word-level original sentiment scores, and do not require contextual word or phrase-level tags, or specific

Table 1: Dataset details for Text Corpus and Annotated data

Text Corpus	EcoRev I	EcoRev II	Yahoo	Sentiment 140	
Training					
positive reviews	20,000	35,000	30,612	650,000	
negative reviews	20,000	35,000	9,388	650,000	
Validation					
positive reviews	2,000	2,000	3,387	50,000	
negative reviews	2,000	2,000	1,613	50,000	
Test					
positive reviews	4,000	4,000	7,538	100,000	
negative reviews	4,000	4,000	2,462	100,000	
vocabulary size v	8,071	11,130	33,080	71,316	
Annotated data	EcoRev I	EcoRev II	Yahoo	Sentiment 140	
word polarity list					
Positive	348	337	422	1,843	
Negative	391	387	372	947	
sentiment shift tags					
Shifted tags	872	859	378	429	
Non-shifted tags	3,762	3,740	2,391	4,504	
word-level global important point tags					
Important tags (1)	6,632	6,631	1,526	-	
Unimportant tags (0)	62,652	62,652	48,890	-	
word-level and phrase-level contextual polarity tags					
Level	word	word	word	word	phrase
Shifted Negative	776	756	227	169	-
Non-shifted Negative	1,491	1483	1,187	1,294	-
Shifted Positive	96	96	151	260	-
Non-shifted Positive	2,271	2179	1,204	3,210	-
Negative (total)	2,267	2239	1,414	1,463	3,634
Positive (total)	2,367	2,275	1,355	3,470	5,907

knowledge for word-level contexts. In general, it is often difficult to prepare the sufficient volume of contextual word or phrase-level tags, and the specific knowledge for word-level contexts can not be available for non-grammatical documents such as tweets and posts on micro-blogs; whereas the fifty scores of terms can be easily collected using crows sourcing or other similar methods.

4.5 Experimental Evaluation for Word-level Contextual Sentiment Analysis Ability

This section experimentally evaluates the WCSA ability of the SINN in terms of the A) contextual word-level polarity, B) phrase-level polarity, and C) document-level polarity.

4.5.1 Evaluation Metrics

A) Contextual word-level polarity In the evaluation from this aspect, we evaluated the SINN in terms of the agreement between the polarity of word-level contextual sentiment for $w_{it}^{\mathbf{Q}}$ and the positive or negative polarity of $c_{it}^{\mathbf{Q}}$. We used the word-level contextual polarity tags included in the annotation datasets for this evaluation. They indicate the positive or negative word-level contextual polarities as follows.

Table 2: Evaluation Result for Interpretability

(A) Evaluation Result for WOSL

	EcoRev I	EcoRev II	Yahoo	Sentiment 140
PMI	.734	.745	.793	.733
LFW	.715	.740	.766	.725
SONN	.702	.724	.725	.705
GINN	.723	.755	.754	.735
$SINN^{Base}$	.492	.513	.487	.444
$SINN^{LEXIL}$	**.839**	**.856**	.817	.737
$SINN^{JSP}$	.829	.844	**0.820**	**0.753**

(B) Evaluation Result for LWCL

	EcoRev I	EcoRev II	Yahoo	Sentiment 140
Baseline	.660	.712	.579	.560
NegRNN	.536	.626	.564	.558
RNTN	-	-	-	.436
$SINN^{Base}$	.350	.440	.495	.365
$SINN^{LEXIL}$	**.800**	.821	.646	**.759**
$SINN^{JSP}$	.778	**.834**	**.719**	.746

(C) Evaluation Result for GWCL

	EcoRev I	EcoRev II	Yahoo	Sentiment 140
ATT	-.015	-.081	.062	–
HN-ATT	.108	.188	.262	–
SNNN	.281	.456	.192	–
LBSA	.333	.344	**.405**	–
$SINN^{Base}$	.053	.131	.017	–
$SINN^{LEXIL}$	**.588**	.508	.278	–
$SINN^{JSP}$	.542	**.518**	**.367**	–

Table 3: Ablation Analysis for the interpretability in SINN

(A) Evaluation Result for WOSL

	EcoRev I	EcoRev II	Yahoo	Sentiment 140
$SINN^{LEXIL}$ (0)	.492	.513	.487	.444
$SINN^{LEXIL}$ (50)	.844	.854	.802	.751
$SINN^{LEXIL}$ (100)	.842	.854	.816	.742
$SINN^{LEXIL}$ (200)	.839	.856	.817	.737

(B) Evaluation Result for LWCL

	EcoRev I	EcoRev II	Yahoo	Sentiment 140
$SINN^{LEXIL}$ (0)	.350	.440	.495	.365
$SINN^{LEXIL}$ (50)	.776	.837	.659	.739
$SINN^{LEXIL}$ (100)	.815	.857	.670	.742
$SINN^{LEXIL}$ (200)	.800	.821	.646	.759

(C) Evaluation Result for GWCL

	EcoRev I	EcoRev II	Yahoo	Sentiment 140
$SINN^{LEXIL}$ (0)	.053	.131	.017	–
$SINN^{LEXIL}$ (50)	.602	.522	.263	–
$SINN^{LEXIL}$ (100)	.637	.535	.285	–
$SINN^{LEXIL}$ (200)	.588	.508	.278	–

(1) In total, we are in a $bull^+$ market.
(2) This room is not $clean^-$.
(3) Products in this shop are too $expensive^-$.

We used the macro average scores between the macro F1 score for the shifted terms and that for the non-shifted terms for the evaluation basis to test whether each method could accurately correspond to both shifted and non-shifted terms. We excluded the terms used in the Lexical Initialization, for fairness in comparison with the other methods.

B) Phrase-level polarity In the evaluation from this aspect, we evaluated the SINN in terms of the agreement between the polarity of the phrase-level sentiment for a phrase $\{w_{im}^{\mathbf{Q}}, \cdots, w_{in}^{\mathbf{Q}}\}$ and the polarity of $\sum_{t=n}^{m} c_{it}^{\mathbf{Q}}$ using the phrase-level polarity tags in the message annotated dataset. These tags indicate the positive or negative phrase-level polarity as follows.

(1) In total, we are in a $\{bullmarket\}^+$.
(2) This room is $\{notclean\}^-$.
(3) Products in this shop are $\{tooexpensive\}^-$.

C) Document-level polarity In the evaluation from this aspect, we evaluated the SINN in terms of the agreement between the positive or negative polarity of the review $\mathbf{Q}$ and the polarity of $\sum_{i=1}^{L}\sum_{t=1}^{T} c_{it}^{\mathbf{Q}}$. We applied the document-level sentiment tags of reviews in test datasets for this evaluation and used the macro F_1 score as the evaluation basis.

Table 1 summarizes the numbers of tags used in the evaluations A), B), and C).

4.5.2 Comparison Methods We compared the result of SINN with those from the following word-level sentiment analysis methods: RNTN, PMI, LFW, SONN, GINN, Grad + RNN [27], LRP + RNN [1], and IntGrad + RNN [56], for this evaluation. The last three approaches are the developed LSTM interpretation-based approaches.

Moreover, to grasp the upper limitation in the predictability, we compared the predictability of the SINN with that of the follwoing DNNs: Convolutional NN (CNN) [30], RNN, ATT [66], HN-ATT [66], SNNN [16], and LBSA [64].

4.5.3 Result and Discussion Tables 4 summarize the results. The $SINN^{LEXIL}$ and $SINN^{JSP}$ significantly outperformed the WCSA methods ($p < 0.05$ in five trials). These results demonstrate the high WCSA ability of SINN, though its interpretability is high as demonstrated in Section 4.4. In addition, in most cases, both the $SINN^{LEXIL}$ and $SINN^{JSP}$ outperformed the compared DNNs in terms of the document-level analysis. This result demonstrates the usefulness of our method.

4.5.4 Output Example We experimentally demonstrate that both the interpretability and WCSA ability are high in SINN. We then introduce the text-visualization examples produced by SINN (Figs. 6). Like these examples, SINN can analyze word-level contextual sentiments in an interpretable manner. From the first example in Japanese, we can see that the word-level contextual sentiment of "Fuel (Increase)" is strongly negative because its word-level original sentiment and local word-level context scores are positive and nega- tive, respectively, and its global word-level context score is large. In addition, we can see that its sentiment shift occurs due to the left-oriented (i.e., backward) sentiment shift by "Nai (Not)" from the values in the LWCL. On the other hand, in the second example in English, we can see that the word-level contextual sentiment of "great" is strongly negative because its word-level original sentiment and local word-level context scores are positive and negative, respectively, and its global word-level context score is large. Moreover, we can see that its sentiment shift occurs due to the right-oriented (i.e., forward) sentiment shift by "Not" from the values in the LWCL.

Additionally, We briefly checked the validity for the explanation framework in SINN in terms of the personal communication check. We asked for five financial professionals who have experienced in a security company, financial bank, or asset management company whether the explanation framework from SINN is valid or not, using the output example from SINN. All of them answered as "Yes" to this question. From this brief check, we can see that the text-visualization framework in SINN is valid.

4.6 Conclusion

This chapter introduces a sentiment interpretable neural network (SINN) as a specific example of BINN. This chapter applies LEXIL and JSP learning in a practical manner to the development of SINN. We experimentally demonstrated that PLEXIL and JSP learning were effective for improving the interpretabilities of $SINN^{LEXIL}$ and $SINN^{JSP}$ as well as that both the interpretability and WCSA ability of the SINNs were high. The SINNs outperformed the comparative methods in the WCSA task on several domain datasets including Japanese and English datasets, while also featuring high interpretability. This success of the SINNs demonstrates that the proposed LEXIL and JSP learning can be utilized in actually developing

Table 4: Evaluation Result for WCSA Ability

Evaluation Result in (A) Word-level polarity or (B) Phrase-level polarity					
	EcoRev I	EcoRev II	Yahoo	Sentiment 140	
Level	word	word	word	word	phrase
PMI	.578	.548	.575	.631	.822
RNTN	-	-	-	.670	.620
Grad + RNN	.578	.621	.601	.681	.743
IntGrad + RNN	.607	.621	.625	.679	.796
LRP + RNN	.597	.518	.579	.638	.808
LFW	.549	.545	.578	.587	.749
SONN	.555	.542	.566	.600	.787
GINN	.569	.555	.577	.623	.831
$SINN^{Base}$	.550	.605	.573	.750	.821
$SINN^{LEXIL}$	**.719**	.741	.651	**.787**	**.863**
$SINN^{JSP}$	.687	**.756**	**.699**	.777	.849

(C) Evaluation Result in Document-level polarity				
	EcoRev I	EcoRev II	Yahoo	Sentiment 140
RNTN	-	-	-	
LR	.878	.879	.741	.785
LFW	.876	.840	.751	.745
SONN	.863	.876	.717	.776
Grad + RNN	.870	.899	.724	.718
IntGrad + RNN	.909	.929	.750	.755
LRP + RNN	.909	.909	.751	.818
CNN	.894	.911	.757	.820
RNN	.922	.932	.749	**.837**
ATT	.924	.937	.750	.835
HN-ATT	.927	.940	.750	**.837**
SNNN	.918	.928	.752	.827
LBSA	.922	.941	.762	.832
$SINN^{Base}$	.922	.941	.731	.834
$SINN^{LEXIL}$	.928	.942	**.766**	.834
$SINN^{JSP}$	**.929**	**.946**	.760	.833

interpretable NNs. However, we can not observe the effect of SSL regularization in this chapter. In the development of SINN, the SSL regularization is not required; however, this requirement is not always established. In the next chapter, we introduce the example where the SSL regularization is required for the success of the interpretability in each layer of the BINNs.

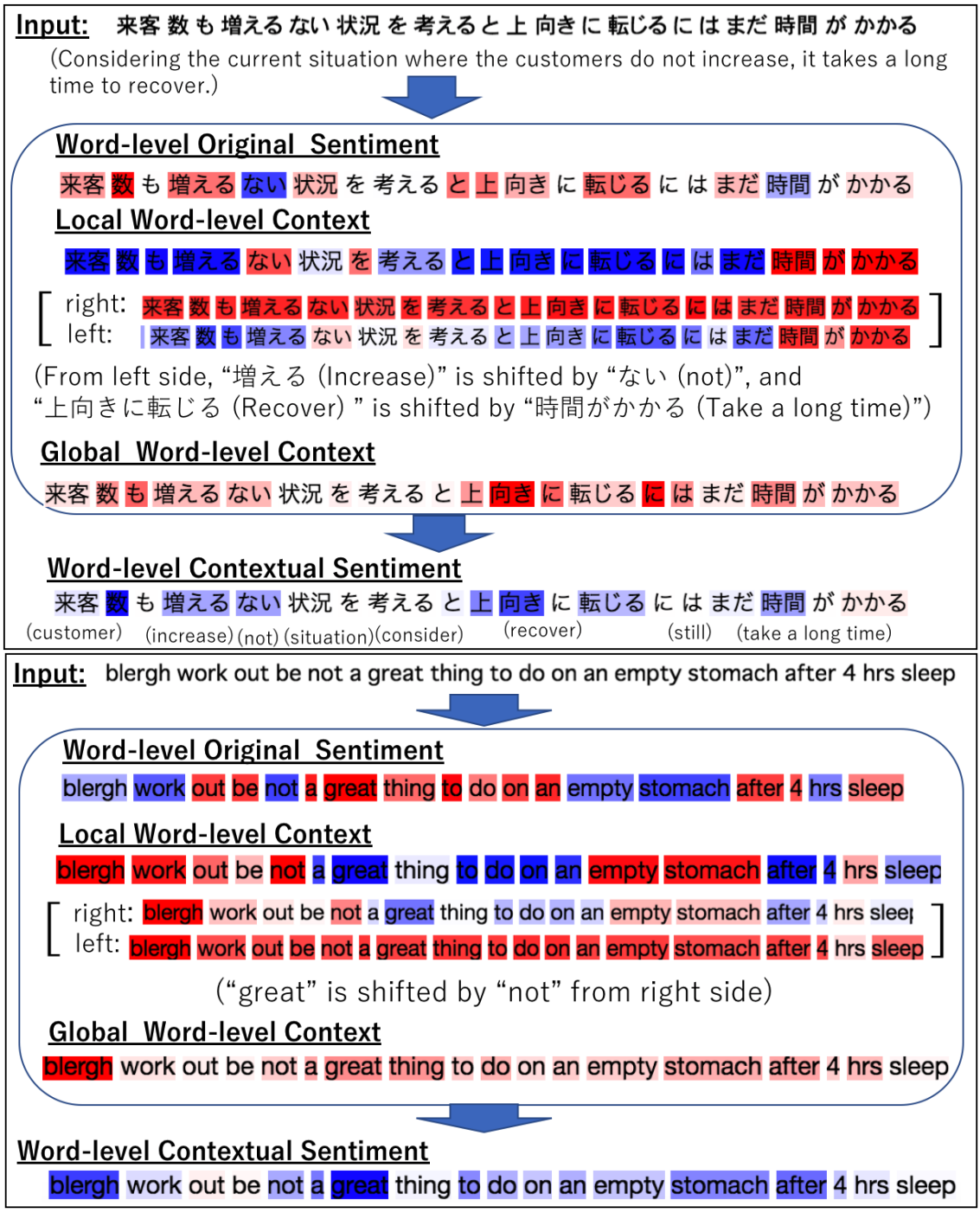

Figure 6: Text-visualization example by SINN. Colors mean their polarities (red: positive, blue: negative). The upper and below are reviews in EcoRev and Sentiment 140.

Chapter 5

Sentiment Shift Neural Network (SSNN)

This chapter introduces the sentiment shift neural network (SSNN) [23] as the second example of BINN. We introduce the SSNN as an example where the SSL regularization is required to satisfy the interpretability of each layer. In some situations, the PLEXIL failed to satisfy the interpretability in the layers of SSNN; however, PJSP learning succeeded it in such a situation. The success of the SSNN development using PJSP learning means that the proposed JSP learning can be utilized in the actual development of interpretable NNs.

We first introduce the SSNN in Section 5.1 and then explain the detailed SSNN structure and the learning strategy of SSNN in Section 5.2. We then experimentally evaluate the LEXIL and JSP learning using real textual datasets in Sections 5.3 and 5.4, and conclude this chapter 5.5.

5.1 Overview

As a specific example of BINN, this chapter considers the SSNN [23]. This SSNN includes includes the following three interpretable layers: word-level original sentiment layer (WOSL), sentiment shift layer (SSL), and word-level contextual sentiment layer (WCSL) as shown in Fig. 7. The WOSL, SSL, and WCSL represent the word-level original sentiment, sentiment shift, and contextual sentiment of each term in a review, respectively. WOSL is represented in a word sentiment dictionary manner. SSL is represented using long short-term memories (LSTM) cells [50]. The values of WCSL are represented by multiplying the values of WOSL and SSL. Then, the sum of the values in WCSL represents the document-level sentiment of the review.

Therefore, SSNN is valuable in a case where the extraction of the word-level original sentiment, sentiment shift, and word-level contextual sentiment are required in the explanation, as shown in Fig. 8.

5.2 Structure of SSNN

This section explains the detailed structure of SSNN.

Notation Before the explanation, we define several symbols. Let $\{(\mathbf{Q}_i, d^{\mathbf{Q}_i})\}_{i=1}^{N}$ be a training dataset where N is the training data size, $\mathbf{Q}_i$ is a review, and $d^{\mathbf{Q}_i}$ is its sentiment tag (1 is

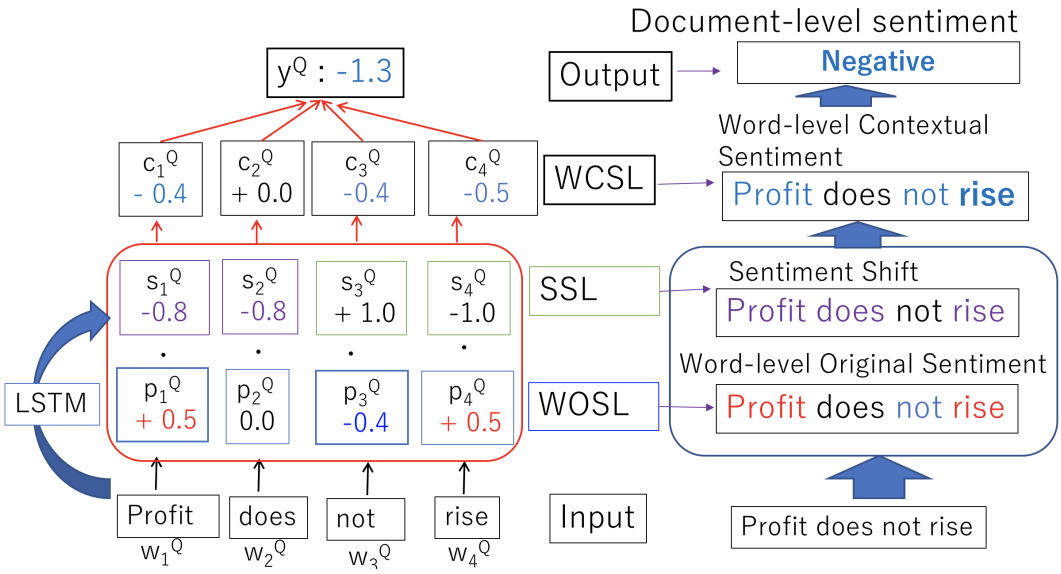

Figure 7: SSNN

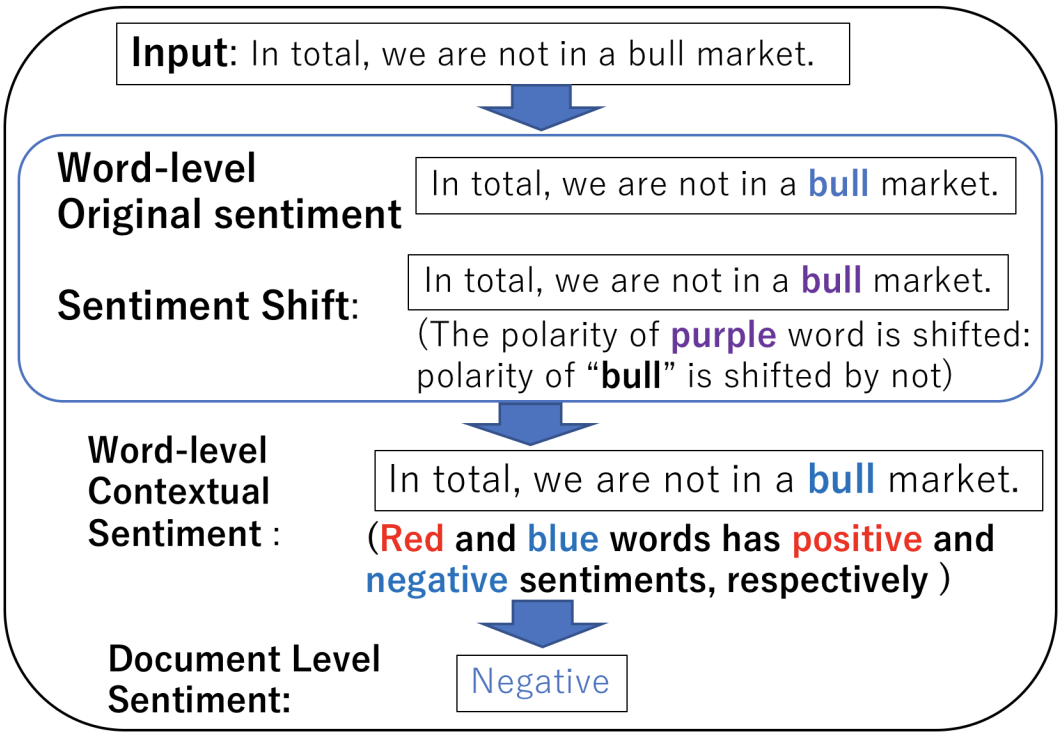

Figure 8: Goal: development of NN that can explain its prediction results using three types of scores

positive and 0 is negative). Let $\{w_i\}_{i=1}^{v}$ be the terms that appear in a text corpus of a dataset, and v be the vocabulary size. We define the vocabulary index of word w_i as $I(w_i)$. Therefore, $I(w_i) = i$. Let $\boldsymbol{w}_i^{em} \in \mathbb{R}^e$ be an embedding representation of word w_i, and the embedding matrix $\boldsymbol{W}^{em} \in \mathbb{R}^{v \times e}$ be $[\boldsymbol{w}_1^{emT}, \cdots, \boldsymbol{w}_v^{emT}]^T$ where e is the dimension size of word embedding and $\|\boldsymbol{w}_i^{em}\|_2 = 1$ for each i. $\boldsymbol{W}^{em}$ is the constant value obtained using the skip-gram method [39] and a text corpus.

WOSL Given a review $\mathbf{Q} = \{w_t^{\mathbf{Q}}\}_{t=1}^{T}$, this layer converts the words $\{w_t^{\mathbf{Q}}\}_{t=1}^{T}$ to word-level original sentiment representations $\{p_t^{\mathbf{Q}}\}_{t=1}^{T}$:

$$p_t^{\mathbf{Q}} = w^p_{I(w_t^{\mathbf{Q}})} \tag{30}$$

where $\boldsymbol{W}^p \in \mathbb{R}^v$ represents the original sentiment scores of words, and w_i^p is the $i-$th element of $\boldsymbol{W}^p$. The w_i^p value represents the original sentiment score of word w_i.

SSL First, this layer, converts terms $\{w_t^{\mathbf{Q}}\}_{t=1}^{T}$ in a review $\mathbf{Q}$ into their word-level embeddings $\{\boldsymbol{e}_t^{\mathbf{Q}}\}_{t=1}^{T}$ using $\boldsymbol{W}^{em}$, and converts them to context representations $\{\overrightarrow{\boldsymbol{h}}_t^{\mathbf{Q}}\}_{t=1}^{T}$ and $\{\overleftarrow{\boldsymbol{h}}_t^{\mathbf{Q}}\}_{t=1}^{T} \in \mathbb{R}^e$ using forward and backward LSTMs, $\overrightarrow{\text{LSTM}}$ and $\overleftarrow{\text{LSTM}}$ [50]:

$$\overrightarrow{\boldsymbol{h}}_t^{\mathbf{Q}} = \overrightarrow{\text{LSTM}}(\boldsymbol{e}_t^{\mathbf{Q}}), \overleftarrow{\boldsymbol{h}}_t^{\mathbf{Q}} = \overleftarrow{\text{LSTM}}(\boldsymbol{e}_t^{\mathbf{Q}}). \tag{31}$$

It then converts $\{\overrightarrow{\boldsymbol{h}}_t^{\mathbf{Q}}\}_{t=1}^{T}$ and $\{\overleftarrow{\boldsymbol{h}}_t^{\mathbf{Q}}\}_{t=1}^{T}$ to right and left oriented sentiment shift representations, $\overrightarrow{s}_t$ and $\overleftarrow{s}_t^{\mathbf{Q}}$:

$$\overleftarrow{s}_t^{\mathbf{Q}} = \tanh(\boldsymbol{v}^{left^T} \cdot \overleftarrow{\boldsymbol{h}}_t^{\mathbf{Q}}), \overrightarrow{s}_t^{\mathbf{Q}} = \tanh(\boldsymbol{v}^{right^T} \cdot \overrightarrow{\boldsymbol{h}}_t^{\mathbf{Q}}).$$

Here, $\boldsymbol{v}^{right}, \boldsymbol{v}^{left} \in \mathbb{R}^e$ are parameter values. $\overrightarrow{s}_t^{\mathbf{Q}}$ and $\overleftarrow{s}_t^{\mathbf{Q}}$ denote whether or not the sentiment of $w_t^{\mathbf{Q}}$ is shifted by the left-side and right-side terms of $w_t^{\mathbf{Q}}$: $\{w_{t'}^{\mathbf{Q}}\}_{t'=1}^{t-1}$ and $\{w_{t'}^{\mathbf{Q}}\}_{t'=t+1}^{n}$, respectively.

Finally, this layer converts them into word-level sentiment shift scores $\{s_t^{\mathbf{Q}}\}_{t=1}^{T}$:

$$s_t^{\mathbf{Q}} := \overrightarrow{s}_t^{\mathbf{Q}} \cdot \overleftarrow{s}_t^{\mathbf{Q}}. \tag{32}$$

$s_t^{\mathbf{Q}}$ denotes whether the sentiment of $w_t^{\mathbf{Q}}$ is shifted ($s_t^{\mathbf{Q}} < 0$) or not ($s_t^{\mathbf{Q}} \geq 0$).

WCSL By using the SSL and WOSL, this layer converts $\{p_t^{\mathbf{Q}}\}_{t=1}^{T}$ into word-level local contextual sentiment representations $\{c_t^{\mathbf{Q}}\}_{t=1}^{T}$:

$$c_t^{\mathbf{Q}} = s_t^{\mathbf{Q}} \cdot p_t^{\mathbf{Q}}. \tag{33}$$

Output Finally, SSNN outputs a document-level sentiment $y^{\mathbf{Q}}$ using $\{c_t^{\mathbf{Q}}\}_{t=1}^{T}$:

$$y^{\mathbf{Q}} = \textstyle\sum_{t=1}^{T} c_t^{\mathbf{Q}}$$

where $y^{\mathbf{Q}} > 0$ ($y^{\mathbf{Q}} < 0$) means positive (negative).

SSNN can be developed with the interpretability using the LEXIL or PJSP learning in an ideal case. Through LEXIL or PJSP learning, WOSL, SSL, and WCSL learn to represent their corresponding sentiments, gradually. After the learning has finished, SSNN can analyze document-level sentiment through extracting the word-level original sentiment, sentiment shift, and word-level contextual sentiment from WOSL, SSL, and WCSL, respectively, in an ideal case where (1) the size of S^d is large enough to satisfy $S^* \in \Omega(\Phi(S^d))$, and (2) $S^d \in S^*$ is satisfied.

5.3 Experimental Evaluation for Explainability

This section experimentally evaluates the explanation ability of the proposed method in terms of the interpretability in A) WOSL, B) SSL, and C) WCSL.

5.3.1 Text Corpus This evaluation used the same text corpus dataset used in Sections 4.4 and 4.5, namely, EcoRev I, EcoRev II, Yahoo Rev, and Sentiment 140.

EcoRevs and Yahoo Rev are Japanese datasets, and Tweets is English. We used them to verify whether the SSNN can be applied irrespective of the language or domain. We divided each dataset into training, validation, and test datasets, as outlined in Table 1.

5.3.2 SSNN Development Setting We developed the SSNN using each training and validation datasets in the following settings.

Lexicon Initialization Part of Japanese financial word sentiment dictionary (JFWS dict) and the Vader word sentiment dictionary (Vader dict) [17] were used in Lexicon Initialization. They contain words with sentiment scores. After excluding words with zero sentiment scores from these dictionaries, we extracted 200 words that appeared mostly in each training dataset from them and used their sentiment scores in Lexicon Initialization.

Others We calculated the word embedding matrix $\boldsymbol{W}^{em}$ with the skip-gram method (window size = 5) based on each textual corpus. We set the dimension of the hidden and embedding vectors to 200 and epoch to 50 with early stopping. We set λ to 0.01. We used the mean score of the five trials for evaluation.

5.3.3 Evaluation Metrics We evaluated the interpretability in A) WOSL, B) SSL, and C) WCSL of the SSNN as follows.

A) Evaluation for WOSL We evaluated the interpretability of WOSL in the same manner as described in Section 4.4.

This evaluation used the economic, Yahoo, and LEX word polarity list (`http://quanteda.io/reference/data_dictionary_LSD2015.html`), which include words along with their positive or negative polarities. The economic and Yahoo word polarity lists include Japanese economic terms, and the LEX word polarity list includes English terms. If we used the EcoRev I or II, Yahoo reviews, and Tweets in training, then, we utilized the economic, Yahoo, and LEX word polarity lists, respectively. We used only terms that appeared in the training dataset and are not included in S^d. Table 1 summarizes the number of words used in this evaluation. We evaluated the interpretability of the WOSL based on the agreement between the polarities of word w_i (= answer) and w_i^p (= prediction) and used the macro F_1 score for the evaluation basis.

B) Evaluation for SSL We utilized the word-level sentiment shift tags indicating whether the sentiments of the terms were shifted (1: shifted) or not (0: non-shifted) included in the Economy, Yahoo, and message annotated datasets (Table 1) for this evaluation.

Using these tags, we evaluated the interpretability of the SSL according to the agreement between the sentiment shift tag of $w_t^{\mathbf{Q}}$ and the polarity of $s_t^{\mathbf{Q}}$ (shifted: $s_t^{\mathbf{Q}} < 0$ and non-shifted: $s_t^{\mathbf{Q}} > 0$). We used the macro F_1 score for the evaluation basis.

C) Evaluation for WCSL In the evaluation from this aspect, we evaluated the SSNN in terms of the agreement between the polarity of word-level contextual sentiment for $w_t^{\mathbf{Q}}$ and the positive or negative polarity of $c_t^{\mathbf{Q}}$. We used the *word-level contextual polarity* tags included in

the annotation datasets (Table 1) for this evaluation.

We used the macro average scores between the macro F_1 score for the shifted terms and that for the non-shifted terms for the evaluation basis to test whether each method could accurately correspond to both shifted and non-shifted terms. We excluded the terms used in the Lexicon Initialization, for fairness in comparison with the other methods.

5.3.4 Comparison for the learning strategy As explained in Section 3.5, Lexicon initialization and SSL regularization are important for realizing the interpretability of SSNN. To investigate their effects, we compared the results of three types of SSNNs, namely, $SSNN^{Base}$, $SSNN^{LEXIL}$, and $SSNN^{JSP}$ Their structures are the same as that of the SSNN; however, they are different from SSNN in the following way:

1) $SSNN^{Base}$ was learned without lexicon initialization or SSL regularization. The values of $\boldsymbol{W}^p$ in $SSNN^{Base}$ were initialized according to $U(-1, 1)$ where $U(a, b)$ is a uniform distribution between a and b and $SSNN^{Base}$ was learned using $L^{\mathbf{Q}}_{doc}$ instead of $L^{\mathbf{Q}}_{joint}$.

2) $SSNN^{LEXIL}$ was learned without SSL regularization, that is, it was learned using $L^{\mathbf{Q}}_{doc}$ instead of $L^{\mathbf{Q}}_{joint}$. We used the same 200 words used in SSNN (200) in the Lexicon initialization for $SSNN^{LEXIL}$.

3) $SSNN^{JSP}$ was learned was learned using the PJSP learning. To analyze the results in cases where fewer words were used, we evaluated the $SSNN^{JSP}$ developed with 50, 100, or 200 words: $SSNN^{JSP}$ (50), $SSNN^{JSP}$ (100) or $SSNN^{JSP}$ (200).

5.3.5 Comparison Method In addition, to evaluate the interpretability of each layer in SSNN, we compared the results of each layer in SSNN with that of the corresponding comparison methods as follows.

A) Interpretability in WOSL To evaluate the interpretability of WOSL, we compared the results of the SSNN and following word-level original sentiment analysis methods: PMI [40], logistic fixed weight model (LFW) [58], sentiment-oriented NN (SONN) [34], and GINN [21]. PMI is a statistical analysis method, while the others are interpretable NN based methods.

B) Interpretability in SSL To evaluate the interpretability of SSL, we compared the results of the SSNN with that of the baseline, NegRNN methods, and RNTN [53].

C) Interpretability in WCSL To evaluate the interpretability of WCSL, we compared the result of SSNN with those from the following word-level sentiment analysis methods: PMI, LFW, SONN, GINN, Grad + RNN [27], LRP + RNN [1], and IntGrad + RNN [56], for this evaluation. The last three approaches are the developed LSTM interpretation-based approaches.

5.3.6 Result and Discussion

Overall Result Tables 5 indicate the results. The SSNN significantly outperformed the other comparative methods in most cases ($p < 0.05$ in five trials), demonstrating the high interpretability of the SSNN. Moreover, the results of SSNN (50) indicate that we can develop SSNN with only fifty scores of word sentiment dictionary, showing the practicality of our approach.

Effect of Lexicon Initialization $SSNN^{LEXIL}$ outperformed the $SSNN^{Base}$ in all the cases, showing the effectiveness of Lexicon Initialization to the improvement of the interpretability, as

expected in Section 3.5.
Effect of SSL regularization The results of interpretability in WOSL and SSL for the $SSNN^{LEXIL}$ and SSNNs demonstrate that SSL regularization was effective for improving the interpretability in them, as expected in Section 3.5. Especially, the $SSNN^{LEXIL}$ failed to detect the sentiment shift in EcoRev II and Yahoo. In this cases, $SSNN^{LEXIL}$ predict all the terms as non-shifted. Meanwhile, the $SSNN^{JSP}s$ succeeded even in these cases. These results crucially demonstrate the effect of SSL regularization.

5.4 Experimental Evaluation for Predictability

This section experimentally evaluates the predictability of the proposed approach.

5.4.1 Evaluation Metrics We evaluated the predictability of the SSNN based on whether it could predict the polarity of reviews in each test dataset. We used the macro F_1 score as the evaluation basis.
Comparison Method To demonstrate the predictability of the SSNNs, we compared the results of SSNNs with those of the following comparative methods: logistic regression model (LR), convolutional NN (CNN) model [30], a bidirectional recurrent NN model with LSTM cells (RNN), word attention network (ATT) model [66], hierarchical attention network (HN-ATT) model [66], sentiment, and negation neural network (SNNN) model [16], and lexicon-based supervised attention (LBSA) model [64]. The last six DNNs are known to have strong prediction ability.

5.4.2 Result and Discussion Tables 6 summarize the results. The SSNN significantly outperformed the LR, CNN, ATT, SNNN, and LBSA in most cases ($p < 0.05$ in five trials). In addition, the $SSNN^{JSP}s$ was as predictable as the HN-ATT in EcoRevs and outperformed it in the Yahoo review. These results demonstrate the high predictability of the $SSNN^{JSP}s$. Moreover, the SSNNs outperformed the $SSNN^{LEXIL}$ and $SSNN^{Base}$, showing that the Lexicon initialization and SSL regularization were effective for improving the predictability.

The experimental results in this section and Section 7.4.3 demonstrate that the proposed SSNN has both the high explanation ability and high prediction ability.

5.4.3 Text-Visualization Example This section introduces some examples of text-visualization produced by the $SSNN^{JSP}$ (200). Figure 9 shows the text-visualization examples. Users can explain the $SSNN^{JSP}$ (200)'s prediction process based on this type of text-visualizations.

From the first example in Japanese, we can see that the word-level contextual sentiment of "Fuel (Increase)" is strongly negative because its word-level original sentiment and local word-level context scores are positive and negative, respectively, and its global word-level context score is large. In addition, we can see that its sentiment shift occurs due to the left-oriented (i.e., backward) sentiment shift by "Nai (Not)" from the values in the SSL. On the other hand, in the second example in English, we can see that the word-level contextual sentiment of "great" is strongly negative because its word-level original sentiment and local word-level context scores are positive and negative, respectively, and its global word-level context score is large. Moreover, we can see that its sentiment shift occurs due to the right-oriented (i.e., forward) sentiment shift

Table 5: Evaluation Result for Explanaiability

(A) Evaluation Result for WOSL

	EcoRev I	EcoRev II	Yahoo	Tweets
PMI	0.734	0.745	**0.793**	0.733
LFW	0.715	0.740	0.766	0.725
SONN	0.702	0.724	0.725	0.705
GINN	0.723	0.755	0.754	0.735
$SSNN^{Base}$	0.525	0.472	0.516	0.493
$SSNN^{LEXIL}$	0.720	0.750	0.755	0.731
$SSNN^{JSP}$ **(200)**	**0.778**	**0.772**	0.776	**0.755**
$SSNN^{JSP}$ **(100)**	**0.788**	**0.777**	0.777	**0.751**
$SSNN^{JSP}$ **(50)**	**0.779**	**0.813**	0.767	**0.754**

(B) Evaluation Result for SSL

	EcoRev I	EcoRev II	Yahoo	Tweets
Baseline	0.660	0.712	0.579	0.560
NegRNN	0.536	0.626	0.564	0.558
RNTN	-	-	-	0.436
$SSNN^{Base}$	0.433	0.402	0.469	0.377
$SSNN^{LEXIL}$	0.480	0.800	0.500	0.710
$SSNN^{JSP}$ **(200)**	**0.806**	**0.804**	**0.662**	**0.713**
$SSNN^{JSP}$ **(100)**	**0.804**	**0.813**	**0.668**	**0.713**
$SSNN^{JSP}$ **(50)**	**0.800**	**0.798**	**0.690**	**0.729**

(C) Evaluation Result for WCSL

	EcoRev I	EcoRev II	Yahoo	Tweets
RNTN	-	-	-	0.670
PMI	0.578	0.548	0.575	0.631
Grad + RNN	0.578	0.621	0.601	0.681
IntGrad + RNN	0.607	0.621	0.625	0.679
LRP + RNN	0.597	0.518	0.579	0.638
LFW	0.549	0.545	0.578	0.587
SONN	0.555	0.542	0.566	0.600
GINN	0.569	0.555	0.577	0.623
$SSNN^{Base}$	0.538	0.582	0.549	0.716
$SSNN^{LEXIL}$	0.546	0.719	0.566	0.780
$SSNN^{JSP}$ **(200)**	**0.726**	**0.739**	**0.649**	**0.764**
$SSNN^{JSP}$ **(100)**	**0.713**	**0.727**	**0.640**	**0.760**
$SSNN^{JSP}$ **(50)**	**0.723**	**0.720**	**0.662**	**0.784**

by "Not" from the values in the SSL.

5.5 Conclusion

This chapter introduces a sentiment shift neural network (SSNN) as a specific example of BINN. In the experimental evaluation using textual datasets, in some situations, LEXIL failed to develop

Table 6: Evaluation Result for Predictability

	EcoRev I	EcoRev II	Yahoo	Tweets
LR	0.878	0.879	0.741	0.785
CNN	0.894	0.911	0.757	0.820
RNN	0.922	0.932	0.749	**0.837**
ATT	0.924	0.937	0.750	0.835
HN-ATT	**0.927**	**0.940**	0.750	**0.837**
SNNN	0.918	0.928	0.752	0.827
LBSA	0.922	**0.940**	0.762	0.832
$SSNN^{Base}$	0.884	0.924	0.753	0.828
$SSNN^{LEXIL}$	0.920	0.928	0.737	0.827
$SSNN^{JSP}$ **(200)**	**0.927**	**0.940**	**0.779**	0.835
$SSNN^{JSP}$ **(100)**	0.926	0.939	0.776	0.834
$SSNN^{JSP}$ **(50)**	0.925	**0.940**	0.770	0.834

SSNN; however, PJSP learning succeeded in even such cases. This result demonstrates that the SSL regularization is useful for improving the interpretability. This success of PJSP learning for SSNN development demonstrates that JSP learning can be utilized in real cases.

From this and the previous chapters, we can see that LEXIL and JSP learning can be applied to the BINNs in real situations. In the next two chapters, we apply these learning strategies to the more complex interpretable NNs than BINNs. It should be noted that in such complex NNs, PLEXIL and PJSP learning can not be directly available. However, even in such situations, we can apply them by converting them precisely. We introduce the concrete strategy for the conversion in the next two chapters.

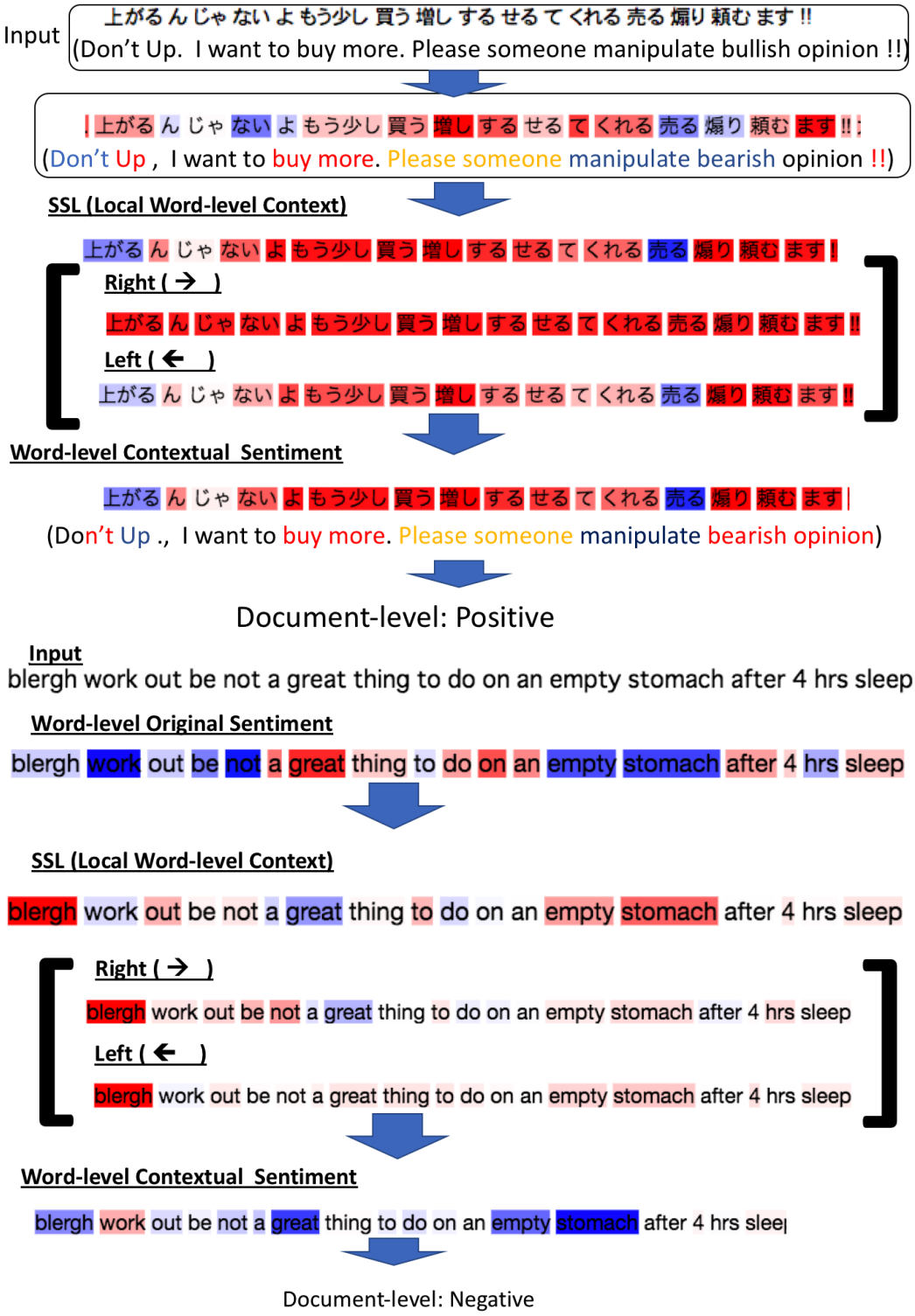

Figure 9: SSNN's text visualization examples in Yahoo (the above) and Tweets (the below). Colors of terms mean their positive (red) or negative (blue) polarities.

Chapter 6

Gradient Interpretable Neural Network (GINN)

This chapter introduces a gradient interpretable neural network (GINN) as an example of the application of LEXIL (proposed in Chapter 3). It should be noted that original LEXIL and PLEXIL (proposed in Chapter 4) can not be used in this case because the structure of the GINN is a little different from BINNs. Therefore, to develop GINN, we propose a novel development strategy for GINN called Importance of infiltration (II) algorithm by converting PLEXIL. From this application, we can see that the idea behind the proposition of LEXIL can be utilized in a flexible way and the proposed basic learning strategy for developing interpretable NNs can be utilized to many cases by appropriate conversions.

We first introduce the motivation for developing the GINN in Section 6.1 and then explain learning strategy for GINN in Section 6.2. In Section 6.3, we then experimentally evaluate our approach using real textual datasets and conclude this chapter in Section 6.4.

6.1 Introduction

6.1.1 Motivation and purpose Understanding technical documents such as financial reports and legal documents is often difficult for nonexperts. One of the reasons is that the meaning of a word or phrase in a specific domain may differ from the general meaning. For example, the word "climb" generally has a neutral sentiment, but in the financial domain, it means a price rise and has a positive sentiment; in this context, its meaning is similar to "increase", "rise", "boost", and "boom". This research aims to present sentiments and concepts included in words and phrases that appear in specialized documents and help nonexperts understand these documents. Therefore, a keyword list containing sentiments and similarity information in specialized fields is necessary; however, manually building a keyword list for each specialized area requires enormous effort. Therefore, we develop a method for constructing a keyword list from specialized documents using neural networks. We then propose a method of visualizing financial texts for nonexperts.

As an example, consider the sentence "It developed strong and powerful technologies. Poor price will rebound and surge." We aim to visualize this sentence on the right side of Fig. 10 in the following steps.

Step 1 "Strong" and "Powerful" are positive in the sense of the Trend concept , and "Rebound"

and "Surge" are positive in the sense of Ability concept.

Step 2 "Trends" and "Ability" concepts are important in this context.

Step 3 Therefore, this sentence is positive.

We define a set of synonyms and antonyms as *concept cluster* and sense of each concept cluster as *concept*. It would be helpful to describe some terms in each concept cluster for capturing the sense of the concept. By visualizing texts in the above manner, even nonexperts can easily capture the market sentiments of financial documents and explain the process of market sentiment analysis. We call this type of text-visualization framework as *Financial Text Visualizer*.

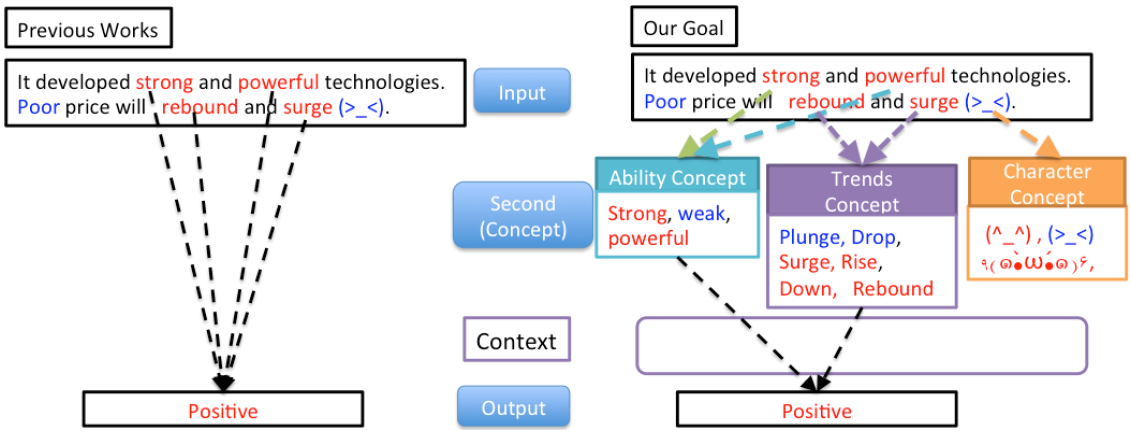

Figure 10: Previous visualization methods (left side) vs. our visualization goal (right side)

6.1.2 Main approach and problem settings Our aim is to develop market sentiment analysis models that can visualize documents as shown on the right side of Fig. 10. It is certain that linear models like support vector machine (SVM) [45] and methods for interpreting NNs [2, 15] can be useful for text visualization. Using these previous works, the visualization as shown on the left side of Fig. 10 can be realized. However, visualizing texts as shown on the right side of Fig. 10 by simply using these previous works is difficult because they alone cannot represent concepts. To achieve our goal, we propose a novel interpretable NN architecture called *gradient interpretable neural networks* (*GINN*) as shown in Fig. 11 [21]. Layers of GINN can be interpreted as follows.

The input layer represents the words in a document. Each node in the input layer corresponds to a word.

The second layer (concept layer) represents the sentiment scores of concept units. Each node in the second layer corresponds to a concept.

The output layer represents an entire sentiment value of the document.

Using GINN, we can visualize text in the following steps:

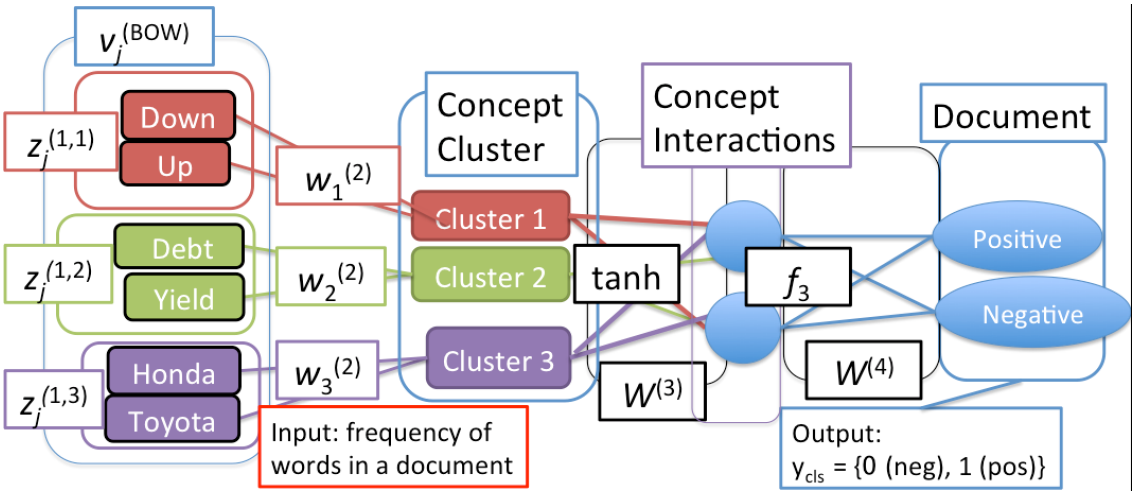

Figure 11: GINN architecture

Step 1: Extract word sentiment scores from the weight matrix between the input and second layers and concept sentiment scores from the second layer.

Step 2: Extract concept clusters that are important for the sentiment analysis decision using the gradient method [15].

Step 3: Extract an entire sentiment score of a document from the output layer.

To conduct the above text-visualization accurately, GINN must satisfy the following three conditions:

Condition 1: the connections between input and second layer nodes are determined by cluster analysis: if word X is in concept cluster Y, there is a link between X and Y,

Condition 2: when word X is in concept cluster Y, the value of the link between X and Y corresponds to the sentiment score of X, and

Condition 3: the output layer value is valid.

To evaluate whether Conditions 1–3 are satisfied and the validity of the text visualization by GINN, we evaluate the following *Interpretability*, *Cluster interpretability* and *Market mood predictability*.

Interpretability refers to the degree of accuracy with which the sentiment scores of words can be extracted from only weight matrix values between the input and second layers. Here, we consider words that frequently appear in positive (negative) documents than negative (positive) ones as positive (negative). We aim to satisfy Condition 2 by improving the interpretability.

Cluster interpretability refers to the validity of word clustering in the process of developing GINN.

Market mood predictability refers to the validity of the output layer value. We aim to satisfy Condition 3 by achieving high market mood predictability. This is equivalent to the predictability for an entire document sentiment.

By *clustering interpretability* and *interpretability*, we can evaluate the validity of Step 1 in the text visualization process by GINN. By *market mood predictability*, we can evaluate the validity of Steps 2 and 3 in the text visualization process. We aim to develop GINN whose structure satisfies Condition 1 and whose *interpretability*, *cluster interpretability* and *market mood predictability* are high.

The main contributions of this chapter are as follows.

- We proposed and developed a novel interpretable NN architecture called *GINN* that can visualize financial texts in the way as shown in Fig. 10. To develop this GINN, we propose a novel development strategy called *Importance of infiltration (II) algorithm* (Section 6.2).
- We experimentally demonstrated validity for the text visualizations by *GINN*. (Section 6.3).

The rest of this chapter is organized as follows. Section 6.2 introduces the method for constructing *GINN*. Section 6.3 demonstrates property of GINN using real data.

6.2 Importance of infiltration (II) algorithm

This section introduces the framework for developing *GINN*. We develop *GINN* according to the following steps.

Step 1 Prepare a dataset of documents and their positive or negative tags.

Step 2 Cluster words and construct the NN model (Subsection 6.2.1).

Step 3 Initialize parameter values using *Init* and obtain parameter values from the learning process using $Update^*$ (Subsection 6.2.2).

We refer to the series of flows from Step 2 to Step 3 as the II algorithm. Conditions 1 and 2–3 in Section 6.1.1 are realized by Step 2 and Step 3, respectively. We develop the II algorithm based on the following two ideas:
1. Assigning sentiment scores from a manually created polarity dictionary to specific edges between the input and second layers, and propagating the sentiment scores to the other edge values through the learning process. Consequently, each unit in the second layer will represent its sentiment information.
2. Necessitating the addition of certain limitations for the polarity propagation process, and such limitations should not reduce the *market mood predictability* of the model.

Ideas 1 and 2 are realized by *Init* and *Update* in Step 3, respectively.

6.2.1 Setup of NN model

To cluster words, we represent each word as a numerical vector using word2vec [39].

For a given number of clusters, K, we cluster similar words into the same cluster using the spherical K-means method [26] by their cosine distances. These clusters correspond to *concept clusters*. Using the results of clustering words, we construct an NN model that satisfies Condition 1 using the following layers:

Input layer: We assign a cluster number, k ($k = 1, 2, \cdots, K$) to each cluster and an ID number in the cluster to each word. Let $w_{k,i}$ be a word that is included in the kth cluster and whose ID number in the cluster is i, $\boldsymbol{z}_{j,i}^{(1,k)}$ be the frequency of the word $w_{k,i}$ in a document j, $n(k)$ be the number of words included in the kth cluster, m be $\sum_{k=1}^{K} n(k)$, and $\boldsymbol{z}_j^{(1,k)}$ be $\left[\boldsymbol{z}_{j,1}^{(1,k)}, \boldsymbol{z}_{j,2}^{(1,k)} \cdots, \boldsymbol{z}_{j,n(k)}^{(1,k)}\right]^T$. We represent the input vector value $\boldsymbol{v}_j^{(\mathrm{BOW})} \in \mathbb{R}^m$ (i.e., the frequencies of the words that appear in document j) as

$$\boldsymbol{v}_j^{(\mathrm{BOW})} := [\boldsymbol{z}_j^{(1,1)^T}, \boldsymbol{z}_j^{(1,2)^T}, \cdots, \boldsymbol{z}_j^{(1,K)^T}]^T.$$

Second (concept) layer: We set the second-layer vector, $\boldsymbol{v}_j^{(\mathrm{CS})} \in \mathbb{R}^K$, as

$\boldsymbol{v}_j^{(\mathrm{CS})} := \tanh([\boldsymbol{z}_j^{(1,1)} \cdot \boldsymbol{w}_1^{(2)}, \cdots, \boldsymbol{z}_j^{(1,K)} \cdot \boldsymbol{w}_K^{(2)}]^T)$

where $\boldsymbol{w}_k^{(2)} \in \mathbb{R}^{n(k)}$ for each k. Let $w_{k,i}^{(2)}$ be the ith element of $\boldsymbol{w}_k^{(2)}$ and $v_{k,j}^{(\mathrm{CS})}$ be the kth element of $\boldsymbol{v}_j^{(\mathrm{CS})}$. If $w_{k,i}^{(2)}$ represents the sentiment score of word $w_{k,i}$, then $\boldsymbol{v}_j^{(\mathrm{CS})}$ represent the sentiment scores of concept cluster units.

Output layer: Let $\boldsymbol{W}^{(3)} \in \mathbb{R}^{K2 \times K}$ be the weight matrix between the second and third layers, $\boldsymbol{W}^{(4)} \in \mathbb{R}^{2 \times K2}$ be the weight matrix between the third and output layers, $\boldsymbol{w}_i^{(l)^T}$ and $w_{i,j}^{(l)}$ be the ith row and the (i,j) component of $\boldsymbol{W}^{(l)} (l = 3, 4)$, and $\boldsymbol{b}_0 \in \mathbb{R}^2$ be the bias vector. Here $K2$ is a scalar value. We represent the output layer value as

$\boldsymbol{y}_j = \mathrm{Softmax}(\boldsymbol{W}^{(4)} f_3(\boldsymbol{W}^{(3)} \boldsymbol{v}_j^{(\mathrm{CS})}) + \boldsymbol{b_0}), y_j^{(cls)} = \mathrm{argmax}\ \boldsymbol{y}_j,$

where $y_j^{(cls)} \in \{0\ (\text{negative}), 1\ (\text{positive})\}$ is the output layer value that corresponds to the predicted tag for the document j. We set f_3 to be tanh.

6.2.2 Initialization and learning of parameters After constructing the NN model, we can develop the GINN using the revised LEXIL including the following Lexical Initialization and *Update**. In the above, *Update**is the main different point from LEXIL and PLEXIL. We utilize *Update** due to the difference in structures between the GINN and BINNs.

Lexical Initialization After constructing the NN model, we initialize $w_{k,i}^{(2)}$ using a manually-created polarity dictionary. Let $PS(w_{k,i})$ be the sentiment score for $w_{k,i}$ given by the polarity dictionary. We set the initial value of $w_{k,i}^{(2)}$ as

$$w_{k,i}^{(2)} = \begin{cases} PS(w_{k,i}) & (w_{k,i} \text{ is included in the polarity dictionary}) \\ 0 & (\text{otherwise}) \end{cases}.$$

This initialization strategy realizes Idea 2, and we refer to this as *Init*.

Learning with *Update** We determine the parameter values not in $\{\boldsymbol{w}_k^{(2)}\}_{k=1}^K$ via the general backpropagation method with the softmax cross entropy as a loss function. However, we determine the values of $\{\boldsymbol{w}_k^{(2)}\}_{k=1}^K$ by updating $\{\boldsymbol{w}_k^{(2)}\}_{k=1}^K$ according to Algorithm 5 (called as *Update**) in each training iteration. This *Update** corresponds to the revised version of *Update* where *Update** is designed to address the GINN. In *Update**, using $\boldsymbol{H}^{*(j,t)}$ instead of $\boldsymbol{H}^{(j,t)}$ is specific and necessary for realizing the high *interpretability* of *GINN*s.

The values of $\boldsymbol{w}_k^{(2)}$ change during the learning process by the propagation of the sentiment scores from the dictionary (Fig. 13). After the learning stage is completed, we obtain the

Algorithm 5 Update strategy of $\{\boldsymbol{w}_k^{(2)}\}_{k=1}^K$ in the th training iteration ($Update^*$)

Input: $\{\boldsymbol{w}_k^{(2)}\}_{k=1}^K$, $\boldsymbol{W}^{(4)}$, $\boldsymbol{W}^{(3)}$, minibatch dataset in the tth training iteration Ω_m;

1: **for** $j \in \Omega_m$ **do**

2: $\boldsymbol{d_j} := \begin{cases} (0,1)^T & (j \text{ is positive}) \\ (1,0)^T & (j \text{ is negative}) \end{cases}$, $\boldsymbol{u}_j^{(2)} := \tanh^{-1}(\boldsymbol{v}_j^{(CS)})$, $\boldsymbol{u}_j^{(3)} := \boldsymbol{W}^{(3)}\boldsymbol{v}_j^{(CS)}$;

3: $\Delta_j^{(4)} := \boldsymbol{y}_j - \boldsymbol{d}_j$; $\boldsymbol{H}^{(j,t)} := \boldsymbol{W}^{(4)}\mathrm{diag}(f_3'(\boldsymbol{u}_j^{(3)}))\boldsymbol{W}^{(3)} (\in \mathbb{R}^{2\times K})$;

4: $\boldsymbol{H}^{*(j,t)} \in \mathbb{R}^{2\times K} \leftarrow$ zeros; Here, $(\boldsymbol{H}^{*(j,t)})_{l,k} = h^{*(j,t)}_{l,k}$ and $(\boldsymbol{H}^{(j,t)})_{l,k} = h^{(j,t)}_{l,k}$.

5: **for** $k \leftarrow 1$ to K **do**

6: **if** $h^{(j,t)}_{1,k} < 0$ **then** $h^{*(j,t)}_{1,k} \leftarrow h^{(j,t)}_{1,k}$; **if** $h^{(j,t)}_{2,k} > 0$ **then** $h^{*(j,t)}_{2,k} \leftarrow h^{(j,t)}_{2,k}$;

7: $\Delta_j^{(2)*} := (1 - \tanh^2(\boldsymbol{u}_j^{(2)})) \odot (\boldsymbol{H}^{*(j,t)^T}\Delta_j^{(4)})$,

8: **for** $k \leftarrow 1$ to K **do**

9: $\partial \boldsymbol{w}_k^{(2)*} := \frac{1}{N}\sum_{j\in\Omega_m} \Delta^{(2)*}_{k,j}\boldsymbol{z}_j^{(1,k)}$ where $\Delta^{(2)*}_{k,j}$ is the kth component of $\Delta_j^{(2)*}$;

10: Update $\boldsymbol{w}_k^{(2)}$ using $\partial \boldsymbol{w}_k^{(2)*}$ instead of using the gradient value of $\boldsymbol{w}_k^{(2)}$;

sentiment scores of unknown words by extracting the $\boldsymbol{w}_k^{(2)}$ values. The value of $w_{k,i}^{(2)}$ corresponds to the sentiment score of word $w_{k,i}$.

6.2.3 Proposed and baseline models We introduce two types of baseline models: base multilayer perceptron (MLP), plus MLP, and our proposed model, GINN. Their structures are constructed as discussed in Subsection 6.2.1, but they exhibit the following differences.

In **base MLP**, neither Init nor Update is used (i.e., developed by the general backpropagation method).

In **plus MLP**, Init is not used; however, Update is used.

In **GINN**, both Init and Update are used (i.e., developed by the II algorithm).

Let t^+ and t^- be positive values, $\Omega_{pw}^{(k,t^+)}$ (*positive word set*) be a set of words that satisfy $p^+(w_{k,i}) > t^+$ and whose cluster number is k, and $\Omega_{nw}^{(k,t^-)}$ (*negative word set*) be a set of words that satisfy $p^-(w_{k,i}) > t^-$ and whose cluster number is k. We can theoretically explain that the II algorithm develops GINNs whose *interpretability* and *market mood predictability* are both high in the ideal case: the II algorithm assigns the value of $w_{k,i}^{(2)}$ to a positive value if $w_{k,i} \in \Omega_{pw}^{(k,t^+)}$, and a negative value if $w_{k,i} \in \Omega_{nw}^{(k,t^+)}$ obtaining a local optimization solution in the ideal case (from Propositions 6.4.1–6.4.3 in Appendix 6.4).

6.3 Text visualization demonstration using real data

This section applies our text-visualization method for financial textual data. First, we evaluate our method in terms of interpretability, clustering interpretability, and market mood predictability (introduced in Subsection 6.1.2). Then, we present an example of text-visualization produced by GINN.

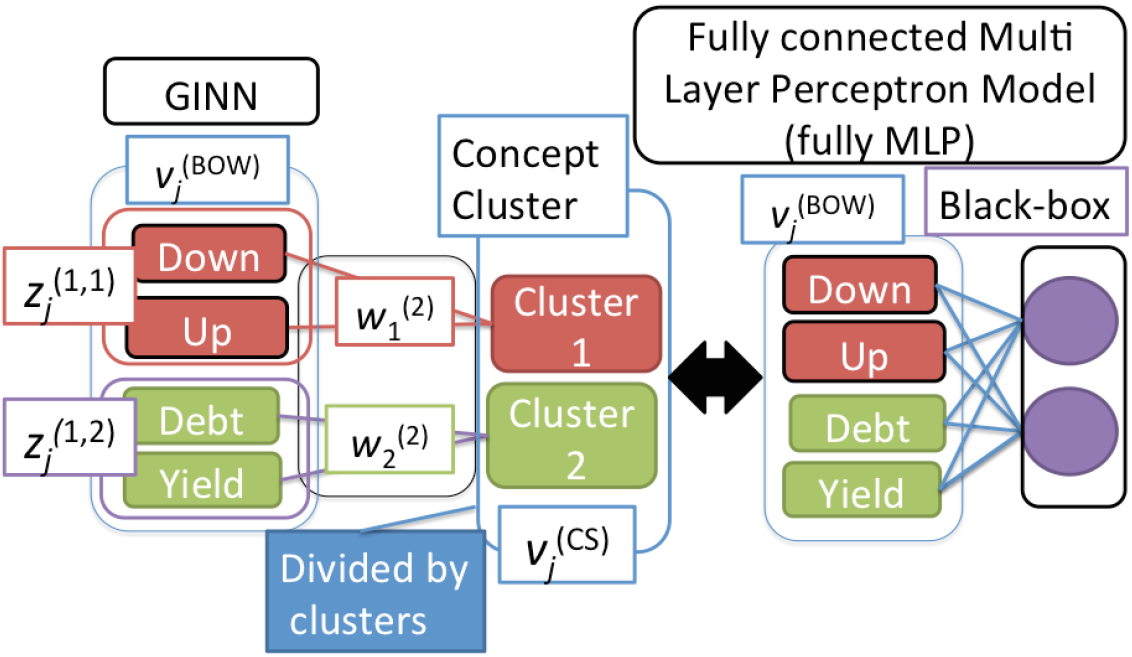

Figure 12: GINN vs MLP(fully)

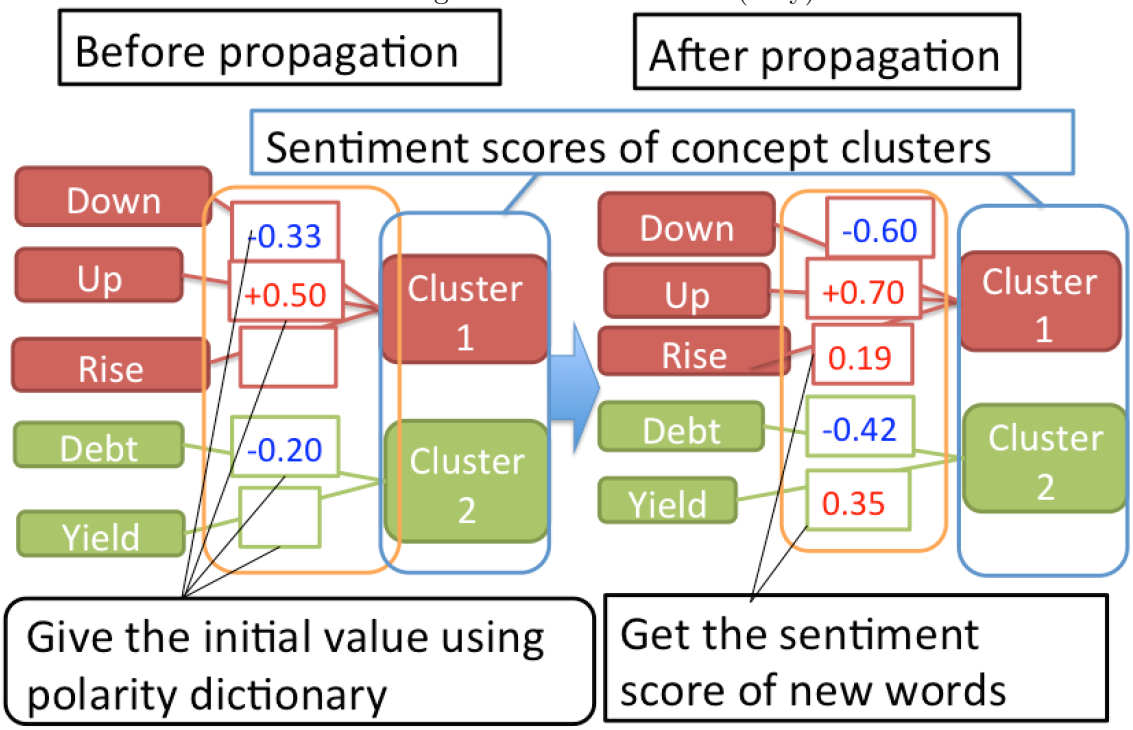

Figure 13: Polarity propagation process

6.3.1 Dataset and model development We used a dataset constructed from posts on the Yahoo! Finance Board[1] between September 1, 2015 and September 30, 2015 and their sentiment tags (i.e., *Yahoo! dataset*). We extracted all the posts tagged as negative (want to sell strongly) or positive (want to buy strongly), and sorted them in descending order by the date when they were posted. We then divided them into five equal parts while maintaining the order for a five fold cross-validation. After that, we prepared five train-validation and test dataset pairs by extracting each part in turn for use as the test dataset and using the remaining four parts as the train-validation dataset. We randomly extracted 10% of the train-validation data, taking equal percentages of samples from each class, for use as validation data. The remaining train-validation data were used as training data. The numbers of negative and positive posts were 15,887 and 50,843, respectively, and m was 28, 261.

Using each train-validation data, we developed the following five prediction models for the evaluations: SVM, fully connected MLP (fully MLP), base MLP, plus MLP and GINN. Here, fully, plus and base MLPs and GINN had four layers, and the kernel of SVM was linear. The hyper-parameters were determined using the validation data, and we used stratified sampling [69], the Adam optimizer [8], and Dropout [55]. The number of words that were included in the manually created polarity dictionary and used in the process of *Init* was 285.

Other Details for Experimental Setting We introduce experimental details in this section.

Preprocessing. First, each sentence was divided into morphemes (i.e., the smallest units of meaning) using the Japanese language morphological analysis system, Mecab [32]. We used the name lists from the Nikkei thesaurus[2], Wikipedia[3], Hatena[4] and Nico Nico Daihyakka[5] as user dictionaries. We extracted all the nouns, verbs, adjectives, and adverbs that appeared more than five times in the entire whole text corpus from the documents to calculate the feature vectors. There were 28, 261 vocabulary items.

Experimental settings and hyper-parameters We randomly extracted 10% of the training-validation data, taking equal percentages of samples from each class, for use as validation data. The hyper-parameters were then determined using this validation data.

Common settings for the fully MLP, base MLP, plus MLP and GINN Common settings for the MLPs GINN were as follows. In both experiments, we set the mini-batch size to 256, and the training epoch to 20 with early stopping. The dropout rates were 0.5, 0.2 or 0.0 and the second layer dimension was 100, 500 or 1000.

Settings for the fully MLP The settings for fully MLP were three layers, with the initial parameter value set according to $Norm(0, 0.01)$.

Settings for the SVM The linear SVM model settings were: class weight = "balanced," penalty= l2, and C $\in \{0.1, 1.0, 10\}$.

Settings for word2vec used with the MLPs (base and plus) and GINN The word2vec parameters were: size= 200, window= 5, min count= 5, and model = skipgram. The text corpuses used with word2vec consisted of all the posts on the Yahoo! Finance Board between September 1, 2015 and September 30, 2015. Morphological analysis was conducted using Mecab [32] to divide

[1] http://textream.yahoo.co.jp/category/1834773
[2] http://t21.nikkei.co.jp/public/help/contract/price/01/help_kiji_thes_field.html
[3] http://dumps.wikimedia.org/jawiki/latest/jawiki-latest-all-titles-in-ns0.gz
[4] http://d.hatena.ne.jp/images/keyword/keywordlist_furigana.csv
[5] http://www.nii.ac.jp/cscenter/idr/nico/nicopedia-apply.html

sentences to terms. We extracted all the nouns, verbs, adjectives, and adverbs that appeared more than five times in the entire whole text corpus. There were 57, 863 vocabulary items.

Handmade polarity dictionary The manually created polarity dictionary used for Init 1 was created by six financial professionals. Each professional assigned some important words polarity scores between {-2 (very negative), -1 (negative), 0 (neutral) , 1 (positive) , 2 (very positive)}. We used the mean values of these scores as the words' polarity scores. The number of words that were assigned non-zero was 285 in both experiments.

Experimental dataset Table 7 introduces the details about the datasets in Subsection 6.3.1. NP and NN are the numbers of documents tagged as positive and negative, respectively, in the dataset.

Table 7: Dataset details for the five-cross validation

ID	1	2	3	4	5
NN (training-validation)	12879	12802	12987	12228	12652
NP (training-validation)	40505	40582	40397	41156	40732
NN (test)	3008	3085	2900	3659	3235
NP (test)	10338	10261	10446	9687	10111

Cluster interpretability word list Base words and 100 words was used for the evaluation of cluster interpretability.

Interpretability evaluation We evaluated each model's interpretability by the following $Fw^{t^+,t^-}_{S^{ew},D}$ score.

Step 1: We set positive and negative word sets, $\Omega^{(k,t+)}_{pw}$ and $\Omega^{(k,t^-)}_{nw}$, for each k using a document dataset D according to Subsection 6.2.3. For each word $w \in S^{ew}$, we assign a positive (negative) label for the answer label if $w \in \cup^K_{k=1}\Omega^{(k,t^+)}_{pw}$ ($w \in \cup^K_{k=1}\Omega^{(k,t^-)}_{nw}$).

Step 2: We assigned a positive or negative label for the prediction label to each word $w \in S^{ew}$ using the prediction model. For the GINN and the plus and base MLPs, we assigned word $w_{k,i}$ a positive (negative) label if $w^{(2)}_{k,i} > 0$ ($w^{(2)}_{k,i} < 0$).

Step 3: We evaluated each method by the macro F_1 score for the answer and prediction labels (defined as $Fw_{S^{ew},D}(t^+, t^-)$). We set S^{ew}_D to be a set of words that appear more than ten times in D and were not included in the manually created polarity dictionary, t^+ to be the mean value of $\{p^+(w)|w \in S^{ew}_D\}$ and t^- to be $1 - t^+$. We evaluated methods in both the case where D was a training dataset and that where D was a union set of validation and test datasets (i. e., test-valid dataset). We compared the results of plus and base MLPs and GINN.

Result The first and second columns of Table 8 summarize the results. GINN shows significant improvement over baseline approaches: base and plus MLPs.

Discussion These results demonstrate that the II algorithm realized the high interpretability of the GINN as intended. To measure the limit value for interpretability, we also measured how much $Fw^{t^+,t^-}_{S^{ew},D}$ scores could be produced by other high-performance methods for assigning

Table 8: Fw scores are F_1 score results for interpretability: "train" and "test-valid" mean the case where D is a training dataset and that where D is a test-valid dataset, respectively. HF scores are F_1 score results for human interpretability.

Methods	Fw score		HF
	training dataset	test-valid dataset	
base MLP (baseline model)	0.488	0.493	0.465
plus MLP (baseline model)	0.516	0.506	0.484
GINN (proposed model)	**0.739**	**0.630**	**0.742**

sentiment scores to words: the gradient method with fully MLP and the SVM method. Such methods cannot achieve our goal because they cannot visualize concept cluster information. For the gradient method with fully MLP, we assigned each word $w \in S^{ew}$ a positive (negative) label if the input gradient value corresponding to the word w calculated by the gradient method [15] and the fully MLP model was positive (negative) (See the supplementary material[2] for the details). For the SVM method, we assigned each word $w \in S^{ew}$ a positive (negative) label if the support vector value corresponding to word w was positive (negative). The $Fw^{t^+,t^-}_{S^{ew},D}$ scores in the case where D was a training dataset and that where D was a valid-test dataset were 0.704 and 0.604, respectively, for the SVM method and 0.753 and 0.620, respectively, for the gradient method with fully MLP. These results show that GINN was able to produce more satisfactory results than other methods when D was a valid-test dataset, demonstrating the high interpretability of GINN.

Human interpretability evaluation (additional evaluation): We also evaluated word sentiment scores given by GINN in terms of whether they fit peoples' feelings. We randomly extracted 100 posts tagged as negative and positive from the test dataset. Three individual investors then manually extracted important words for the sentiment decision from each post and tagged them as positive or negative. We evaluated the models by their ability to accurately assign sentiment tags to these words in the same way as Step 3. We used the mean F_1 score for the three investors as the evaluate base (i.e., HF score). The right column of Table 8 summarizes the result, showing that GINN had more satisfactory results than the base and plus MLPs. Moreover, HF scores for the SVM method and the gradient method with fully MLP were 0.753 and 0.759, respectively, close to the HF score of GINN. Thus, we consider that sentiment scores given to terms by GINN sufficiently fit peoples' feelings.

6.3.2 Clustering interpretability evaluation We briefly checked the validity of word clustering in the II algorithm. After deciding the cluster number K as 1000 and clustering words appeared in the Yahoo! dataset using the spherical K-means method [26], we randomly extracted six clusters and 100 words in total from these six clusters. We then randomly selected one word that was not included in the extracted 100 words from each cluster in the six clusters as a base word (total six words). Two individual investors then manually reclustered the 100 words into six clusters by deciding the closest word to each word in these words from six base words. We evaluated the clustering result by measuring the proximity of the manually clustered result to the clustering result that uses the the spherical K-means method [26] in terms of macro F_1 score.

The mean F_1 score between investors was $\mathbf{0.93(>> 0.16)}$. From this result, we consider that the word clustering by our approach sufficiently fits peoples' feelings, and clustering interpretability is sufficiently high.

6.3.3 Market mood predictability evaluation We evaluate the market mood predictability by whether each model can accurately predict sentiment tags of documents in the test dataset in terms of the mean F_1 scores for the five-fold cross-validation, and compare the results between the following methods: SVM, fully, base and plus MLPs, and GINN. The F_1 scores were 0.733, 0.737, 0.692, 0.681 and **0.743** for SVM, fully MLP, base MLP, plus MLP and **GINN**, respectively. These results show that GINN produced the more satisfactory result than the others in market mood predictability.

6.3.4 Text Visualization From the evaluations for interpretability, clustering interpretability and market mood predictability of GINN, we can demonstrate both the validity for visualization by GINN and the improvement by the II algorithm.

We then present a text-visualization example produced by the GINN. We call this type of text visualization framework as Financial Text Visualizer.

Text visualization Process We visualized an input post in the following step. We colored each word $w_{k,i}$ in a post as blue if $w_{k,i}^{(2)} < -0.05$ and red if $0.05 < w_{k,i}^{(2)}$, and displayed concept cluster information of words appeared in a post by displaying some words included in the same clusters. We then extracted the four most important concept clusters for the decision by Algorithm 6, and printed the cluster numbers after the terms included in these clusters.

Algorithm 6 Extract the important clusters for the sentiment analysis

Input: document j, the second and output layer unit values, $\boldsymbol{v}_j^{(CS)}$ and $\boldsymbol{y}_j$

1: $loss \leftarrow \max \boldsymbol{y}_j$, $\boldsymbol{H}_{grad}^{(2)} \leftarrow \frac{\partial loss}{\partial \boldsymbol{v}_j^{(CS)}} \odot \boldsymbol{v}_j^{(CS)}$ (by the gradient method [15]);

2: $I_{grad}^{(3)} \leftarrow$ sorted indices in ascending order by the values of $\boldsymbol{H}_{grad}^{(2)}$;

3: **return** the first four indices of $I_{grad}^{(3)}$;

Text Visualization Result Fig. 14 shows a text-visualization example of a document in the test dataset using the GINN ($K = 1000, K2 = 100$). By visualizing documents as above, we can quickly capture in what sense each word in a document is positive or negative and how the prediction was made. As shown in the text-visualization by the developed GINN, this GINN should be helpful in a situation where non-experts want to understand the specialized documents briefly.

6.4 Conclusion

This chapter introduces a gradient interpretable neural network (GINN) as an example of the application of LEXIL (proposed in Chapter 3) into the development of interpretable NNs. It should be noted that original LEXIL and PLEXIL (proposed in Chapter 4) can not be used in this case because the structure of the GINN is a little different from BINNs. To address this

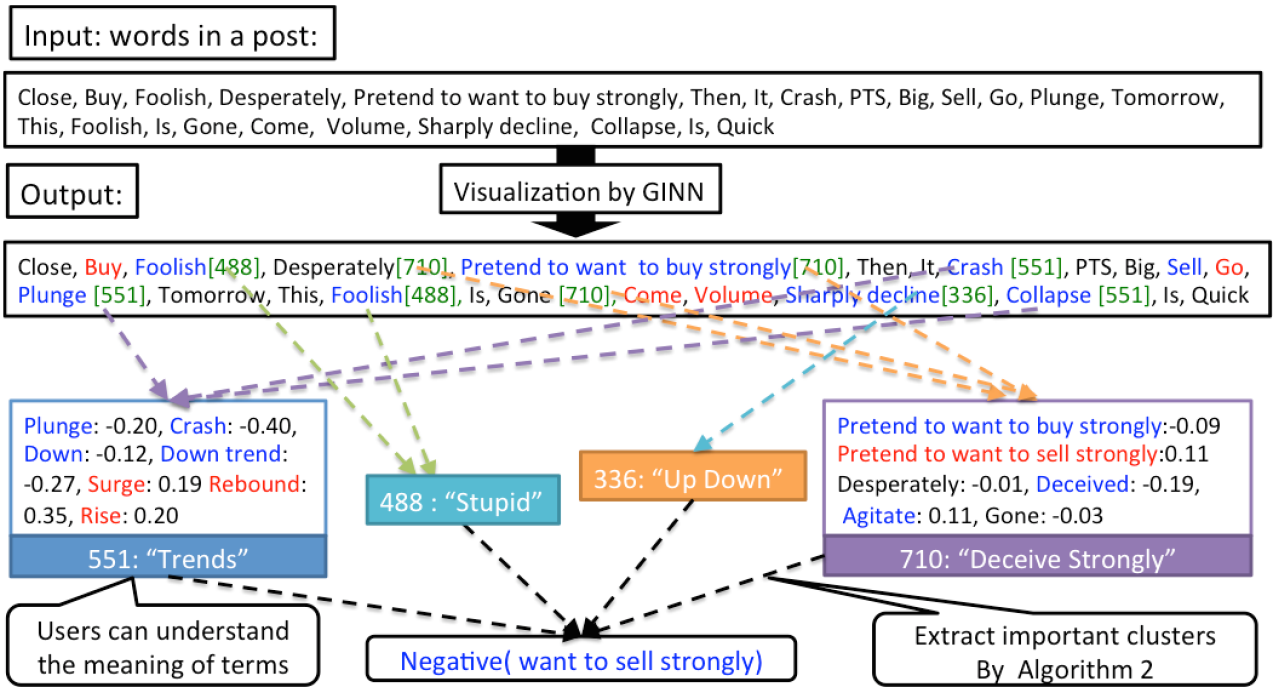

Figure 14: Text-visualization examples from GINN and Yahoo finance board posts. The numbers in green that follow some words are their cluster numbers, and these numbers are the results of the extraction of the most four important concept clusters in Algorithm 6. This post was originally in Japanese and we manually replaced each Japanese word to the corresponding English word for this study [21].

problem, we propose a novel development strategy for GINN called II algorithm. In II algorithm, we utilize Lexical Initialization and $Update^*$, and this $Update^*$ is the revision point from PLEXIL. We experimentally demonstrate the high prediction ability and high interpretability of GINN using a real financial dataset.

This success of the development of the GINN demonstrates that LEXIL can be utilized in a flexible way and the proposed basic learning strategy can be utilized to many cases by appropriate conversions.

Appendix: Theoretical analysis of the II algorithm

Let $\Omega_{dw}^{(k)}$ be a set of words included in the kth cluster and included in the polarity dictionary, $D^{(p)}$ and $D^{(n)}$ be the positive and negative document sets, $\partial w_{k,i}^{(2)*}$ be the ith component of $\partial \boldsymbol{w}_k^{(2)*}$, $p^-(w_{k,i})$ be $p\left(j \in D^{(n)} | z_{j,i}^{(1,k)} > 0\right)$, $p^+(w_{k,i})$ be $1 - p^-(w_{k,i})$, and $\partial \boldsymbol{H}^{(j,t)}$ be the gradient value of $\boldsymbol{H}^{(j,t)}$ in *Update*. Then,

PROPOSITION 6.4.1. *If we utilize* Update *for the parameter updates, then,*

$$\left\{ \begin{array}{ll} E[\partial w_{k,i}^{(2)*}] < 0 & \left(\frac{p^+(w_{k,i})}{p^-(w_{k,i})} > \frac{E[|\Delta_{j,k}^{(2)*}||z_{j,i}^{(1,k)}=1 \cap j \in D^{(n)}]}{E[|\Delta_{j,k}^{(2)*}||z_{j,i}^{(1,k)}=1 \cap j \in D^{(p)}]} \right) \\ E[\partial w_{k,i}^{(2)*}] > 0 & \left(\frac{p^+(w_{k,i})}{p^-(w_{k,i})} < \frac{E[|\Delta_{j,k}^{(2)*}||z_{j,i}^{(1,k)}=1 \cap j \in D^{(n)}]}{E[|\Delta_{j,k}^{(2)*}||z_{j,i}^{(1,k)}=1 \cap j \in D^{(p)}]} \right) \end{array} \right. . \tag{34}$$

is established. Proposition 6.4.1 indicates that if **Cond 1:** *the values of t^+ and t^- are sufficiently large*, and **Cond 2:** *for every word $w_{k,i^+} \in \Omega_{dw}^{(k)} \cap \Omega_{pw}^{(k)}$, and $w_{k,i^-} \in \Omega_{dw}^{(k)} \cap \Omega_{nw}^{(k)}$, the initial values of $w_{k,i^+}^{(2)}$ and $w_{k,i^-}^{(2)}$ given by* Init *are positive and sufficiently large, and negative and sufficiently small, respectively*, are met for every k, then, the II algorithm is expected to award each positive word $\in \Omega_{pw}^{(k)}$ (negative word $\in \Omega_{nw}^{(k)}$) a positive (negative) sentiment score.

Let $\boldsymbol{H}^{d(j,t)}$ be $\boldsymbol{H}^{(j,t)} - \boldsymbol{H}^{*(j,t)}$. Then, the following propositions important for explaining the *market mood predictability* of GINN are established.

PROPOSITION 6.4.2. *If the initial values of $|\boldsymbol{W}^{(3)}|$ and $|\boldsymbol{W}^{(4)}|$ are sufficiently small (***Cond 3***) and for every $j \in \Omega_m^{(t)}$, the values of $\boldsymbol{z}_j^{(2)}$ are $\left\{ \begin{array}{ll} positive & (j \in D^{(p)}) \\ negative & (j \in D^{(n)}) \end{array} \right.$, then, the first and second row vector values of $\partial \boldsymbol{H}^{(j,t)}$ are positive and negative respectively, and $\frac{\sum_{j \in \Omega_m^{(t+1)}} \|\boldsymbol{H}^{d(j,t+1)}\|_1}{\sum_{j \in \Omega_m^{(t+1)}} \|\boldsymbol{H}^{(j,t+1)}\|_1} \leq \frac{\sum_{j \in \Omega_m^{(t+1)}} \|\boldsymbol{H}^{d(j,t)}\|_1}{\sum_{j \in \Omega_m^{(t+1)}} \|\boldsymbol{H}^{(j,t)}\|_1}$.*

PROPOSITION 6.4.3. *If, for every k,* **Cond** *1–3 are established, the values $|\Omega_{pw}^{(k,t^+)}|$, $|\Omega_{nw}^{(k,t^-)}|$ and $|\Omega_m|$ are sufficiently large, then, $\lim_{t \to \infty} \frac{\sum_{j \in \Omega_m^{(t)}} \|\boldsymbol{H}^{d(j,t)}\|_1}{\sum_{j \in \Omega_m^{(t)}} \|\boldsymbol{H}^{(j,t)}\|_1} = 0$.*

See the supplementary material[2] for the proofs and the details.

Proof of Proposition 6.4.1

Proof. Here, for every $k(\leq K)$,

$$\left\{ \begin{array}{ll} \Delta_{k,j}^{(2)*} \leq 0 & (j \in D^{(p)}) \\ \Delta_{k,j}^{(2)*} \geq 0 & (j \in D^{(n)}) \end{array} \right. .$$

Thus,

$$E[\partial w_{k,i}^{(2)*}] = E\left[\frac{1}{|\Omega_m|} \sum_{j \in \Omega_m} \Delta_{k,j}^{(2)*} \boldsymbol{z}_j^{(1,k)} \right]$$

$$= \mathrm{freq}(w_{k,i}) \left(p^{-}(w_{k,i}) E\left[\Delta_{k,j}^{(2)*} \,|\, j \in D^{(p)} \right] - p^{+}(w_{k,i}) E\left[\Delta_{j,k}^{(2)*} \,|\, j \in D^{(n)} \right] \right)$$

Therefore, Proposition 6.4.1 can be established. □

Proof of Proposition 6.4.2

Let us denote $\boldsymbol{Z}^{(2)} := [\boldsymbol{v}_{m(1)}^{(CS)}, \boldsymbol{v}_{m(2)}^{(CS)}, \cdots, \boldsymbol{v}_{m(N)}^{(CS)}] (\in \mathbb{R}^{K \times N})$, $\boldsymbol{U}^{(2)} := \tanh^{-1}(\boldsymbol{Z}^{(2)})$, $\boldsymbol{U}^{(3)} := \boldsymbol{W}^{(3)} \boldsymbol{Z}^{(2)}$, $\boldsymbol{u}_j^{(l)}$ is the jth column of $\boldsymbol{U}^{(l)}$ $(l = 2, 3)$, and $\boldsymbol{z}_j^{(l)}$ and $z_{i,j}^{(l)}$ are the jth column and the (i, j) component $\boldsymbol{Z}^{(l)}$ $(l = 2)$.

Proof. We approximate $\partial \boldsymbol{H}^{(j,t)}$ as follows.

$$\partial \boldsymbol{H}^{(j,t)} = \partial(\boldsymbol{W}^{(4)}(\mathrm{diag}(f_3'(\boldsymbol{u}_j^{(3)}))\boldsymbol{W}^{(3)}))$$

$$\approx \partial(\boldsymbol{W}^{(4)})\mathrm{diag}(f_3'(\boldsymbol{u}_j^{(3)}))\boldsymbol{W}^{(3)} + \boldsymbol{W}^{(4)}\mathrm{diag}(f_3'(\boldsymbol{u}_j^{(3)}))\partial \boldsymbol{W}^{(3)}$$

First, we confirm that If for every $j \in \Omega_m^{(t)}$, the values of $\boldsymbol{z}_j^{(2)}$ are $\begin{cases} positive & (j \in D^{(p)}) \\ negative & (j \in D^{(n)}) \end{cases}$, then, following three lemmas are established.

Lemma 6.4.1. *The first and second row vector values of $\Delta_j^{(4)} \boldsymbol{z}_j^{(2)^T}$ are positive and negative, respectively.*

Lemma 6.4.2. *The first and second rows of*

$$\partial(\boldsymbol{W}^{(4)})\mathrm{diag}(f_3'(\boldsymbol{u}_j^{(3)}))\boldsymbol{W}^{(3)}$$

are positive and negative, respectively.

Lemma 6.4.3. *The first and second rows of*

$$\boldsymbol{W}^{(4)}\mathrm{diag}(f_3'(\boldsymbol{u}_j^{(3)}))\partial \boldsymbol{W}^{(3)}$$

are positive and negative, respectively.

Proof of Lemma 6.4.1

Proof. From the condition,

$$z_{k,j}^{(2)} \begin{cases} > 0 & (j \in D^{(p)}) \\ < 0 & (j \in D^{(n)}). \end{cases} \tag{35}$$

Moreover, from the following Eq (36),

$$\begin{cases} \boldsymbol{d_j} = (0, 1)^T & (j \in D^{(p)}), \\ \boldsymbol{d_j} = (1, 0)^T & (j \in D^{(n)}) \end{cases} \tag{36}$$

$$\Delta_j^{(4)} := \boldsymbol{y}_j - \boldsymbol{d}_j \begin{cases} (|\Delta_{1,j}^{(4)}|, -|\Delta_{1,j}^{(4)}|)^T & (j \in D^{(p)}) \\ (-|\Delta_{1,j}^{(4)}|, |\Delta_{1,j}^{(4)}|)^T & (j \in D^{(n)}). \end{cases} \tag{37}$$

Thus, from Eq (35) and Eq (37), Lemma 6.4.1 is established. □

Proof of Lemma 6.4.2

Proof.

$$\boldsymbol{W}^{(4)}\mathrm{diag}(f_3'(\boldsymbol{u}_j^{(3)}))\partial\boldsymbol{W}^{(3)} = \frac{1}{N}\boldsymbol{W}^{(4)}\mathrm{diag}(f_3'(\boldsymbol{u}_j^{(3)}))\Delta^{(3)}{\boldsymbol{Z}^{(2)}}^T$$
$$= \frac{1}{N}\boldsymbol{W}^{(4)}\mathrm{diag}(f_3'(\boldsymbol{u}_j^{(3)}))\sum_i(f_3'(\boldsymbol{u}_i^{(3)})\odot({\boldsymbol{W}^{(4)}}^T\Delta^{(4)}))\boldsymbol{z}_i^{(2)T}$$
$$= \frac{1}{N}\sum_{i=1}^{N}\boldsymbol{W}^{(4)}\mathrm{diag}(f_3'(\boldsymbol{u}_j^{(3)}))\mathrm{diag}(f_3'(\boldsymbol{u}_i^{(3)}))\boldsymbol{W}^{(4)T}\Delta_i^{(4)}\boldsymbol{z}_i^{(2)T}$$

Here,

$$\partial\boldsymbol{w}_1^{(4)} = -\partial\boldsymbol{w}_2^{(4)},$$

because $\partial\boldsymbol{W}^{(4)} = \Delta^{(4)}{\boldsymbol{Z}^{(3)}}^T$ and $\Delta_{1,j}^{(4)} = -\Delta_{2,j}^{(4)}$ for every j. Considering that $\boldsymbol{W}^{(4)}$ is the sum of the values of $\partial\boldsymbol{W}^{(4)}$ in the previous updates, if the initial value of $|\boldsymbol{W}^{(4)}|$ is sufficiently small, then, we can approximate as

$$\boldsymbol{w}_1^{(4)} \approx -\boldsymbol{w}_2^{(4)}. \tag{38}$$

let us denote A^l as follows.

$$A^l := \boldsymbol{W}^{(4)}\mathrm{diag}(f_3'(\boldsymbol{u}_j^{(3)}))\mathrm{diag}(f_3'(\boldsymbol{u}_i^{(3)}))\boldsymbol{W}^{(4)T}.$$

We define $v_{1,i}^{\Delta^{(4)}\boldsymbol{Z}^{(3)}}$ as the ith component of $\boldsymbol{w}_1^{(4)}$, and $F_{i,j}$ as the (i,i) component of $\mathrm{diag}(f_3'(\boldsymbol{u}_j^{(3)}))\mathrm{diag}(f_3'(\boldsymbol{u}_i^{(3)}))$. Then,

$$A^l = \begin{pmatrix} \sum_{i=1}^{K2}F_{i,j}|v_{1,i}^{\Delta^{(4)}\boldsymbol{Z}^{(3)}}|^2 & -\sum_{i=1}^{K2}F_{i,j}|v_{1,i}^{\Delta^{(4)}\boldsymbol{Z}^{(3)}}|^2 \\ -\sum_{i=1}^{K2}F_{i,j}|v_{1,i}^{\Delta^{(4)}\boldsymbol{Z}^{(3)}}|^2 & \sum_{i=1}^{K2}F_{i,j}|v_{1,i}^{\Delta^{(4)}\boldsymbol{Z}^{(3)}}|^2 \end{pmatrix}$$

Thus, from Lemma 6.4.1, if the initial value of $|\boldsymbol{W}^{(4)}|$ is sufficiently small, then, the first and second row vector values of $A^l\Delta_i^{(4)}{\boldsymbol{z}_i^{(2)}}^T$ are positive and negative, respectively. □

Proof of Lemma 6.4.3

Proof.

$$\partial(\boldsymbol{W}^{(4)})\mathrm{diag}(f_3'(\boldsymbol{u}_j^{(3)}))\boldsymbol{W}^{(3)} = \frac{1}{N}\Delta^{(4)}f_3({\boldsymbol{Z}^{(2)}}^T{\boldsymbol{W}^{(3)}}^T)\mathrm{diag}(f_3'(\boldsymbol{u}_j^{(3)}))\boldsymbol{W}^{(3)}$$
$$= \frac{1}{N}\sum_{i=1}^{N}\Delta_i^{(4)}f_3({\boldsymbol{z}_i^{(2)}}^T{\boldsymbol{W}^{(3)}}^T)\mathrm{diag}(f_3'(\boldsymbol{u}_j^{(3)}))\boldsymbol{W}^{(3)}$$
$$= \frac{1}{N}\sum_{i=1}^{N}\Delta_i^{(4)}{\boldsymbol{z}_i^{(2)}}^T{\boldsymbol{W}^{(3)}}^T\boldsymbol{M}^i\mathrm{diag}(f_3'(\boldsymbol{u}_j^{(3)}))\boldsymbol{W}^{(3)}$$

Let us define the matrix $\boldsymbol{M}^i$as follows.

$$\boldsymbol{M}^i := \mathrm{diag}\left(\frac{f_3(u_i)}{u_i}\right)$$

We define A^r as follows.

$$\boldsymbol{A}^r := {\boldsymbol{W}^{(3)}}^T \boldsymbol{M}^i \mathrm{diag}(f_3'(\boldsymbol{u}_j^{(3)})) \boldsymbol{W}^{(3)}$$

Here,

$$\partial \boldsymbol{W}^{(3)} = \frac{1}{N} \sum_i \mathrm{diag}(f_3'(\boldsymbol{u}_i^{(3)})) {\boldsymbol{W}^{(4)}}^T \Delta_i^{(4)} {\boldsymbol{z}_i^{(2)}}^T$$

Thus,

$$\partial {\boldsymbol{W}^{(3)}}^T \boldsymbol{M}^i \mathrm{diag}(f_3'(\boldsymbol{u}_j^{(3)})) \partial \boldsymbol{W}^{(3)}$$

$$= \frac{1}{N^2} \sum_{l \in \Omega_m} \sum_{m \in \Omega_m} \boldsymbol{z}_l^{(2)} {\Delta_l^{(4)}}^T \boldsymbol{W}^{(4)} D_{i,j,l,m}^r {\boldsymbol{W}^{(4)}}^T \Delta_m^{(4)} {\boldsymbol{z}_m^{(2)}}^T$$

where we denote

$$\mathrm{diag}(f_3'(\boldsymbol{u}_l^{(3)})) \boldsymbol{M}^i \mathrm{diag}(f_3'(\boldsymbol{u}_j^{(3)})) \mathrm{diag}(f_3'(\boldsymbol{u}_m^{(3)}))$$

as $D_{i,j,l,m}^r$. Considering that $\boldsymbol{w}_1^{(4)} \approx -\boldsymbol{w}_2^{(4)}$ (Eq (38)),

$$\boldsymbol{W}^{(4)} D_{i,j,l,m}^r {\boldsymbol{W}^{(4)}}^T \approx \begin{pmatrix} k^{(4)} & -k^{(4)} \\ -k^{(4)} & k^{(4)} \end{pmatrix}$$

where $k^{(4)} := \boldsymbol{w}_1^{(4)} D_{i,j,l,m}^r {\boldsymbol{w}_1^{(4)}}^T > 0$ because $D_{i,j,l,m}^r$ is the diagonal matrix, and each diagonal element value of $D_{i,j,l,m}^r$ is positive. Moreover, from Eq (37),

$${\Delta_l^{(4)}}^T \begin{pmatrix} k^{(4)} & -k^{(4)} \\ -k^{(4)} & k^{(4)} \end{pmatrix} \Delta_m^{(4)} \begin{cases} > 0 & (\boldsymbol{d}_l = \boldsymbol{d}_m) \\ < 0 & (\boldsymbol{d}_l \neq \boldsymbol{d}_m) \end{cases}$$

Therefore, from Eq (35), each element value of

$$\boldsymbol{z}_l^{(2)} {\Delta_l^{(4)}}^T \begin{pmatrix} k^{(4)} & -k^{(4)} \\ -k^{(4)} & k^{(4)} \end{pmatrix} \Delta_m^{(4)} {\boldsymbol{z}_m^{(2)}}^T$$

is positive.

Thus, each element value of

$$\partial {\boldsymbol{W}^{(3)}}^T \boldsymbol{M}^i \mathrm{diag}(f_3'(\boldsymbol{u}_j^{(3)})) \partial \boldsymbol{W}^{(3)}$$

is positive. Considering that $\boldsymbol{W}^{(3)}$ is the sum of the values of $\partial \boldsymbol{W}^{(3)}$ in the previous updates, if the initial value of $\boldsymbol{W}^{(3)}$ is sufficiently small and N is sufficiently large then, each element value of $\boldsymbol{A}^r$ is positive. Thus, from Lemma 6.4.1 and the above, the first array vector values of $\Delta_i^{(4)} {\boldsymbol{z}_i^{(2)}}^T A^r$ are positive, and the second array vector values are negative. Thus, if N is sufficiently large, then, Lemma 6.4.3 is established.

□

Summarization Considering

$$\partial \boldsymbol{H}^{(j,t)} = \frac{1}{N} \sum_{i=1}^{N} A^l \Delta_i^{(4)} {\boldsymbol{z}_i^{(2)}}^T + \Delta_i^{(4)} {\boldsymbol{z}_i^{(2)}}^T A^r,$$

and Lemmas 6.4.2 and 6.4.3, the first and second row vector values of $E[\partial \boldsymbol{H}^{(j,t)}]$ are positive and negative for every j. Thus, Proposition 6.4.3 is established. □

Explanation of Proposition 6.4.3 Proposition 6.4.3 can be explained as follows.

Proof. If the following conditions are met for every k.

Cond 1 the values of t^+ and t^- are sufficiently large,

Cond 2 for every word $w_{k,i^+} \in \Omega_{dw}^{(k)} \cap \Omega_{pw}^{(k)}$, and $w_{k,i^-} \in \Omega_{dw}^{(k)} \cap \Omega_{nw}^{(k)}$, the initial value of $w_{k,i^+}^{(2)}$ given by **Init** is positive and sufficiently large, and negative and sufficiently small, respectively,

Cond 3 the initial values of $|\boldsymbol{W}^{(3)}|$ and $|\boldsymbol{W}^{(4)}|$ are sufficiently small, and

Cond 4 the values $|\Omega_{pw}^{(k,t^+)}|$, $|\Omega_{nw}^{(k,t^-)}|$ and $|\Omega_m|$ are sufficiently large,

then, from **Cond 1**, **Cond 2**, **Cond 4** and Proposition 6.4.1, Eq (35) is established. Thus, from Proposition 6.4.2 and **Cond 3**, Proposition 6.4.3 is established. □

Propositions 6.4.2 and 6.4.3 indicates that we can obtain the local optimal solution using the II algorithm in the ideal case because the influence of **Update** gradually disappears over time. Moreover, from these propositions, we can see that **Init** maintains the model predictability because **Init** is useful for satisfying Cond 2.

From Proposition 6.4.1, we can explain the interpretability of GINN, and from Propositions 6.4.2 and 6.4.3, we can confirm the predictability in the ideal case.

Gradient method for assigning terms their polarity scores using fully MLP

We introduce the method for assigning sentiment scores to words using the gradient method [15] and fully MLP. We consider a fully connected MLP model $f^{MLP} : \mathbb{R}^m \to \mathbb{R}^2$ as follows. When the input value is

$$\boldsymbol{v}_j^{(\mathrm{BOW})} = [\boldsymbol{z}_j^{(1,1)^T}, \boldsymbol{z}_j^{(1,2)^T}, \cdots, \boldsymbol{z}_j^{(1,K)^T}]^T,$$

the output value $\boldsymbol{y}_j^{mlp}$ can be represented as

$$\boldsymbol{y}_j^{mlp} = f^{MLP}(\boldsymbol{v}_j^{(\mathrm{BOW})})$$

, and the model predicts the sentiment tag of a document j as

$$\begin{cases} \text{negative} & (\text{argmax } \boldsymbol{y}_j^{mlp} = 1) \\ \text{positive} & (\text{argmax } \boldsymbol{y}_j^{mlp} = 2). \end{cases}$$

Let us denote a document set in the training dataset as D_{train}. We calculate the gradient sentiment value of word $w_{k,i}$, $Gr(w_{k,i})$, using the backpropagation method as follows.

$$\boldsymbol{y}_j^{mlp+} := \boldsymbol{y}_j^{mlp} \odot (1,0)^T, \boldsymbol{y}_j^{mlp-} := \boldsymbol{y}_j^{mlp} \odot (0,1)^T,$$

$$Gr(w_{k,i}) := \frac{\sum_{j \in D_{train}} \frac{\partial \boldsymbol{y}_j^{mlp+}}{\partial z_{j,i}^{(1,k)}} - \frac{\partial \boldsymbol{y}_j^{mlp-}}{\partial z_{j,i}^{(1,k)}}}{|D_{train}|}.$$

Chapter 7

Contextual Sentiment Neural Network (CSNN)

This chapter introduces a contextual sentiment neural network (CSNN) [22] as an example of the revised usage of LEXIL and JSP learning (proposed in Chapter 3). It should be noted that original PLEXIL and PJSP learning (proposed in Chapter 4) can not be used in this case because the structure of the CSNN is more complex than BINNs. Therefore, to develop CSNN, we convert PLEXIL to the revised form that can be utilized in developing the CSNN. From this example, we can see that the idea behind the proposition of LEXIL can be utilized in a more complicated NNs than BINN. This means that the proposed basic learning strategy has the potential for developing many types of interpretable NNs by its appropriate conversion.

We first introduce the CSNN briefly in Section 7.1 and then explain the CSNN structure and the learning strategy of CSNN in Section 7.2. In this learning strategy, we convert PLEXIL and PJSP learning into the appropriate learning strategy for developing the CSNN. We then experimentally evaluate our approach using real textual datasets in Section 7.4 and then conclude this chapter in Section 7.5.

7.1 Overview

As a specific example of BINN, this chapter considers the CSNN [22]. This CSNN has the following six interpretable layers: word-level original sentiment layer (WOSL), sentiment shift layer (SSL), word-level local contextual sentiment layer (WLCSL), global important point layer (GIL), word-level global contextual sentiment layer (WGCSL), and concept-level contextua sentiment layer (CCSL) as shown in Fig. 15. The WOSL, WLCSL, and WGCSL represent the word-level original, local contextual, and global contextual sentiments of each term in a review, respectively. The CCSL represtns the concept-level contextual sentiment of each concept cluster. The SSL indicates whether a sentiment of each term in a review is shifted or not (i.e., local word-level context), and GIL indicates the global word-level context in a review. WOSL is represented in a dictionary manner. SSL and GIL are represented using long short-term memories (LSTM) cells [50]. The values of WLCSL and WGCSL are represented by multiplying the values of WOSL and SSL, and by multiplying the values of WLCSL and GIL, respectively.

Therefore, CSNN is valuable in a case where the extraction of the word-level original sentiment, word-level local context, word-level local contextual sentiment, word-level global

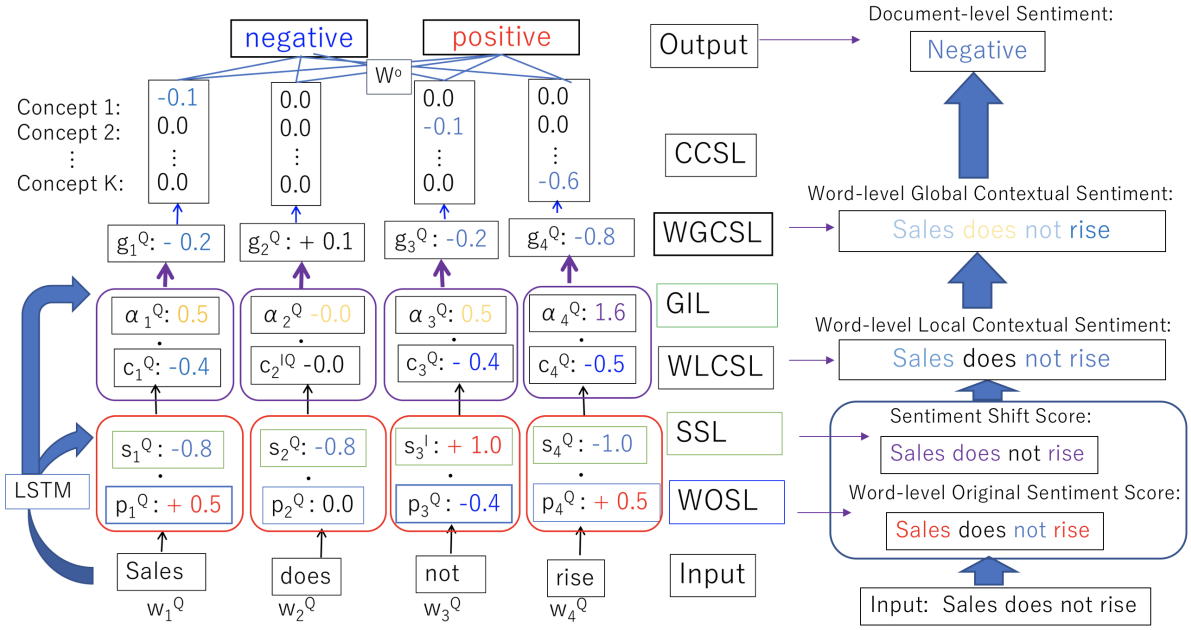

Figure 15: CSNN

context, word-level global contextual sentiment, and concept-level sentiment are required in the explanation of the document-level analysis result, as shown in Fig. 16.

7.2 Structure of CSNN

This section introduces the detailed structure of CSNN.

Notation. We first define several symbols. Let $\{(\mathbf{Q}_i, d^{\mathbf{Q}_i})\}_{i=1}^{N}$ be a training dataset where N is the training data size, $\mathbf{Q}_i = \{w_t^{\mathbf{Q}_i}\}_{t=1}^{n}$ is a review, $d^{\mathbf{Q}_i}$ is its sentiment tag (1 is positive and 0 is negative), and $w_t^{\mathbf{Q}_i}$ is a tth term in $\mathbf{Q}_i$. Let $\{w_i\}_{i=1}^{v}$ represent the terms that appear in a text corpus, and v be the vocabulary size. We define the vocabulary index of word w_i as $I(w_i)$. Therefore, $I(w_i) = i$. Let $\boldsymbol{w}_i^{em} \in \mathbb{R}^e$ be an embedding representation of word w_i, and the embedding matrix $\boldsymbol{W}^{em} \in \mathbb{R}^{v \times e}$ be $[\boldsymbol{w}_1^{emT}, \cdots, \boldsymbol{w}_v^{emT}]^T$. where e is the dimension size of word embedding and $\|\boldsymbol{w}_i^{em}\|_2 = 1$. $\boldsymbol{W}^{em}$ is the constant value obtained using the skip-gram method [39] and each text corpus in a training dataset.

WOSL This layer represents word-level original sentiment representations $\{p_t^{\mathbf{Q}}\}_{t=1}^{n}$ as

$$p_t^{\mathbf{Q}} = w^p_{I(w_t^{\mathbf{Q}})}$$

where $\boldsymbol{W}^p \in \mathbb{R}^v$ represents the original sentiment scores of words, and w_i^p is the $i-$th element of $\boldsymbol{W}^p$. The w_i^p value corresponds to the original sentiment score of the word w_i.

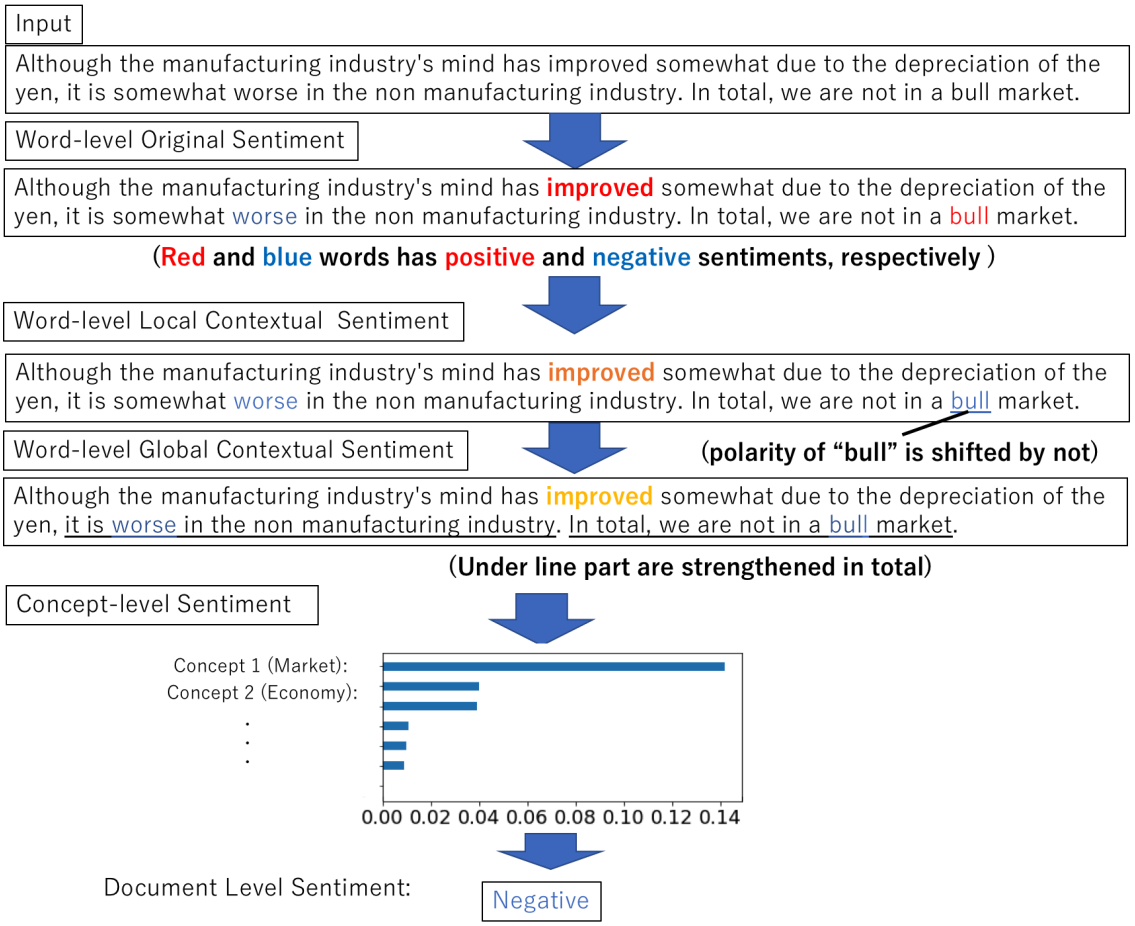

Figure 16: Goal: development of neural network (NN) that can explain its prediction results using five types of sentiments

SSL This layer represents the word-level sentiment shift scores $\{s_t^{\mathbf{Q}}\}_{t=1}^n$ ($s_t^{\mathbf{Q}} < 0$: shifted, and $s_t^{\mathbf{Q}} > 0$: not shifted) as

$$\overrightarrow{\boldsymbol{h}}_t^{\mathbf{Q}} = \overrightarrow{\mathrm{LSTM}}(\boldsymbol{e}_t^{\mathbf{Q}}), \overleftarrow{\boldsymbol{h}}_t^{\mathbf{Q}} = \overleftarrow{\mathrm{LSTM}}(\boldsymbol{e}_t^{\mathbf{Q}}), \tag{39}$$

$$\overleftarrow{s}_t^{\mathbf{Q}} = \tanh(\boldsymbol{v}^{left} \cdot \overleftarrow{\boldsymbol{h}}_t^{\mathbf{Q}}), \overrightarrow{s}_t^{\mathbf{Q}} = \tanh(\boldsymbol{v}^{right} \cdot \overrightarrow{\boldsymbol{h}}_t^{\mathbf{Q}}), \tag{40}$$

$$s_t^{\mathbf{Q}} := \overrightarrow{s}_t^{\mathbf{Q}} \cdot \overleftarrow{s}_t^{\mathbf{Q}}. \tag{41}$$

Here, $\boldsymbol{e}_t^{\mathbf{Q}}$ represents the embedding of $w_t^{\mathbf{Q}}$ from $\boldsymbol{W}^{em}$, $\overrightarrow{\mathrm{LSTM}}$ and $\overleftarrow{\mathrm{LSTM}}$ represents the conversions by forward and backward LSTMs, $\boldsymbol{v}^{right}, \boldsymbol{v}^{left} \in \mathbb{R}^e$ are parameter values. $\overrightarrow{s}_t^{\mathbf{Q}}$ and $\overleftarrow{s}_t^{\mathbf{Q}}$ denote whether or not the sentiment of $w_t^{\mathbf{Q}}$ is shifted by the left-side and right-side terms of $w_t^{\mathbf{Q}}$: $\{w_{t'}^{\mathbf{Q}}\}_{t'=1}^{t-1}$ and $\{w_{t'}^{\mathbf{Q}}\}_{t'=t+1}^{n}$, respectively.

GIL This layer represents the word-level global important point representations $\{\alpha_t^{\mathbf{Q}}\}_{t=1}^n$ using a revised self-attention mechanism [57, 60] as

$$\alpha_t^{\mathbf{Q}} := \sum_{t'=1}^{T} \frac{e^{\tanh(\overrightarrow{\boldsymbol{h}}_t^{\mathbf{Q}T}\overrightarrow{\boldsymbol{h}}_{t'}^{\mathbf{Q}} + \overleftarrow{\boldsymbol{h}}_t^{\mathbf{Q}T}\overleftarrow{\boldsymbol{h}}_{t'}^{\mathbf{Q}})}}{\sum_{t=1}^{T} e^{\tanh(\overrightarrow{\boldsymbol{h}}_t^{\mathbf{Q}T}\overrightarrow{\boldsymbol{h}}_{t'}^{\mathbf{Q}} + \overleftarrow{\boldsymbol{h}}_t^{\mathbf{Q}T}\overleftarrow{\boldsymbol{h}}_{t'}^{\mathbf{Q}})}}.$$

WLCSL This layer represents word-level local contextual sentiment representations $\{c_t^{\mathbf{Q}}\}_{t=1}^n$ as

$$c_t^{\mathbf{Q}} = s_t^{\mathbf{Q}} \cdot p_t^{\mathbf{Q}}.$$

WGCSL This layer represents word-level global contextual sentiment representations $\{g_t^{\mathbf{Q}}\}_{t=1}^n$ as

$$g_t^{\mathbf{Q}} := c_t^{\mathbf{Q}} \alpha_t^{\mathbf{Q}}.$$

CCSL This layer represents the contextual concept-level sentiment representation as

$$\boldsymbol{v}^{\mathbf{Q}} := \sum_{t=1}^{n} g_t^{\mathbf{Q}} \boldsymbol{b}_t^{\mathbf{Q}}$$

where $\boldsymbol{b}_t^{\mathbf{Q}} := \max(\mathrm{Softmax}(\boldsymbol{W}_c \boldsymbol{e}_t^{\mathbf{Q}} - t_c), 0)$, $\boldsymbol{v}_t^{\mathbf{Q}} \in \mathbb{R}^K$, $\boldsymbol{b}_t^{\mathbf{Q}} \in \mathbb{R}^K$, $t_c > 0$ is a hyper-parameter value, $\boldsymbol{W}_c \in \mathbb{R}^{K \times e}$ is centroid vectors of $\{\boldsymbol{w}_i^{em}\}_{i=1}^v$ calculated using a spherical k-means method [26] where the cluster number is K.

Output Then, CSNN outputs a predicted sentiment tag $y^{\mathbf{Q}} \in \{0(\text{negative}), 1(\text{positive})\}$ as

$$\boldsymbol{a}^{\mathbf{Q}} = \mathrm{Softmax}(\boldsymbol{W}^O \tanh(\boldsymbol{v}^{\mathbf{Q}}))y^{\mathbf{Q}} = \mathrm{argmax}\ \boldsymbol{a}^{\mathbf{Q}}$$

where $\boldsymbol{W}^O \in \mathbb{R}^{2 \times K}$ is the parameter value.

7.3 Learning Strategy for CSNN

7.3.1 Initialization and Propagation (IP) Learning This section describes the learning strategy of the CSNN. Overall process is described in Algorithm 7 where $w_{i,j}^o$ is the (i,j) element of $\boldsymbol{W}^O$, and $L_{doc}^{\mathbf{Q}}$ is the cross entropy between $\boldsymbol{a}^{\mathbf{Q}}$ and $d^{\mathbf{Q}}$. IP learning utilizes the two specific techniques called *Update* and *Lexical Initialization*. *Update* is a strategy for improving the interpretability in $WLCSL$ and $WGCSL$. *Lexical Initialization* is a strategy for improving the interpretability in $WOSL$ and GIL. Using both the *Update* and *Lexical Initialization*, the interpretability in SSL is also expected to be improved (as theoretically analyzed in Section 7.5).

In the above, *Update* is the main different point from LEXIL and PLEXIL. We utilize *Update* due to the difference in structures between the CSNN and BINNs.

Update First, $\boldsymbol{W}^O$ is updated according to processes 6–7 in Algorithm 7. This makes $WLCSL$ and $WGCSL$ to represent the corresponding sentiment scores (Proposition 7.5.1 in Section 7.5) without violating the learning process after sufficient iterations (Proposition 7.5.2 in Section 7.5).

Lexical Initialization Then, $\boldsymbol{W}^p$ is initialized as process 2 in Algorithm 7, where $PS(w_i)$ is the sentiment score for word w_i given by the word sentiment dictionary, and S^d is a set of words from the dictionary. Init makes $WOSL$ and SSL represent the corresponding scores in the condition that Update is utilized.

Through this IP learning, for every word sufficiently similar to any of the words in S^d, the $WOSL$, SSL, $WLCSL$, GIL, and $WGCSL$ learn to represent the corresponding scores, as theoretically analyzed in Section 7.5. After the learning, the CSNN can explain its prediction result using these layers.

Algorithm 7 Initialization and Propagation (IP) Learning

1: **for** $i \leftarrow 1$ to v **do**
2: $w_i^p \leftarrow \begin{cases} PS(w_i) & (w_i \in S^d) \\ 0 & (\text{otherwise}) \end{cases}$;
3: **while** learning has not been finished **do**
4: Update $\boldsymbol{W}^p$, $\boldsymbol{v}^{right}$, $\boldsymbol{v}^{left}$, $\boldsymbol{W}^O$ and the LSTM cells in CSNN using the gradient values by $L_{doc}^{\mathbf{Q}}$. ;
5: **for** $k \leftarrow 1$ to K **do**
6: **if** $w_{1,k}^o > 0$ **then** $w_{1,k}^o \leftarrow 0$;
7: **if** $w_{2,k}^o < 0$ **then** $w_{2,k}^o \leftarrow 0$;

7.3.2 Joint Initialization and Propagation (JIP) Learning In the same way as the IP learning, the JSP learning can be converted to the suitable version for the CSNN as shown in Algorithm 8.

Algorithm 8 Joint Initialization and Propagation (JIP) Learning

1: **for** $i \leftarrow 1$ to v **do**
2: $w_i^p \leftarrow \begin{cases} PS(w_i) & (w_i \in S^d) \\ 0 & (\text{otherwise}) \end{cases}$;
3: **while** learning has not been finished **do**
4: Update $\boldsymbol{W}^p$, $\boldsymbol{v}^{right}$, $\boldsymbol{v}^{left}$, $\boldsymbol{W}^O$ and the LSTM cells in CSNN using the gradient values by $L_{joint}^{\mathbf{Q}}$. ;
5: **for** $k \leftarrow 1$ to K **do**
6: **if** $w_{1,k}^o > 0$ **then** $w_{1,k}^o \leftarrow 0$;
7: **if** $w_{2,k}^o < 0$ **then** $w_{2,k}^o \leftarrow 0$;

7.4 Experimental Evaluation

This section experimentally tests the explanation ability and predictability of the CSNN and investigate the effect of IP learning for the interpretability of the layers in the CSNN.

7.4.1 Dataset

Text Corpus We used the following four textual corpora, namely, EcoRevs I, EcoRevs II, Yahoo review, and Sentiment 140, which are used in Sections 4.4 and 4.5. Each textual corpus includes reviews and their sentiment tags, for this evaluation. They were used for developing CSNN. EcoRevs and Yahoo review were Japanese datasets, and Sentiment 140 was an English dataset. We used them to verify whether the CSNN can be used irrespective of the language or domain. We divided each dataset into the training, validation, and test datasets, as presented in Table 1.

Annotated Dataset For this evaluation, we used the annotated dataset that is also used in Sections 4.4 and 4.5 including the Economy, Yahoo, and message annotated datasets.

7.4.2 CSNN Development Setting We developed the CSNN using each training and validation datasets in the following settings.

Setting in Lexical Initialization. *Lexical Initialization* used a part of a Japanese financial word sentiment dictionary (JFWS dict) developed by six financial professionals and the Vader word sentiment dictionary (Vader dict) [17]. These dictionaries contain words and their sentiment scores. After we excluded the words with zero sentiment scores and those with absolute sentiment scores of less than 1.0 from JFWS dict and the Vader dict, respectively, we extracted most frequen 200 words in each training dataset from these dictionaries and used their sentiment scores in *Lexical Initialization*.

Other settings. We calculated the word embedding matrix $\boldsymbol{W}^{em}$ by the skip-gram method (window size = 5) [39] based on each textual dataset. We set the dimensions of the hidden and embedding vectors to 200, epoch to 50 with early stopping, K to $[100, 500, 1000]$, t_c to $1/K$. We determined the hyper-parameters using the validation data. We used the mean score of the five trials for the evaluations in this evaluation.

7.4.3 Evaluation Metrics in Explanation ability Evaluation Metric. We evaluated the explanation ability of the CSNN based on the validity in WOSL, SSL, WLCSL, GIL, and WGCSL in the following way.

Evaluation Metric

WOSL. We evaluated the validity of WOSL based on the agreement between the polarities of word w_i and w_i^p using the economic, Yahoo, and LEX word polarity list[1]. These lists include words and their positive or negative polarities. LEX word-polarity list includes English terms, and the others include Japanese economic terms.

SSL. Using the sentiment shift tags, we evaluated the validity of the SSL based on the agreement between the sentiment shift tag of $w_t^{\mathbf{Q}}$ and the polarity of $s_t^{\mathbf{Q}} > 0$ (shifted: $w_i^p < 0$ and non-shifted: $w_i^p > 0$).

WLCSL. Using the word or phrase level contextual sentiment tags, we evaluated the validity of the WLCSL based on the agreement between the polarity of $c_t^{\mathbf{Q}}$ and the contextual word-level sentiment tag of $w_t^{\mathbf{Q}}$ or the agreement between the polarity of the summed scores for terms involved in each phrase accurately and its phrase-level sentiment. We used the micro and macro average scores between the macro F_1 score for shifted terms and that for non-shifted terms for the evaluation basis. We used the micro-average score to test whether each method could work in real situations, and macro-average score to test whether each method could accurately correspond to both shifted and non-shifted terms.

GIL. Using the gold word-level global important points, we evaluated the validity of GIL based on the correlation between $\{\alpha_t^{\mathbf{Q}}\}_{t=1}^n$ and gold word-level global important points. We used the Pearson correlation for this evaluation.

WGCSL. We evaluated the explanation validity of the WGCSL based on the agreement between the polarities of $\sum_{t=1}^n g_t^{\mathbf{Q}}$ and the document-level sentiment tag of $\mathbf{Q}$. We used the macro F_1 score as an evaluation basis.

In the above evaluations, we used the Economy, Yahoo, and message annotated datasets when developing CSNNs with the corresponding text corpus, respectively. We only employed tags of

[1] http://quanteda.io/reference/data_dictionary_LSD2015.html

terms that were not used in Lexical Initialization and appeared in the training dataset. Table 1 summarizes the numbers of tags used.

Comparison for the learning strategy

To evaluate the effect of IP learning, we compared the results of the following five types of CSNNs, namely, $CSNN^{Base}$, $CSNN^{Rand}$, $CSNN^{NoUp}$, $CSNN^{IP}$, and $CSNN^{JIP}$. The structures of these models are the same as that of CSNN; the differences are summarized as below.

I) $CSNN^{Base}$ is developed using the general backpropagation and without Update or Lexical Initialization strategy.

II) $CSNN^{Rand}$ is developed with only Update strategy.

III) $CSNN^{NoUp}$ is developed with only Lexical Initialization strategy.

III) $CSNN^{IP}$ is developed with IP learning.

VI) $CSNN^{JIP}$ is developed with JIP learning.

Comparison Method

To evaluate the explanation ability of CSNN, we compared the evaluation result of CSNN with other comparative methods in each layer validity.

1) WOSL: This evaluation compared the CSNN with the other word-level original sentiment assignment methods, namely, PMI [40], logistic fixed weight model (LFW) [58], sentiment-oriented NN (SONN) [34], and gradient interpretable neural network (GINN) [21].

2) SSL: This evaluation compared the CSNN with the baseline , NegRNN methods, and Recursive Neural Tensor Network (RNTN) [53]. In the baseline, we predicted $w_t^{\mathbf{Q}}$ as "shifted" if the sentiment of $d^{\mathbf{Q}}$ predicted by the RNN and sentiment tag of $w_t^{\mathbf{Q}}$ assigned by the PMI were different and as "not shifted" in other cases. In NegRNN, we used the RNN that predicts polarity shifts [11] developed with the the polarity shifting training data created by the weighed frequency odds method [36].

3) WLCSL: This evaluation compared the CSNN with the other word-level sentiment assignment methods: PMI, LFW, SONN, GINN, Grad + a bidirectional LSTM model (RNN) [27], LRP + RNN [1], and IntGrad + RNN [56], and Recursive Neural Tensor Network (RNTN) [53].

4) GIL: This evaluation compared the CSNN with the other word-level important point assignment methods using the RNNs using attention mechanism: word attention network (ATT) [66], hierarchical attention network (HN-ATT) [66], sentiment and negation neural network (SNNN) [16], and lexicon-based supervised attention (LBSA) [64]. SNNN and LBSA are set up in a form that the attention weights of terms with the strong word-level original sentiment are strengthened. We used the attention score of each model as the score.

5) WGCSL: This evaluation compared the CSNN with the comparative methods used in the evaluation in WLCSL.

7.4.4 Evaluation Metrics in Predictability Evaluation Metric. We evaluates the predictability of the CSNN based on whether it can predict the sentiment tags of reviews in each test dataset.

Comparison Method. We compared the CSNN and the following methods: logistic regression (LR), LFW [58], SONN [34], GINN [21], a bi-LSTM based RNN (RNN), convolutional

NN (CNN) [30], ATT [66], HN-ATT [66], SNNN [16], LBSA [64]. We used the macro F_1 score as the evaluation basis.

7.4.5 Result and Discussion

Explanation ability and Predictability Tables 9 and 10 summarize the results for explanation ability, indicating that the proposed CSNN outperformed the other methods in most cases. Table 11 summarizes the results, indicating that HN-ATT had greater predictability than the proposed CSNNs; however, CSNN (200) had greater predictability than LR and some deep NNs such as CNN and SNNN, and had predictability equivalent to that of ATT or LBSA.

These results demonstrate that the proposed CSNN has both the high explanation ability and high predictability.

Effect of IP learning The results of CSNNs, $CSNN^{Base}$, $CSNN^{NoUp}$, and $CSNN^{Rand}$ for explainability demonstrate the effect of IP learning as follows. The $CSNN^{Rand}$ outperformed the $CSNN^{Base}$ in WLCSL and WGCSL, indicating that Update promoted the validity in WLCSL and WGCSL; whereas, the $CSNN^{NoUp}$ outperformed the $CSNN^{Base}$ in WOSL and GIL, indicating that Init promoted the validity in WOSL and GIL. Consequently, the validity in all the five layers were improved by using both Update and Lexical Initialization, and the CSNNs outperformed the $CSNN^{Base}$ in all the cases. This is the expected result as described in Section 3.4.1.

Effect of the SSL regularization As for the SSL regularization, its effect was not observed in this case. This is due to the success of the IP learning for the CSNN in all the cases.

Ablation Abalysis To analyze the results in cases where fewer words were used in the Lexical Initialization, we evaluated the $CSNN^{IP}$ developed with 50, and 100 words: $CSNN^{IP}$ (50) and $CSNN^{IP}$ (100). The results are shown in Tables 9, 10, and 11, indicating that the interpretability in the layers can succeed even when we used fifty terms. This result indicates the availability of our approach.

7.4.6 Text-Visualization Example This section introduces some examples of text-visualization produced by the CSNN. Fig. 17 shows the text-visualization examples. Users can explain the CSNN's prediction process based on this type of text-visualizations.

7.5 Conclusion

This chapter introduces a CSNN as an example of the more complex version of the BINN. In this CSNN, PLEXIL and PJSP learning can not be available directly due to the complex structure of the CSNN. Therefore, this chapter converts the LEXIL and JSP lerning to the suitable manner for the CSNN, that is IP learning and JIP learning. Using several textual datasets, we experimentally demonstrate that 1) IP learning and JIP learning are effective for improving the interpretability of layers in CSNN, and that 2) both the explanation ability and predictability of the CSNN are high.

From this example, we can see that the idea behind the proposition of LEXIL and JSP learning can be utilized in a flexible way and the proposed basic learning strategy for developing

interpretable NNs can be utilized to many cases by appropriate conversions.

Table 9: Evaluation Result for Explanation Ability

	EcoRev I	EcoRev II	Yahoo	Sentiment 140
	Evaluation Result of WOSL (Macro F_1 score)			
PMI	0.734	0.745	0.793	0.733
LFW	0.715	0.740	0.766	0.725
SONN	0.702	0.724	0.725	0.705
GINN	0.723	0.755	0.754	0.735
$CSNN^{Base}$	0.417	0.381	0.499	0.373
$CSNN^{NoUp}$	0.832	0.846	0.798	**0.754**
$CSNN^{Rand}$	0.452	0.543	0.460	0.430
$CSNN^{IP}$	0.837	**0.865**	**0.825**	0.742
$CSNN^{JIP}$	0.841	0.831	0.811	0.746
$CSNN^{IP}$ (100)	0.838	0.851	0.817	0.744
$CSNN^{IP}$ (50)	**0.843**	**0.865**	0.805	0.743
	Evaluation Result of SSL (Macro F_1 score)			
Baseline	0.660	0.712	0.579	0.560
NegRNN	0.536	0.626	0.564	0.558
RNTN	-	-	-	0.436
$CSNN^{Base}$	0.661	0.311	0.244	0.314
$CSNN^{NoUp}$	0.374	0.246	0.360	0.417
$CSNN^{Rand}$	0.263	0.531	0.315	0.293
$CSNN^{IP}$	0.777	0.804	**0.691**	0.743
$CSNN^{JIP}$	0.783	0.830	0.660	0.741
$CSNN^{IP}$ (100)	0.780	**0.816**	0.681	0.751
$CSNN^{IP}$ (50)	**0.784**	0.809	0.675	**0.762**

Evaluation Result of WLCSL (Macro F_1 score)

Level	EcoRev I word		EcoRev II word		Yahoo word		Sentiment 140 word		Sentiment 140 phrase
	micro	macro	micro	macro	micro	macro	micro	macro	
RNTN	-	-	-	-	-	-	.670	.570	.620
PMI	.792	.578	.788	.548	**.823**	.575	.854	.631	.822
Grad + RNN	.703	.578	.743	.621	.713	.601	.793	.681	.743
IntGrad + RNN	.801	.607	.775	.621	.752	.625	.842	.679	.79.6
LRP + RNN	.805	.597	.741	.518	.761	.579	.834	.638	.808
LFW	.789	.549	.791	.545	.811	.578	.832	.587	.749
SONN	.767	.555	.788	.542	.769	.566	.866	.600	.787
GINN	.796	.569	.790	.555	.770	.577	.861	.623	.831
$CSNN^{Base}$	.378	.355	.626	.521	.522	.490	.612	.575	.595
$CSNN^{NoUp}$	.427	.416	.273	.316	.566	.526	.505	.509	.512
$CSNN^{Rand}$	*.714*	*.606*	*.763*	*.621*	*.674*	*.516*	*.810*	*.794*	*.748*
$CSNN^{IP}$	.855	.676	.878	.711	.817	.669	.891	**.788**	.858
$CSNN^{JIP}$	.679	.861	.868	.762	.792	.663	.891	.782	0.858
$CSNN^{IP}$ (100)	.849	.679	.879	**.723**	.812	**.675**	**.893**	.784	**.862**
$CSNN^{IP}$ (50)	**.861**	**.692**	**.880**	.719	.797	.670	.889	**.788**	.857

Table 10: Evaluation Result for Explanation Ability

	Evaluation Result of GIL (Pearson Correlation)			
ATT	-0.015	-0.081	0.062	-
HN-ATT	0.108	0.188	0.262	-
SNNN	0.281	0.456	0.192	-
LBSA	0.333	0.344	**0.405**	-
$CSNN^{Base}$	0.014	0.170	0.171	-
$CSNN^{NoUp}$	**0.607**	**0.590**	0.329	-
$CSNN^{Rand}$	0.207	0.224	0.164	-
$CSNN^{IP}$	0.595	0.580	0.325	-
$CSNN^{JIP}$	0.611	0.558	0.338	
$CSNN^{IP}$ (100)	0.584	0.567	0.308	
$CSNN^{IP}$ (50)	0.585	0.562	0.321	-
	Evaluation Result of WGCSL (Macro F_1 score)			
PMI	0.827	0.800	0.673	0.759
LFW	0.876	0.840	0.751	0.745
SONN	0.863	0.876	0.717	0.776
GINN	0.860	0.859	0.740	0.782
Grad + RNN	0.870	0.899	0.724	0.718
IntGrad + RNN	0.909	0.929	0.750	0.755
LRP + RNN	0.909	0.909	0.751	0.818
$CSNN^{Base}$	0.248	0.709	0.534	0.615
$CSNN^{NoUp}$	0.417	0.239	0.533	0.565
$CSNN^{Rand}$	0.911	0.916	0.717	**0.831**
$CSNN^{IP}$	**0.923**	0.937	**0.771**	**0.830**
$CSNN^{JP}$	0.922	0.932	0.755	0.830
$CSNN^{IP}$ (100)	0.916	0.935	0.768	0.829
$CSNN^{IP}$ (50)	0.918	**0.938**	0.766	**0.831**

Table 11: F_1 score results for the predictability evaluation

	EcoRev I	EcoRev II	Yahoo	Sentiment 140
LR	0.878	0.879	0.741	0.785
CNN	0.894	0.911	0.757	0.820
RNN	0.922	0.932	0.749	**0.837**
ATT	0.924	0.937	0.750	0.835
HN-ATT	**0.927**	**0.940**	0.750	**0.837**
SNNN	0.918	0.928	0.752	0.827
LBSA	0.922	**0.941**	0.762	0.832
$CSNN^{IP}$	0.921	0.938	**0.768**	0.833
$CSNN^{JIP}$	0.919	0.937	0.762	0.833
$CSNN^{IP}$ (100)	0.914	0.937	**0.762**	0.835
$CSNN^{IP}$ (50)	0.916	0.939	**0.765**	0.833

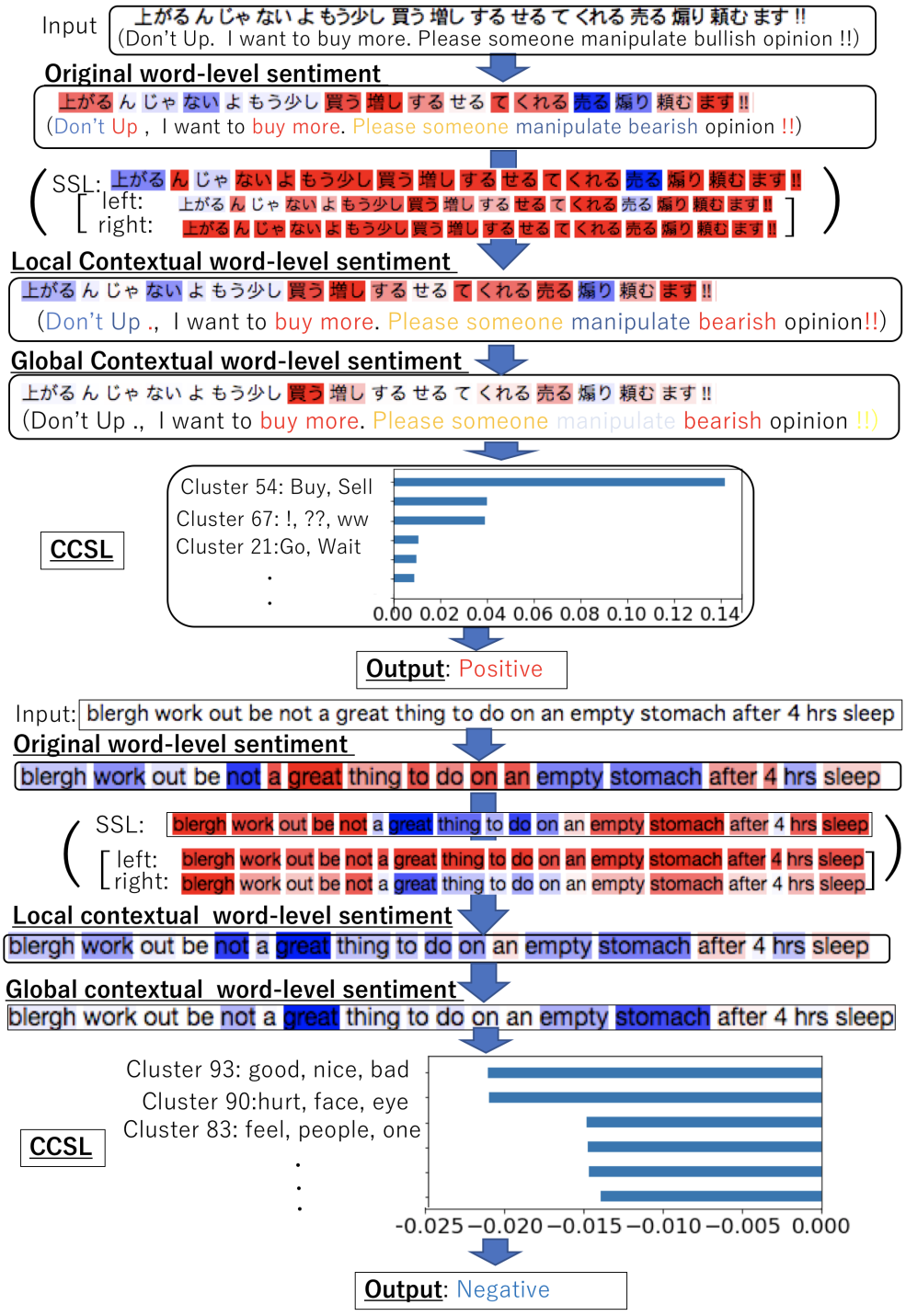

Figure 17: Local Sentiment Text-visualization Example. Left: Yahoo review and right: Sentiment 140. The color and depth of terms mean polarity (red: > 0 and blue: < 0) and scale of word-level sentiments in each layer.

Appendix: Theoretical Analysis for IP learning

In IP learning the following two propositions are satisfied.

PROPOSITION 7.5.1. *For every* $c_{it}^{\mathbf{Q}} \in \{\{c_{it}^{\mathbf{Q}}\}_{t=1}^{n} | \mathbf{Q} \in \Omega^{tr}\}$,

$$\frac{\partial L_{doc}^{\mathbf{Q}}}{\partial c_{it}^{\mathbf{Q}}} \begin{cases} < 0 & (d^{\mathbf{Q}} = 1) \\ > 0 & (d^{\mathbf{Q}} = 0) \end{cases} \tag{42}$$

and

$$\frac{\partial L_{doc}^{\mathbf{Q}}}{\partial g_{it}^{\mathbf{Q}}} \begin{cases} < 0 & (d^{\mathbf{Q}} = 1) \\ > 0 & (d^{\mathbf{Q}} = 0) \end{cases} \tag{43}$$

are established.

PROPOSITION 7.5.2. *Let the values of* $\boldsymbol{W}^{O}$ *before and after performing* Update *in Algorithm 3 in the* tth *iteration be* $\boldsymbol{W}_t^{O,a}$ *and* $\boldsymbol{W}_t^{O,b}$*, respectively. Then,* $\frac{\|\boldsymbol{W}_t^{O,a} - \boldsymbol{W}_t^{O,b}\|_2}{\|\boldsymbol{W}_t^{O,b}\|_2} \xrightarrow[t\to\infty]{} 0$. *is established.*

PROPOSITION 7.5.3. *When Init is used, then, if* $\min_{w_j \in S^d} |e_t^{\mathbf{Q}} - \boldsymbol{w}_j^{em}| < \epsilon$ *where* $\epsilon > 0$ *is sufficiently small, then,*

$$\operatorname{sign}\left(\frac{\partial L^{\mathbf{Q}'}}{\partial p_{t'}^{\mathbf{Q}'}}\right) = \begin{cases} R(w_{t'}^{\mathbf{Q}'(w_{t'}^{\mathbf{Q}'}, w_j)}) & (d^{\mathbf{Q}} = 0) \\ -R(w_{t'}^{\mathbf{Q}'(w_{t'}^{\mathbf{Q}'}, w_j)}) & (d^{\mathbf{Q}} = 1) \end{cases}.$$

where

$$I(a,b) := \begin{cases} 1 & (a = b) \\ 0 & (a \neq b), \end{cases}, \Psi(a,b) := \begin{cases} 1 & (a = b) \\ -1 & (a \neq b) \end{cases}$$

is established.

- From 'Proposition 7.5.1, the validity of the IP learning for the interpretability of layers in $BINN^{type1}$ can be explained in the same manner as in LEXIL.
- From Proposition 7.5.2, we can see that using IP learning, the $BINN^{type1}$ can acquire the local optima because the effect of Update decreases and will be vanished after the sufficient times of iterations.
- Proposition 7.5.3 explains the effect of Init for the word-level original sentiment assignment property of CSNN.

We introduce the proofs of Propositions 7.5.1–7.5.2

Proof of Proposition 7.5.1

$$\frac{\partial L^{\mathbf{Q}}}{\partial c_{it}^{\mathbf{Q}}} = \frac{\partial L^{\mathbf{Q}}}{\partial \boldsymbol{a}^{\mathbf{Q}}} \frac{\partial \boldsymbol{a}^{\mathbf{Q}}}{\partial (\tanh(\sum_{t=1}^{n} \boldsymbol{v}_{it}^{\mathbf{Q}}))} \frac{\partial (\tanh(\sum_{t=1}^{n} \boldsymbol{v}_{it}^{\mathbf{Q}}))}{\partial \sum_{t=1}^{n} \boldsymbol{v}_{it}^{\mathbf{Q}}} \frac{\partial \sum_{t=1}^{n} \boldsymbol{v}_{it}^{\mathbf{Q}}}{\partial g_{it}^{\mathbf{Q}}} \frac{\partial g_{it}^{\mathbf{Q}}}{\partial c_{it}^{\mathbf{Q}}}$$

$$= \boldsymbol{\Delta}_o^{\mathbf{Q}} \boldsymbol{W}^o \boldsymbol{b}_{it}^{\mathbf{Q}} \mathrm{diag}(1 - (\tanh(\sum_{i=1}^{L} \sum_{t=1}^{T} \boldsymbol{v}_{it}^{\mathbf{Q}}))^2) \alpha_{it}^{\mathbf{Q}} = \boldsymbol{M}_{it}^{\mathbf{Q}^T} \boldsymbol{\Delta}_o^{\mathbf{Q}} \alpha_{it}^{\mathbf{Q}}$$

where

$$\boldsymbol{M}_{it}^{\mathbf{Q}} = \boldsymbol{W}^o \mathrm{diag}(1 - (\tanh(\sum_{i=1}^{L} \sum_{t=1}^{n} \boldsymbol{v}_{it}^{\mathbf{Q}}))^2) \boldsymbol{b}_{it}^{\mathbf{Q}}$$

$$\boldsymbol{\Delta}_o^{\mathbf{Q}} = \begin{cases} \boldsymbol{a}^{\mathbf{Q}} - (1,0)^T & (d^{\mathbf{Q}} = 0) \\ \boldsymbol{a}^{\mathbf{Q}} - (0,1)^T & (d^{\mathbf{Q}} = 1) \end{cases}$$

In addition,

$$\frac{\partial L_{doc}^{\mathbf{Q}}}{\partial g_{it}^{\mathbf{Q}}} = \frac{\partial L_{doc}^{\mathbf{Q}}}{\partial \boldsymbol{a}^{\mathbf{Q}}} \frac{\partial \boldsymbol{a}^{\mathbf{Q}}}{\partial (\tanh(\sum_{t=1}^{n} \boldsymbol{v}_{it}^{\mathbf{Q}}))} \frac{\partial (\tanh(\sum_{t=1}^{n} \boldsymbol{v}_{it}^{\mathbf{Q}}))}{\partial \sum_{t=1}^{n} \boldsymbol{v}_{it}^{\mathbf{Q}}} \frac{\partial \sum_{t=1}^{n} \boldsymbol{v}_{it}^{\mathbf{Q}}}{\partial g_{it}^{\mathbf{Q}}}$$

$$= \boldsymbol{\Delta}_o^{\mathbf{Q}} \boldsymbol{W}^o \boldsymbol{b}_{it}^{\mathbf{Q}} \mathrm{diag}(1 - (\tanh(\sum_{t=1}^{n} \boldsymbol{v}_{it}^{\mathbf{Q}}))^2) \alpha_{it}^{\mathbf{Q}} = \boldsymbol{M}_{it}^{\mathbf{Q}^T} \boldsymbol{\Delta}_o^{\mathbf{Q}}.$$

Here, $\frac{\partial L}{\partial c^{\mathbf{Q}}}$ and $\frac{\partial L}{\partial g^{\mathbf{Q}}}$ are positive and negative when $d^{\mathbf{Q}} = 0$ and $d^{\mathbf{Q}} = 1$, respectively, $(t = 1, 2, \cdots, n,)$ because $m_{it}^{\mathbf{Q}}, 0 \leq 0$ and $m_{it}^{\mathbf{Q}}, 1 \geq 0$ by *Update*. Therefore, the proposition is established.

Proof of Proposition 7.5.2 *Proof* After the sufficient time of update iterations, for every j,

$$\boldsymbol{u}^{3,\mathbf{Q}} := \tanh(\sum_{t=1}^{n} \boldsymbol{v}_{it}^{\mathbf{Q}})$$

$$E\left[\frac{\partial L^{\mathbf{Q}}}{\partial w_{1,j}^{O}}\right] = E\left[\sum_{\mathbf{Q} \in D^{mini}} \Delta_{o,1}^{\mathbf{Q}} (u_j^{3,\mathbf{Q}})^T\right] > 0$$

$$E\left[\frac{\partial L^{\mathbf{Q}}}{\partial w_{2,j}^{O}}\right] = E\left[\sum_{\mathbf{Q} \in D^{mini}} \Delta_{o,2}^{\mathbf{Q}} (u_j^{3,\mathbf{Q}})^T\right] < 0$$

where $w_{i,j}^{O}$ is the (i,j) element of $\boldsymbol{W}^O$ and D^{mini} is the mini-batch dataset in the learning. Therefore, considering that each value of $\boldsymbol{u}^{3,\mathbf{Q}}$ is negative and positive when $d^{\mathbf{Q}} = 0$ and $d^{\mathbf{Q}} = 1$, respectively, is established because Proposition 7.5.1 is established. Therefore, Proposition 7.5.2 is established.

Proof of Proposition 7.5.3

$$\frac{\partial L^{\mathbf{Q}}}{\partial c_{it}^{\mathbf{Q}}} = \frac{\partial L^{\mathbf{Q}}}{\partial \boldsymbol{a}^{\mathbf{Q}}} \frac{\partial \boldsymbol{a}^{\mathbf{Q}}}{\partial (\tanh(\sum_{t=1}^{n} \boldsymbol{v}_{it}^{\mathbf{Q}}))} \frac{\partial (\tanh(\sum_{t=1}^{n} \boldsymbol{v}_{it}^{\mathbf{Q}}))}{\partial \sum_{t=1}^{n} \boldsymbol{v}_{it}^{\mathbf{Q}}} \frac{\partial \sum_{t=1}^{n} \boldsymbol{v}_{it}^{\mathbf{Q}}}{\partial g_{it}^{\mathbf{Q}}}$$

$$= \boldsymbol{\Delta}_o^{\mathbf{Q}} \boldsymbol{W}^o \boldsymbol{b}_{it}^{\mathbf{Q}} \mathrm{diag}(1 - (\tanh(\sum_{t=1}^{n} \boldsymbol{v}_{it}^{\mathbf{Q}}))^2) \alpha_{it}^{\mathbf{Q}} = \boldsymbol{M}_{it}^{\mathbf{Q}^T} \boldsymbol{\Delta}_o^{\mathbf{Q}} \alpha_{it}^{\mathbf{Q}}$$

Therefore,

$$\frac{\partial L^{\mathbf{Q}}}{\partial p_{it}^{\mathbf{Q}}} = \frac{\partial L^{\mathbf{Q}}}{\partial c_{it}^{\mathbf{Q}}} \frac{\partial c_{it}^{\mathbf{Q}}}{\partial p_{it}^{\mathbf{Q}}} = \boldsymbol{M}_{it}^{\mathbf{Q}^T} \boldsymbol{\Delta}_o^{\mathbf{Q}} \alpha_{it}^{\mathbf{Q}} s_{it}^{\mathbf{Q}}$$

$$\frac{\partial L^{\mathbf{Q}}}{\partial s_{it}^{\mathbf{Q}}} = \frac{\partial L^{\mathbf{Q}}}{\partial c_{it}^{\mathbf{Q}}} \frac{\partial c_{it}^{\mathbf{Q}}}{\partial s_{it}^{\mathbf{Q}}} = \boldsymbol{M}_{it}^{\mathbf{Q}^T} \boldsymbol{\Delta}_o^{\mathbf{Q}} \alpha_{it}^{\mathbf{Q}} p_{it}^{\mathbf{Q}}$$

where

$$\boldsymbol{M}_{it}^{\mathbf{Q}} = \boldsymbol{W}^o \mathrm{diag}(1 - (\tanh(\sum_{t=1}^{n} \boldsymbol{v}_{it}^{\mathbf{Q}}))^2) \boldsymbol{b}_{it}^{\mathbf{Q}}$$

$$\boldsymbol{\Delta}_o^{\mathbf{Q}} = \begin{cases} \boldsymbol{a}^{\mathbf{Q}} - (1,0)^T & (d^{\mathbf{Q}} = 0) \\ \boldsymbol{a}^{\mathbf{Q}} - (0,1)^T & (d^{\mathbf{Q}} = 1) \end{cases}$$

$$\frac{\partial L^{\mathbf{Q}}}{\partial w_{1,j}^{O}} = \Delta_{o,1}^{\mathbf{Q}} (u_j^{3,\mathbf{Q}})^T, \frac{\partial L^{\mathbf{Q}}}{\partial w_{2,j}^{O}} = \Delta_{o,2}^{\mathbf{Q}} (u_j^{3,\mathbf{Q}})^T \tag{44}$$

where

$$\Delta_{o,1}^{\mathbf{Q}} = -\Delta_{o,2}^{\mathbf{Q}}.$$

Therefore,

$$-w_{1,j}^{O} = w_{2,j}^{O} (= \omega_j) \tag{45}$$

is established. Moreover,

$$\frac{\partial L^{\mathbf{Q}}}{\partial \boldsymbol{u}^{3,\mathbf{Q}}} = \frac{\partial L^{\mathbf{Q}}}{\partial \boldsymbol{a}} \frac{\partial \boldsymbol{a}}{\partial w_j^o} = \boldsymbol{\Delta}_o^{\mathbf{Q}} \boldsymbol{W}^o = \boldsymbol{\Delta}_o^{\mathbf{Q}} [-\boldsymbol{\omega}; \boldsymbol{\omega}]$$

$$= \begin{cases} 2|\Delta_{o,1}^{\mathbf{Q}}| \boldsymbol{\omega} & (d^{\mathbf{Q}} = 0) \\ -2|\Delta_{o,1}^{\mathbf{Q}}| \boldsymbol{\omega} & (d^{\mathbf{Q}} = 1) \end{cases} \tag{46}$$

is established. Therefore, after the sufficient time of iterations,

$$E[\boldsymbol{u}^{3,\mathbf{Q}}] = \begin{cases} -k\boldsymbol{\omega} & (d^{\mathbf{Q}} = 0) \\ k\boldsymbol{\omega} & (d^{\mathbf{Q}} = 1) \end{cases}$$

where $k > 0$ is expected to be established.

Therefore,

$$\boldsymbol{M}_{it}^{\mathbf{Q}^T} \boldsymbol{\Delta}_o^{\mathbf{Q}} = \boldsymbol{b}_{it}^{\mathbf{Q}^T} \mathrm{diag}(1 - (\tanh(\sum_{t=1}^{n} \boldsymbol{v}_{it}^{\mathbf{Q}}))^2) \boldsymbol{W}^{oT} \boldsymbol{\Delta}_o^{\mathbf{Q}}$$

$$= \boldsymbol{b}_{it}^{\mathbf{Q}T} \mathrm{diag}(\mathbf{1} - (\tanh(\sum_{t=1}^{n} \boldsymbol{v}_{it}^{\mathbf{Q}}))^2) \boldsymbol{W}^{oT} \boldsymbol{\Delta}_o^{\mathbf{Q}}$$

$$= \begin{cases} 2|\Delta_{o,2}^{\mathbf{Q}}| \boldsymbol{b}_{it}^{\mathbf{Q}T} \mathrm{diag}(\mathbf{1} - (\tanh(\sum_{t=1}^{n} \boldsymbol{v}_{it}^{\mathbf{Q}}))^2) \boldsymbol{\omega} & (d^{\mathbf{Q}} = 0) \\ -2|\Delta_{o,2}^{\mathbf{Q}}| \boldsymbol{b}_{it}^{\mathbf{Q}T} \mathrm{diag}(\mathbf{1} - (\tanh(\sum_{t=1}^{n} \boldsymbol{v}_{it}^{\mathbf{Q}}))^2) \boldsymbol{\omega} & (d^{\mathbf{Q}} = 1) \end{cases}$$

Moreover, after the sufficient times of iterations,

$$\mathrm{sign}(\omega_1) = \mathrm{sign}(\omega_2) = \cdots = \mathrm{sign}(\omega_k) \tag{47}$$

is established because Eq (44) and

$$\boldsymbol{u}^{3,\mathbf{Q}} = \sum_{t=1}^{n} \boldsymbol{v}_{it}^{\mathbf{Q}} = \sum_{t=1}^{n} g_{it}^{\mathbf{Q}} \boldsymbol{b}_{it}^{\mathbf{Q}}$$

where

$$\mathrm{sign}(v_{t,1}^{\mathbf{Q}}) = \mathrm{sign}(v_{t,2}^{\mathbf{Q}}) = \cdots = \mathrm{sign}(v_{t,k}^{\mathbf{Q}}).$$

are satisfied, and in sufficient times of cases,

$$\boldsymbol{u}^{3,\mathbf{Q}} \simeq g_{\hat{t}}^{\mathbf{Q}} \boldsymbol{b}_{\hat{t}}^{\mathbf{Q}} \tag{48}$$

where

$$\hat{t} = \mathrm{argmax}_{it} g_{it}^{\mathbf{Q}}.$$

Eq (48) occurs because

$$E[p_{it}^{\mathbf{Q}} | w_{it}^{\mathbf{Q}} \in S^d] >> E[p_{it}^{\mathbf{Q}} | w_{it}^{\mathbf{Q}} \notin S^d]$$
$$E[\alpha_{it}^{\mathbf{Q}} | \min_{w_j \in S^d} |w_{it}^{\mathbf{Q}} - \boldsymbol{w}_j^{em}| < \epsilon] >> E[\alpha_{it}^{\mathbf{Q}} | \min_{w_j \in S^d} |w_{it}^{\mathbf{Q}} - \boldsymbol{w}_j^{em}| \geq \epsilon],$$

where ϵ is sufficiently small, and $|S^d|$ is sufficiently small.

Therefore,

$$\mathrm{sign}(\boldsymbol{M}_{it}^{\mathbf{Q}T} \boldsymbol{\Delta}_o^{\mathbf{Q}}) = \begin{cases} -\chi & (d^{\mathbf{Q}} = 0) \\ \chi & (d^{\mathbf{Q}} = 1) \end{cases}.$$

Thus,

$$s_{it}^{\mathbf{Q}} \simeq \epsilon - \sum_{\mathbf{Q}' \in \Omega^{tr}} \sum_{t'=1}^{|\mathbf{Q}'|} \boldsymbol{M}_{t'}^{\mathbf{Q}'T} \boldsymbol{\Delta}_o^{\mathbf{Q}'} \alpha_{t'}^{\mathbf{Q}'} p_{t'}^{\mathbf{Q}'} I(|e_t^{\mathbf{Q}} - e_{t'}^{\mathbf{Q}}| < \epsilon, True).$$

$$\simeq - \sum_{\mathbf{Q}' \in \Omega^{tr}} \sum_{t'=1}^{|\mathbf{Q}'|} \boldsymbol{M}_{t'}^{\mathbf{Q}'T} \boldsymbol{\Delta}_o^{\mathbf{Q}'} \alpha_{t'}^{\mathbf{Q}'} p_{t'}^{\mathbf{Q}'} I(|e_t^{\mathbf{Q}} - e_{t'}^{\mathbf{Q}}| < \epsilon, True) I(w_{t'}^{\mathbf{Q}} \in S^d, True).$$

Here,

$$\mathrm{sign}(\boldsymbol{M}_{t'}^{\mathbf{Q}'T} \boldsymbol{\Delta}_o^{\mathbf{Q}'} \alpha_{t'}^{\mathbf{Q}'} p_{t'}^{\mathbf{Q}'}) = \chi R(w_{t'}^{\mathbf{Q}'})$$

Thus, if $w_{it}^{\mathbf{Q}} \in S^d$, then,

$$\mathrm{sign}(s_{it}^{\mathbf{Q}}) = -\chi R(w_t^{\mathbf{Q}})$$

is established because each $w_j \in S^d$ satisfies Condition 3.5.1. Therefore, in such a situation,

$$\mathrm{sign}\left(\frac{\partial L^{\mathbf{Q}}}{\partial p_{it}^{\mathbf{Q}}}\right) = \mathrm{sign}(\boldsymbol{M}_{it}^{\mathbf{Q}T}\boldsymbol{\Delta}_o^{\mathbf{Q}}\alpha_{it}^{\mathbf{Q}}s_{it}^{\mathbf{Q}})$$

$$\simeq \begin{cases} \chi^2 R(w_{it}^{\mathbf{Q}}) = R(w_{it}^{\mathbf{Q}}) & (d^{\mathbf{Q}} = 0) \\ -\chi^2 R(w_{it}^{\mathbf{Q}}) = -R(w_{it}^{\mathbf{Q}}) & (d^{\mathbf{Q}} = 1). \end{cases}$$

Therefore, if $|e_t^{\mathbf{Q}} - \boldsymbol{w}_j^{em}| < \epsilon$ where $\epsilon > 0$ is sufficiently small, then, the following equation is established.

$$\frac{\partial L^{\mathbf{Q}'}}{\partial p_{t'}^{\mathbf{Q}'}} = \boldsymbol{M}_{t'}^{\mathbf{Q}'T}\boldsymbol{\Delta}_o^{\mathbf{Q}'}\alpha_{t'}^{\mathbf{Q}'}s_{t'}^{\mathbf{Q}'} \simeq \boldsymbol{M}_{t'}^{\mathbf{Q}'T}\boldsymbol{\Delta}_o^{\mathbf{Q}'}\alpha_{t'}^{\mathbf{Q}'}s_{t'}^{\mathbf{Q}'(w_{t'}^{\mathbf{Q}'},w_j)}$$

Therefore,

$$\mathrm{sign}\left(\frac{\partial L^{\mathbf{Q}'}}{\partial p_{t'}^{\mathbf{Q}'}}\right) = \begin{cases} R(w_{t'}^{\mathbf{Q}'(w_{t'}^{\mathbf{Q}'},w_j)}) & (d^{\mathbf{Q}} = 0) \\ -R(w_{t'}^{\mathbf{Q}'(w_{t'}^{\mathbf{Q}'},w_j)}) & (d^{\mathbf{Q}} = 1) \end{cases}.$$

because

$$s_{t'}^{\mathbf{Q}'} \simeq s_{t'}^{\mathbf{Q}'(w_{t'}^{\mathbf{Q}'},w_t^{\mathbf{Q}})}$$

due to $|e_t^{\mathbf{Q}} - e_{t'}^{\mathbf{Q}'}| < \epsilon$ where $\epsilon > 0$ is sufficiently small, is established.

Thus, Proposition 7.5.3 is established.

Part III

Application into Text Visualization

Chapter 8

Conceptual Sentiment Visualization (CSCV)

This chapter introduces a *Conceptual Sentiment Cloud Visualization (CSCV)* as an application of our basic learning strategy for developing interpretable NNs. This application demonstrates that our study can be applied into several real world issues.

We first introduce the motivation behind the development of CSCV in Section 8.1 and then explain the framework of CSCV in Section 8.2. We then experimentally evaluate our approach using real textual datasets In Sections 8.3 and 8.4, and then conclude this chapter in Section 8.7.

8.1 Introduction

8.1.1 Motivation Online customer reviews provide opinion-rich information for diverse decision-making processes in improving the service or products. For example, using online customer reviews, shop owners or EC site managers can detect the malicious troubles such as scams in prices or payment in the early stages. Moreover, we can give feedback for the good points or bad points in shops to shop operators.

However, in general, it is difficult to manually read all the reviews. Table 12 represents the number of reviews in Yahoo! Shopping [1] between 2015. The volume of reviews is so large that this is not realistic to read all of them.

Table 12: Dataset Organization for reviews in Yahoo! Shopping Service between 2015

rating	1	2	3	4	5
volume	94,620	30,254	129,837	456,020	1,646,070

From this background, it is a great demand for this area to develop a method for summarizing and visualizing the online reviews in the form that users can quickly catch-up their summary. In the process of decision-making, we need to accurately catch-up both of the following two types of sentiments in a short time:

- aspect-based sentiment: what is good or bad, and

[1] https://shopping.yahoo.co.jp

- sentiment influence: what type of sentiment causes the above aspect-based sentiment.

Figure 18 shows a simple example for these sentiments. In this example, the information that "orders were bad" is not sufficient. We need more information about the reason for the badness in "orders" for the improvement. In this case, the delay (e.g., "took me a while") was one of the main reason. Therefore, in this case, we should extract both the information that the "orders were bad" (aspect-based sentiment) and the information that delay influenced the badness in orders (sentiment influence).

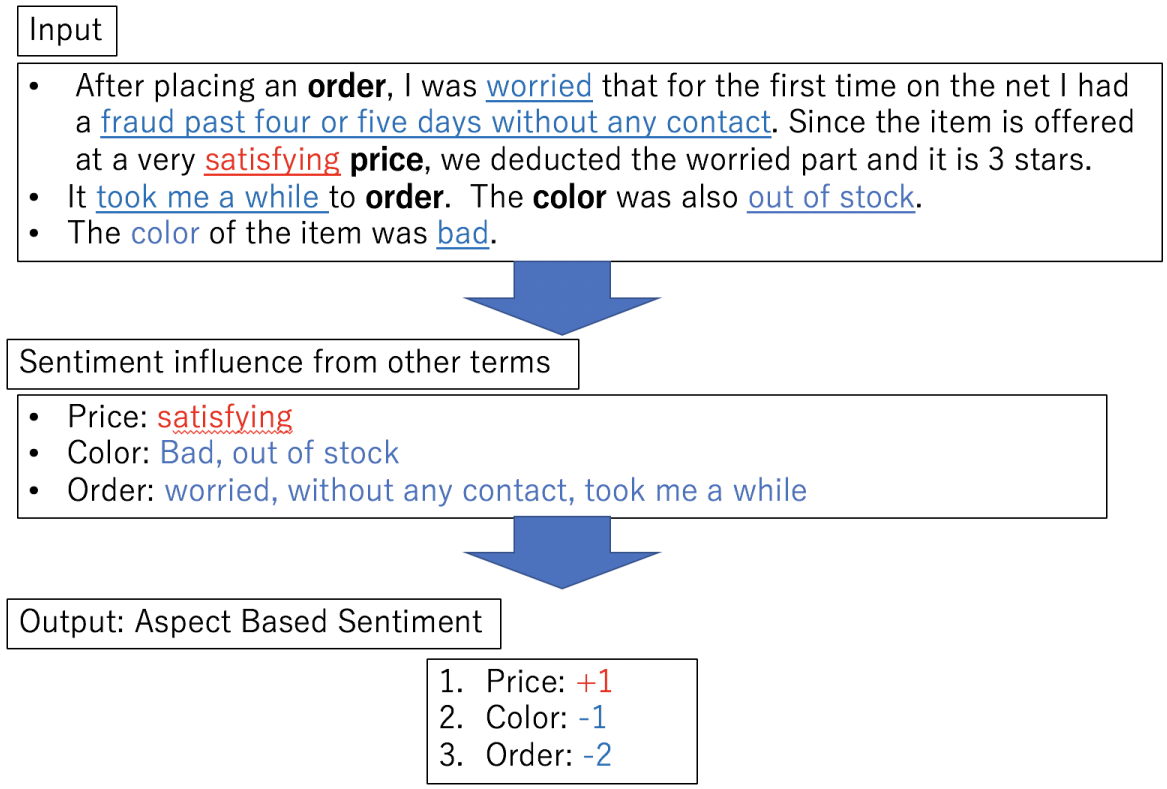

Figure 18: Example for extracting aspect-based sentiment and sentiment influence

In extracting sentiment influences, we should extract both the sentiment score of each word and its influence to the other terms in a review. Here, in extracting the influence to the other terms, we should extract the sentiment score considering the contextual relationships between terms. For example, between "Item arrival is delayed" and "Hospitality is delayed," the former would be worse because the former situation would be more serious for a customer. In extracting aspect-based sentiments, we should extract the total sentiment score of each word that is given by other terms in a review.

Considering the above industrial demand, this study aims to develop a practical method for visualizing both types of sentiments in the form that users can understand the contents of reviews quickly.

8.1.2 Challenge In achieving our aim, we should consider the following two challenges.

Challenge 1: Be Practical ! In this study, we aim to visualize both the aspect-based sentiment and sentiment influence in a practical using *only reviews and their ratings (1–5)* considering the practicality. This problem setting is very challenging because we can not use any specific data or knowledge for these sentiments; however, this problem setting is practical. For example, previous works have been done for extracting aspect-based sentiments [38, 61, 65]; however, most of these methods can not be used because they use the aspect based sentiment tags or special knowledge for aspect-based sentiments. This is not practical because it is not realistic to have such specific data or knowledge in analyzing minor languages such as Japanese.

Challenge 2: Be User-friendly ! In addition, to achieve our purpose, we have to visualize both sense-based and aspect-based sentiments in a user-friendly way. In this chapter, we define user-friendliness as *how accurately* and *how fast* users can catch-up the content of reviews.

8.1.3 Our approach To achieve our aim, we propose a novel text-visualization method called Conceptual Sentiment Cloud Visualization method (CSCV). In CSCV, we use an interpretable neural network model called Text-Visualizing Neural Network Model (TVNN). TVNN corresponds to the simpler version of the CSNN.

TVNN A TVNN includes the following three interpretable layers: original word-level sentiment layer (WOSL), Term Relation Matrix (TRM), and aspect-based word sentiment layer (AWSL). The WOSL represents the original sentiment of each term. TRM represents the relationship between terms in a review. The AWSL represents the sentiment score of each term after considering the influence that was given by the other terms in a document. The TVNN can extract the aspect-based sentiment of each term using AWSL, and the influence sentiments between terms using TRM and WOSL. In addition, it should be noted that this approach is practical because we can develop TVNN only using the reviews and their positive or negative sentiment tags, and do not need any aspect-based sentiment tag.

CSCV The CSCV displays both types of sentiments extracted from the TVNN in a user-friendly way. In the CSCV, the aspect-based sentiments are displayed in concept cluster units in the form that users can grasp what term influences the sentiments of concept clusters using the word cloud approach [28]. Here, a concept means a set of words whose meanings are similar.

8.1.4 Contribution Our contributions are as follows:

1) We proposed a practical text-visualization solution called *CSCV* for visualizing both the aspect-based sentiment and sentiment influence. The method for extracting sentiments and visualizing documents is novel. In addition, the text-visualization design used in the CSCV is novel.

2) Using real user responses, we then demonstrated the usefulness of the CSCV.

8.2 Text Visualization

This section introduces the framework for visualizing reviews using TVNN and CSCV. We can visualize reviews the following steps:

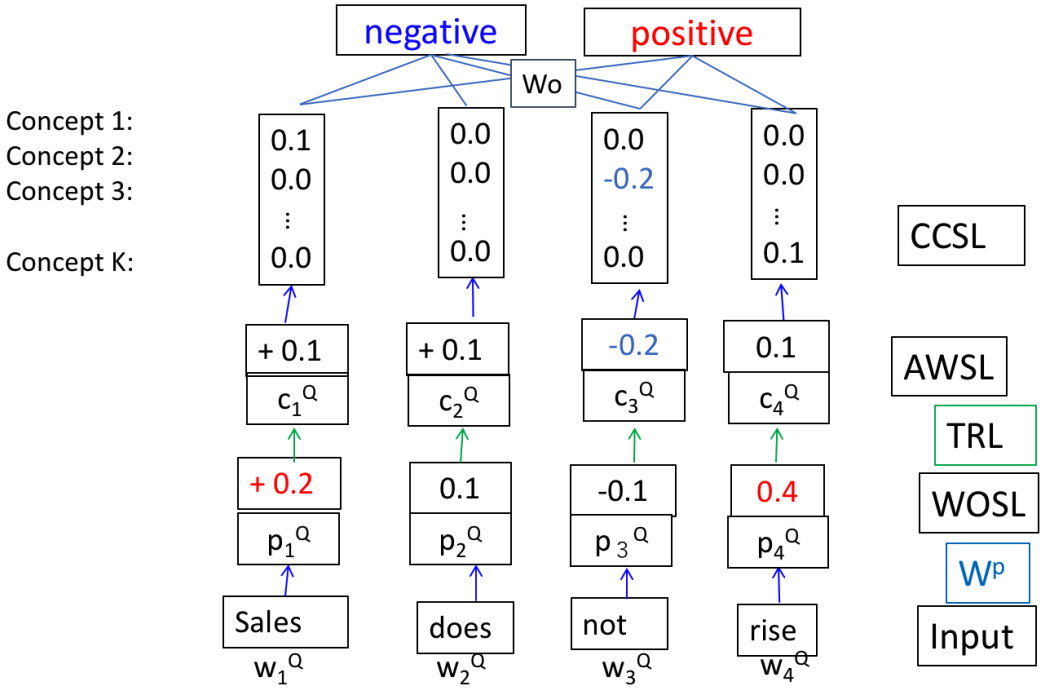

Figure 19: TVNN/TVNN Architecture

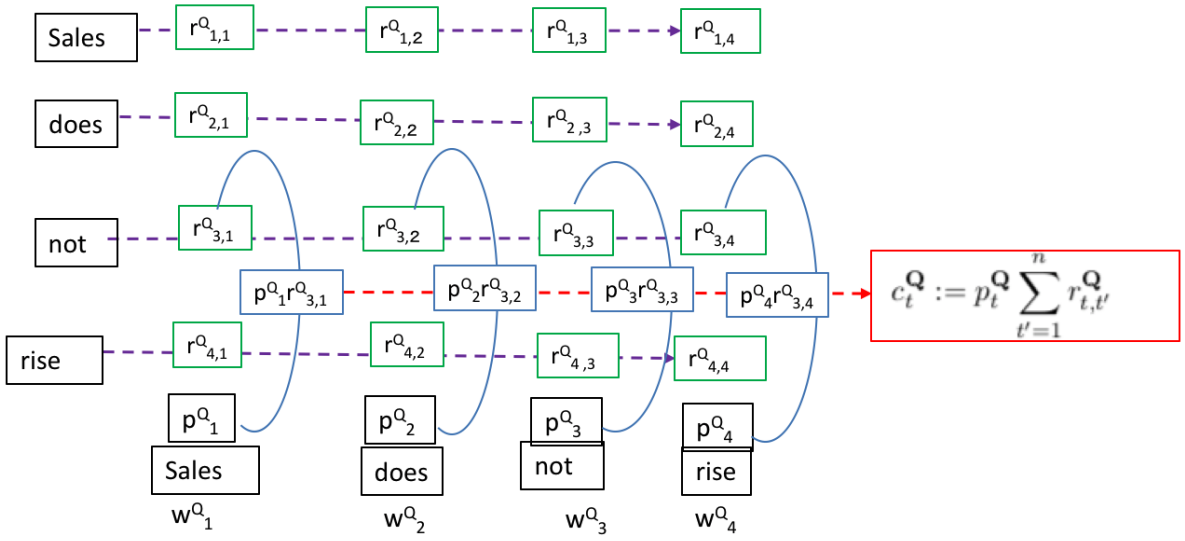

Figure 20: Term Relation matrix

0) prepare a training dataset $\{(\mathbf{Q}_i, d^{\mathbf{Q}_i})\}_{i=1}^{N}$,
1) construct a TVNN model as described in Section 8.2.1,
2) obtain the parameter values of the TVNN model using the prepared datasets (Section 8.2.2), and

3) visualize reviews using the CSCV method with the developed TVNN.

Here, N is the training dataset size. $\mathbf{Q}_i$ is a comment, and $d^{\mathbf{Q}_i}$ is a sentiment tag of $\mathbf{Q}_i$. $d^{\mathbf{Q}_i}$ is 1 if $\mathbf{Q}_i$ is positive and 0 if $\mathbf{Q}_i$ is negative.

8.2.1 TVNN This step constructs a TVNN from the WOSL, AWSL, CCSL, and output layer (Figure 19).

Definitions Before the explanation, we define several symbols. Let $\{w_i\}_{i=1}^v$ represent terms that appear in a text corpus of a dataset, v be the vocabulary size. In addition, we define the vocabulary index number of word w_i as $I(w_i)$. Therefore, $I(w_i) = i$. Let $\boldsymbol{w}_i^{em} \in \mathbb{R}^e$ be an embedding representation of word w_i, and the embedding matrix $\boldsymbol{W}^{em} \in \mathbb{R}^{v \times e}$ be $[\boldsymbol{w}_1^{emT}, \cdots, \boldsymbol{w}_v^{emT}]^T$, where e is the dimension size of word embeddings, and for each i, $\|\boldsymbol{w}_i^{em}\|_2 = 1$. $\boldsymbol{W}^{em}$ is the constant value given by the skip-gram method [39] and the prepared text corpus.

WOSL Given a comment $\mathbf{Q} = \{w_t^{\mathbf{Q}}\}_{t=1}^n$, this layer first converts the words $\{w_t^{\mathbf{Q}}\}_{t=1}^n$ to word sentiment representations $\{p_t^{\mathbf{Q}}\}_{t=1}^n$:

$$p_t^{\mathbf{Q}} := w^p_{I(w_t^{\mathbf{Q}})} \tag{49}$$

where $\boldsymbol{W}^p \in \mathbb{R}^v$ represents the original sentiment scores of words, and w_i^p is the ith element of $\boldsymbol{W}^p$. The w_i^p value corresponds to the original sentiment score of the word w_i.

Relation matrix between terms This layer, first, converts all the words in the comment to their respective word-level embeddings by $\{\boldsymbol{e}_t^{\mathbf{Q}}\}_{t=1}^n$ using $\boldsymbol{W}^{em}$, and then converts them to context representations $\{\boldsymbol{h}_t^{\mathbf{Q}}\}_{t=1}^n$ by using a bidirectional long short-term memory, namely, LSTM [50]:

$$\boldsymbol{h}_t^{\mathbf{Q}} = \mathrm{LSTM}(\boldsymbol{e}_t^{\mathbf{Q}}). \tag{50}$$

Using $\{\boldsymbol{h}_t^{\mathbf{Q}}\}_{t=1}^n$, we represent the relations between terms as follows.

$$g_{t,t'}^{\mathbf{Q}} = \tanh(\boldsymbol{h}_t^{\mathbf{Q}} \cdot \boldsymbol{h}_{t'}^{\mathbf{Q}}), \tag{51}$$

$$\boldsymbol{g}_t^{\mathbf{Q}} = [g_{t,1}^{\mathbf{Q}}, \cdots, g_{t,n}^{\mathbf{Q}}]^T, \tag{52}$$

$$\boldsymbol{R}^{\mathbf{Q}} = \mathrm{Softmax}([g_1^{\mathbf{Q}}, \cdots, g_n^{\mathbf{Q}}])^T \in \mathbb{R}^{n \times n} \tag{53}$$

Let $r_{t,t'}^{\mathbf{Q}}$ be the (t, t') element of $\boldsymbol{R}^{\mathbf{Q}}$, then, $r_{t,t'}^{\mathbf{Q}}$ represents the relation between terms $w_t^{\mathbf{Q}}$ and $w_{t'}^{\mathbf{I}}$.

AWSL This layer converts the original word-level sentiment representations $\{p_t^{\mathbf{Q}}\}_{t=1}^n$ to the aspect contextual word-level sentiment representations $\{c_t^{\mathbf{Q}}\}_{t=1}^n$:

$$c_t^{\mathbf{Q}} := \sum_{t=1}^{n} p_t^{\mathbf{Q}} r_{t,t'}^{\mathbf{Q}}$$

CCSL This layer converts the contextual word-level sentiment representations $\{c_t^{\mathbf{Q}}\}_{t=1}^n$ to the contextual concept-level sentiment representations $\{\boldsymbol{v}_t^{\mathbf{Q}}\}_{t=1}^n$:

$$\boldsymbol{v}_t^{\mathbf{Q}} = c_t^{\mathbf{Q}} \boldsymbol{b}_t^{\mathbf{Q}} \tag{54}$$

where $\boldsymbol{b}_t^{\mathbf{Q}} := \max(\mathrm{Softmax}(\boldsymbol{W}_c \boldsymbol{e}_t^{\mathbf{Q}} - t_c), 0)$, $\boldsymbol{v}_t^{\mathbf{Q}} \in \mathbb{R}^K$, $\boldsymbol{b}_t^{\mathbf{Q}} \in \mathbb{R}^K$, $t_c > 0$ is the hyper-parameter value, $\boldsymbol{W}_c \in \mathbb{R}^{K \times e}$ is the centroid vectors of $\{\boldsymbol{w}_i^{em}\}_{i=1}^v$ calculated with the spherical k-means method [26], the cluster number is K, and the (i, k) element of $\boldsymbol{b}_t^{\mathbf{Q}}$ represents the cluster weight of word $w_t^{\mathbf{Q}}$.

Output layer This layer converts contextual concept-level sentiment representations $\{\boldsymbol{v}_t^{\mathbf{Q}}\}_{t=1}^n$ to a predicted sentiment tag $y^{\mathbf{Q}} \in \{0(\text{negative}), 1(\text{positive})\}$:

$$\boldsymbol{a}^{\mathbf{Q}} = \text{Softmax}(\boldsymbol{W}^O \sum_{t=1}^{n} \boldsymbol{v}_t^{\mathbf{Q}}), y^{\mathbf{Q}} = \text{argmax } \boldsymbol{a}^{\mathbf{Q}}$$

where $\boldsymbol{W}^O \in \mathbb{R}^{2\times K}$ is the parameter value.

8.2.2 Learning Next, the parameter values of the *TVNN* model are obtained by using the backpropagation method according to Algorithm 9. The cross entropy between $\boldsymbol{a}^{\mathbf{Q}}$ and $d^{\mathbf{Q}}$ is used as a loss value. This is the simple version of the IP learning (= a little revised version of the LEXIL).

In this learning process, $\boldsymbol{W}^O$ is updated according to processes 4–6 in Algorithm 9 (i.e., *Update*) where $w_{i,j}^o$ is the (i,j) element of $\boldsymbol{W}^O$. This updating strategy makes $WOSL$ represent original word sentiment scores without violating the learning process after a sufficient number of updating iterations. This *Update* is important for the interpretability of the TVNN because the interpretability of the TVNN mainly depends on whether the WOSL accurately represent the original word-level sentiments.

Algorithm 9 Learning

1: **for** $i \leftarrow 1$ to v **do** $w_{p,i} \leftarrow 0$
2: **while** learning has not been finished **do**
3: Update $\boldsymbol{W}^O$, $\boldsymbol{W}_p$ and the LSTM cells using the backpropagation method.;
4: **for** $k \leftarrow 1$ to K **do**
5: **if** $w_{1,k}^o > 0$ **then** $w_{1,k}^o \leftarrow 0$;
6: **if** $w_{2,k}^o < 0$ **then** $w_{2,k}^o \leftarrow 0$;
7: **end while**

8.2.3 Conceptual Sentiment Cloud Visualization Using the TVNN, we can visualize the sentiment influence and aspect-based sentiment of a document in cluster units using a tag cloud method [28].

Sentiment Extraction

Sentiment influence From the term relation matrix of the TVNN, we can extract the sentiment influence of term $w_t^{\mathbf{Q}}$ to term $w_{t'}^{\mathbf{Q}}$ using $p_t^{\mathbf{Q}} r_{t,t'}^{\mathbf{Q}}$. Therefore, we can extract the sentiment influence between terms by summing these sentiment influences.

Aspect-based sentiment The TVNN can visualize what is positive or negative (i.e., aspect-based sentiment) in $\mathbf{Q}$ by using

$$CS'(w_i) := \sum_{t'=1}^{n} \sum_{t=1}^{n} p_t^{\mathbf{Q}} r_{t,t'}^{\mathbf{Q}} X(w_t'^{\mathbf{Q}}, w_i)$$

where $X(w', w) = \begin{cases} 1 & (w' = w) \\ 0 & (w' \neq w) \end{cases}$, for each w_i in the vocabulary.

Tag Cloud-based Visualization The proposed CSCV visualizes the above scores for the sentiment influence and aspect-based sentiment using the tag cloud approach [28]. In the CSCV, the aspect-based sentiments are displayed in concept cluster units as shown Figure 21. Each circle represents the aspect-based sentiments in concept cluster units. The CSCV colored word w_i as red if $CS'(w_i) > 0$ and blue if $CS'(w_i) < 0$, and determined the size of word w_i by $|CS'(w_i)|$. The concept cluster is determined using the spherical k-means method [26] with the word embedding representations.

In addition, the CSCV represents how each concept cluster are influenced by each term using the outside (gray area) of each circle as shown in Figure 22. In this sentiment influence representation, the CSCV colored the influence score of word w_i to concept cluster Ω using

$$CS(w_i, \Omega) := \sum_{w_j \in \Omega} \sum_{t'=1}^{n} \sum_{t=1}^{n} p_t^{\mathbf{Q}} r_{t,t'}^{\mathbf{Q}} X(w_t'^{\mathbf{Q}}, w_i) X(w_t^{\mathbf{Q}}, w_j)$$

. The CSCV colors as red if $CS(w_i, \Omega) > 0$ and blue if $CS(w_i, \Omega) < 0$, and determined the size of word w_i using $|CS(w_i, \Omega)|$.

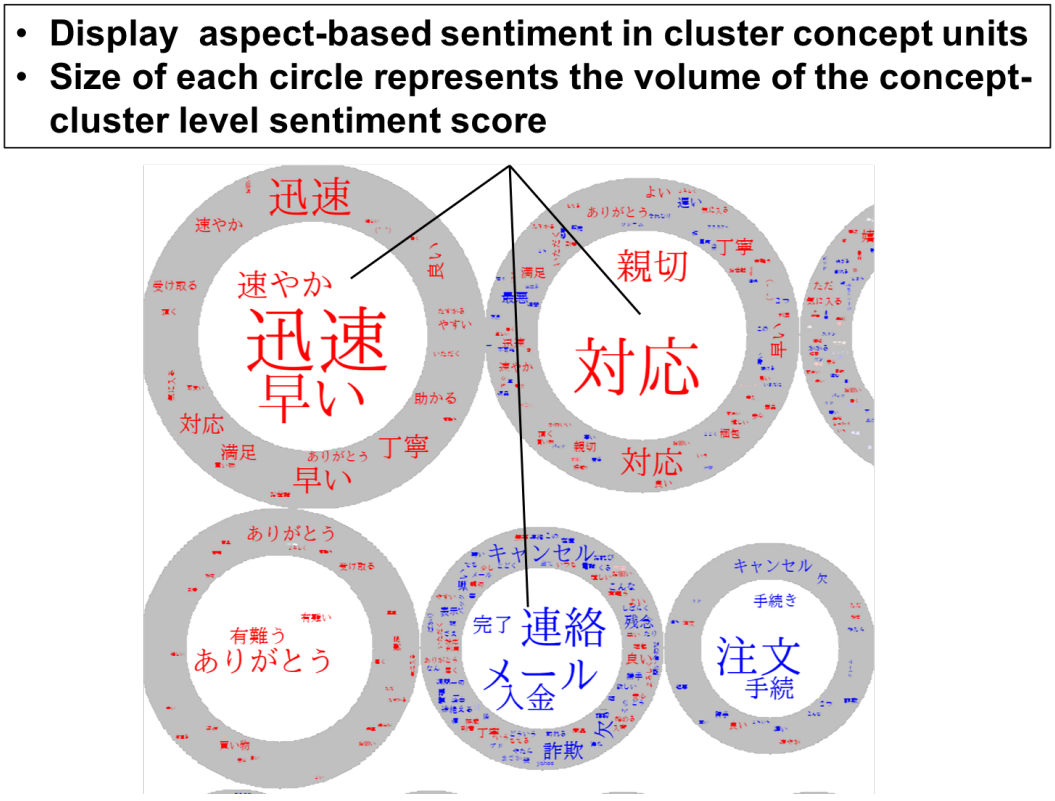

Figure 21: Aspect-based sentiment in the cluster units in the CSCV

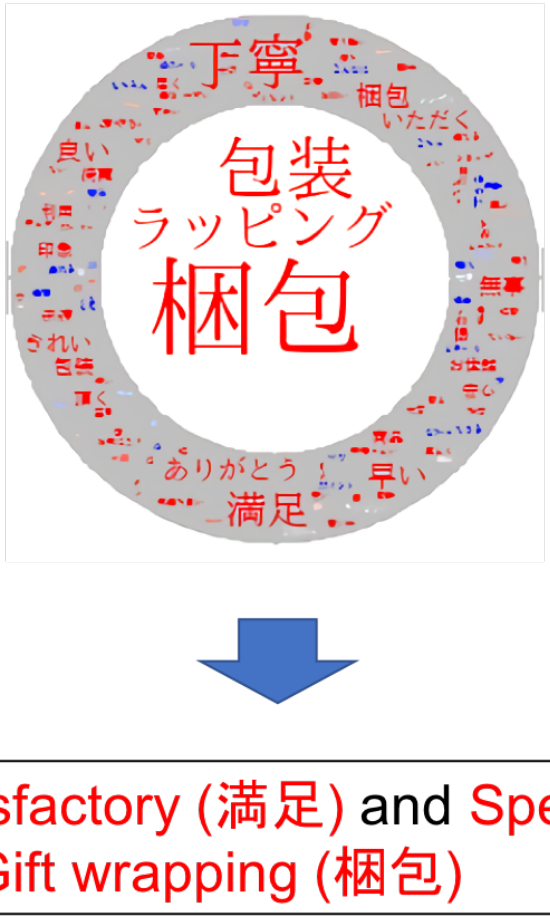

Figure 22: Sentiment influence to each cluster in the CSCV. The size of charactor reprsents the volume of sentiment. The inner circle represents what is posotive (red) or negative (blue), namely, aspect based sentiment. The outer ring represents the sentiment influence from other terms to each cluster.

8.3 Pre-Experimental Evaluation for TVNN

Before evaluating the text-visualization quality of the CSCV, this section briefly tests the TVNN from the following two aspects:

1. whether the WOSL accurately represents the original word-level sentiments or not (Section 8.3.2), and

2. term relation matrix accurately produced the contextual word-level sentiments in the AWSL using the WOSL (Section 8.3.3).

We conducted this evaluation because the validity in the WOSL and the validity in the term relation matrix directly lead to the text-visualization quality in the CSCV.

8.3.1 Model development To test the validity of the TVNN, we developed eight types of TVNN using following eight textual datasets including comments and their sentiment tags.

Dataset *Current economy watchers survey (EcoReview I).* This dataset included Japanse comments for the current economic trend and their positive or negative sentiment tags [2]. This dataset was collected by workers closely related to the regional economy. We used oldest 20,000 positive comments and oldest 20,000 negative comments as the training dataset, oldest 2,000 positive and oldest 2,000 negative comments of the remaining comments as the validation dataset, and newest 4,000 positive and newest 4,000 negative comments of the remaining comments as the test dataset. The vocabulary size v was 8,071.

Future economy watchers survey (EcoReview II). This dataset included Japanse comments[2] for the future economic trend between 2002 and 2017 and their sentiment tags. This dataset included 26,000 positive comments and 26,000 negative comments. We used oldest 35,000 positive comments and oldest 35,000 negative comments as the training dataset, oldest 2,000 positive and oldest 2,000 negative comments of the remaining comments as the validation dataset, and newest 4,000 positive and newest 4,000 negative comments of the remaining comments as the test dataset. The vocabulary size v was 11,130.

Finance review. This dataset included the comments for each stock and their buy (positive) or sell (negative) attitude tags, extracted from Yahoo Finance microblogs[3] between September 2015. We used the oldest 40,000 posts (30,612 positive posts and 9,388 negative posts) as the training dataset, the oldest 5,000 posts from the remaining posts (3,387 positive posts and 1,613 negative posts) as the validation dataset, and the newest 10,000 posts (7,538 positive posts and 2,462 negative posts) as the test dataset. The vocabulary size v was 33,08.

Shop review. In developing the TVNN, we used the customer reviews including comments and their ratings (1: very bad, 2: bad, 3: neutral, 4: good 5: very good) collected from Reviews in Yahoo! Shopping [4] between 2015. We considered reviews with 1 or 2 as negative, and those with 4 or 5 as positive. The rating distribution is as shown in Table 1. The vocabulary size v was 81,130.

Amazon product reviews. This dataset contains product reviews including comments, and their ratings (between 1–5) collected from Amazon [5]. We considered reviews with 1 or 2 as negative, and those with 4 or 5 as positive. In this evaluation, we used reviews for books (Book review), those for movies & TV (Movies & TV review), and those for electronics (Electronics review), respectively. In the Book dataset, we used the oldest 280,000 positive reviews and 280,000 negative reviews as the training dataset, the oldest 20,000 positive reviews and 20,000 negative reviews from the remaining reviews as the validation dataset, and the newest 50,000 positive reviews and 50,000 negative reviews as the test dataset. we used the oldest 80,000 positive reviews and 80,000 negative reviews as the training dataset, the oldest 5,000 positive reviews and 5,000 negative reviews from the remaining reviews as the validation dataset, and the newest 10,000 positive reviews and 10,000 negative reviews as the test dataset. In the Movies & TV dataset, We used the oldest 70,000 positive posts and 70,000 negative posts as the training dataset, the oldest 5,000 positive posts and 5,000 negative posts from the remaining posts as the

[2] https://www5.cao.go.jp/keizai3/watcher-e/index-e.html
[3] http://textream.yahoo.co.jp
[4] https://shopping.yahoo.co.jp
[5] http://jmcauley.ucsd.edu/data/amazon/

validation dataset, and the newest 10,000 positive posts and 10,000 negative posts as the test dataset.

Sentiment 140. This dataset contains 800,000 positive tweets and 800,000 negative tweets [6]. We used the first 650,000 positive tweets and 650,000 negative tweets as the training dataset, the next 50,000 positive tweets and 50,000 negative tweets as the validation dataset, and the remain 100,000 positive tweets and 100,000 negative tweets as the test dataset.

In the above, the EcoReviews and the Yahoo review were Japanese textual datasets, and the others were English textual datasets. We used them to test whether our approach can be used without regard to the language or genre. For each dataset, we split it into the training, validation, and test datasets, as summarized in Table 13.

Table 13: Dataset Organization

	EcoReview I	EcoReview II	Finance Review	Shop Review	Amazon product review			Sentiment 140
					Book	Movie & TV	Electronics	
training dataset								
number of positive comments	20,000	35,000	30,612	50,000	280,000	70,000	80,000	650,000
number of negative comments	20,000	35,000	9,388	50,000	280,000	70,000	80,000	650,000
validation dataset								
number of positive comments	2,000	2,000	3,387	10,000	20,000	5,000	5,000	50,000
number of negative comments	2,000	2,000	1,613	10,000	20,000	5,000	5,000	50,000
test dataset								
number of positive comments	4,000	4,000	7,538	20,000	50,000	10,000	10,000	100,000
number of negative comments	4,000	4,000	2,462	20,000	50,000	10,000	10,000	100,000
vocabulary size v	8,071	11,130	33,080	80,901	331,987	148,494	87,213	71,316
word polarity list								
number of positive words	411	337	469	609	2,754	1,063	1,822	1,843
number of negative words	437	387	402	537	1,267	591	920	947

Other settings The other experimental settings used to develop the TVNN were as follows: We used word embedding matrix $\boldsymbol{W}^{em}$ calculated by the the skip-gram method (window size = 5) [39] using each textual dataset. We set the dimensions of the LSTM cells' hidden and embedding vectors to 200, the epoch to 50 with early stopping, the value of K to 100, the value of t_c to 0.01, and the mini-batch size to 64, We used stratified sampling [69] to analyze imbalanced data, and the Adam optimizer [8], and the dropout [55] method (rate = 0.5).

8.3.2 Validity in the WOSL

Experimental Setting We evaluated the validity of the WOSL in terms of whether the developed TVNN could accurately predict the sentiment tags of terms in a manually created polarity word list using the WOSL.

Word polarity list We used four-word polarity lists including the Economic word polarity list, the financial micro-blog word polarity list, shop review word polarity list, and Lexicoder Sentiment Dictionary (LEX word list) [67], for this evaluation. The Economic word-polarity list included 411 positive Japanese words and 437 negative Japanese words for economics. The financial micro-blog word polarity list included 469 positive Japanese words and 402 negative

[6] https://www.kaggle.com/kazanova/sentiment140

Japanese words for investments. The shop review word polarity list included positive or negative customer review oriented sentiment tags of more than 2000 words. The LEX word list included 2,858 positive words and 1,709 negative words [7].

Prediction of original word polarity we predicted the positive or negative sentiment tags of the words in the word polarity lists using the TVNN and the other comparative method: LR, point-wise mutual information (PMI), LFW, and SONN models.

TVNN. When we used the TVNNs, we predicted word w_i as positive when $w_{p,i} > 0$ and as negative when $w_{p,i} < 0$.].

Other methods. When we used the LR, we assigned each word to the corresponding weight vector value of the LR model as its original sentiment score. When we used the PMI, FLW, and SONN, we calculated the original word-level sentiment score of each terms using the training and validation datasets with the methods described in [40], [58], [34], respectively. After that, we predicted each word as positive (negative) when its score was positive (negative.)

Evaluation Setting After the above predictions, we compared the prediction results in terms of the macro F_1 score. In this evaluation, if we used the EcoReview I or II in the training process, we used the economic word polarity list, if we used the Yahoo dataset, we used the Yahoo word polarity list, if we used the shop review dataset, we used the shop review word polarity list, and if we used the other datasets, we used the LEX word list. Moreover, we used only the terms which appeared more than five times in the training dataset and not used in the *Init* process. Table 1 represents the number of words used in evaluating the CSNN developed with each dataset.

Result Table 14 represents the results, showing that the TVNN outperformed the others, and the values of the WOSL were sufficiently valid.

Table 14: F_1 score results for original sentiment evaluation

	EcoReview I	EcoReview II	Finance Review	Shopping Review	Amazon product review			Sentiment 140	average
					Book	Movie & TV	Electronics		
LR	0.731	0.773	0.728	0.747	0.628	0.628	0.657	0.728	0.697
PMI	0.754	0.757	0.796	0.785	0.6817	0.664	0.692	0.733	0.729
LFW	0.715	0.740	0.766	0.704	0.640	0.600	0.681	0.725	0.696
SONN	0.719	0.748	0.733	0.767	0.643	0.636	0.690	0.705	0.699
TVNN	0.825	0.815	0.810	0.792	0.704	0.685	0.731	0.735	**0.762**

8.3.3 Validity in AWSL We evaluated the validity of the AWSL in terms of whether the developed TVNN could accurately predict the document-level sentiment tags of reviews in each test dataset using the AWSL.

Comparison Method We compared the results of the TVNN with the results of the following comparative methods: Logistic regression model (LR), a Bidirectional recurrent neural network model with LSTM cells (RNN), and convolutional Neural Network Model (CNN) [30], a Logistic fixed weight model (LFW) [58], and the Sentiment-oriented NN (SONN) [34]. The above models were developed with each training and validation datasets.

[7] available at `http://quanteda.io/reference/data_dictionary_LSD2015.html`

We set the dimensions of the RNNs' hidden and embedding vectors to 200, the epoch to 50 with early stopping, the value of K to $[100, 500, 1000]$, the value of t_c to $\frac{1}{K}$, and the mini-batch size to 64. The hyper-parameters were determined using the validation data.

Result The macro F_1 score results for each method are summarized in Table 15, showing that the RNN worked in most high performance; however, the TVNN significantly outperformed the other methods including CNN in the average score between datasets (p-value ¡ 0.05), This result demonstrated that the values of the AWSL were sufficiently valid. From the validities in the WOSL and AWSL, we could demonstrate that the values of the term relation matrix were sufficiently valid.

Table 15: F_1 score results for predictability evaluation

	EcoReview I	EcoReview II	Finance Review	Shopping Review	Amazon product review			Sentiment 140	average
					Book	Movie & TV	Electronics		
LR	0.878	0.879	0.741	0.956	0.915	0.871	0.856	0.785	0.860
LFW	0.876	0.840	0.751	0.951	0.912	0.781	0.819	0.745	0.834
SONN	0.863	0.876	0.717	0.957	0.919	0.875	0.853	0.776	0.855
CNN	0.894	0.911	0.757	0.968	0.951	0.912	0.916	0.820	0.891
RNN	**0.922**	0.932	0.749	**0.971**	**0.960**	**0.925**	**0.936**	**0.837**	**0.904**
TVNN	0.915	**0.936**	**0.766**	0.968	0.954	0.911	0.926	0.829	0.901

8.4 Experimental Evaluation for CSCV

Using real user response. We tested the quality of the images produced from the CSCV method in terms of the user-friendliness: *response accuracy* and *response time*.

8.4.1 Dataset

Test Reviews To test the CSCV method, we prepared the following two types of review text datasets: short review dataset and long review dataset using the customer reviews extracted from the Yahoo Shopping Service. For each set of reviews in the short and long review datasets, we produced the image summarizing the set of reviews using the CSCV. In this process, we used the TVNN developed with the shop review dataset.

1. Short review dataset This dataset included 180 sets of reviews. Each set of reviews included 30 reviews for a certain shop that were not included in the training dataset. Shops were selected in a form that the mean value of the ratings is similar to 3.

2. Long review dataset This dataset included 200 sets of reviews. Each set of reviews included 100 reviews for a certain shop that were not included in the training dataset. The shop was selected in a form that the mean value of the ratings is similar to 3.

User Response Collection To evaluate the proposed method, we collected the user response data in answering the questions using review texts or review images using the crowd sourcing.

First, for each set of review tests and its image output, we prepared three questions about the sentiment as shown in Figures 23 and 24:

In this question, we make users select the best choice for the following four types of choices: As

After reading the reviews for shop X in this link `www.example`, select the most appropriate answer from the following choices

1. For contact and response, this shop is good.
2. For contact and response, this shop is bad.
3. For contact and response, this shop is normal.
4. We can't judge.

Figure 23: Question form for textual review

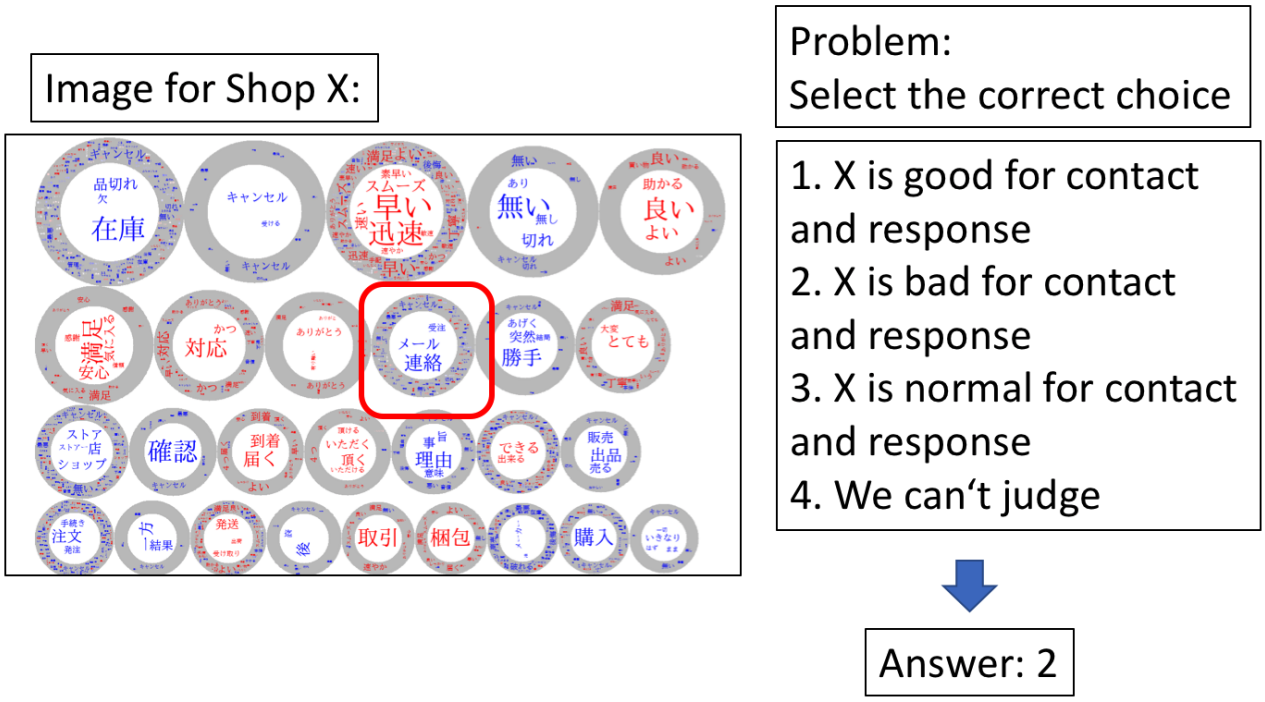

Figure 24: Question form for review image

X, we manually prepared the most important 20 patterns (e.g., payment, price, speed in delivery, insurance, taste, size, appearance, etc.) from the practical perspectives. Three annotators answered for each question. In this annotation, annotators who answered the questions about the image and those who answered the questions about the review texts were different. We decided this setting considering fairness.

After excluding the tags that were not 1: good or 2: bad, we tagged the most frequent answer as the gold answer tag of the question. In this dataset development, we excluded the questions whose three answers were all 3 or 4 and the questions in which the number of answer 1 and that

of answer 2 was the same.

In total, we collected 129 negative sentiment tags and 368 positive sentiment tags were included in the short review dataset, and 230 negative sentiment tags 119 positive sentiment tags were included in the long review dataset from the answers to the review comments.

Agreement Check To evaluate the dataset quality, we tested how accurately the annotators could answer the questions using the gold sentiment tags. The average macro F_1 score results between three annotators were 0.856 in short reviews and 0.823 in long reviews.

8.4.2 Evaluation We then evaluated the image quality from the following two evaluation basis for the user-friendliness: response quality and response speed.

Response quality In this evaluation, we evaluated the review images using the agreement degree between the answer results for the review texts and those for the review images are agreed. We considered the answer results from the review texts as the answer tags, and we evaluated the review image quality in terms of how much the answer results from the review images agreed to the answer tags. In deciding the answers from each image of reviews, we decided the most frequent answer for each question as the answer.

We used the macro F_1 score for the evaluation basis.

Baseline Method To evaluate the proposed method, we compared the results of the CSCV with those of the following baseline method. The baseline method answered all the questions for a set of reviews as 1: good if the mean rating score of the reviews was larger than 3, and answered as 2: bad in the other case.

Response speed In this evaluation, we compared the average response time for answering the questions using the review texts (baseline) and those using the review images.

8.4.3 Result

Response quality Table 16 summarizes the results showing that the proposed method outperformed the baseline method. In addition, we also analyzed the results for only the gold

Table 16: Response quality evaluation result

	Short Reviews	Long Reviews	Total
Baseline	0.510	0.540	0.547
CSCV (proposed)	**0.660**	**0.598**	**0.695**

sentiment tags of questions in which the number of positive (1) or negative (2) annotated tags were more than 2 (i.e., high-quality gold tags). As for the high-quality gold sentiment tags, 321 negative sentiment tags and 168 positive sentiment tags were included in the short review dataset, and 122 negative sentiment tags 75 positive sentiment tags were included in the long review dataset. Table 17 summarizes the results showing that the proposed method outperformed the baseline method.

Table 17: Response quality evaluation result for high-quality tags

	Short Reviews	Long Reviews	Total
Baseline	0.447	0.578	0.532
CSCV (proposed)	**0.680**	**0.652**	**0.741**

Response speed Table 18 show the results, showing that the response time using the review images was shorter than the time using the review comments. These results demonstrated

Table 18: Response speed evaluation result

	Short Reviews	Long Reviews
Baseline (text)	116 sec	156 sec
CSCV (image)	**92 sec**	**134sec**

that the images produced from the proposed CSCV was sufficiently user-friendly than baseline methods.

8.5 Text-Visualization Example

Figures 25 and 26 are the text viausalztiaon examples by CSCV.

Figure 25 show the image example for the clothing shop X (anonymous) produced by the CSCV method. From this image, from the upper right circle, we can catch-up that this shop is good in the product because many opinions say that it is cute, nice, and favorite; however, from the circle in second row, this shop is bad in a order manner. Therefore, this shop should also improve the order manner.

Figure 26 shows the image example for the food shop Y (anonymous) produced by the CSCV method. From this image, we could catch-up that this shop is good in taste from the circle in upper left; however, from the circle in lower left, we can see that this shop has a little claim for the fresh, reliable, and safe of food. Therefore, this shop should improve the management of food.

8.6 Related work

Previous works have been done for aspect-based sentiment analysis [38, 61, 65]. However, they need specific knowledge or sentiment tags for the aspect-based sentiment analysis. This is not practical. Unlike these works, we can use the proposed method just with the documents and their sentiment tags. There have been studies about assigning original sentiment scores to words automatically [33,34,40,58]. However, the proposed TVNN was able to assign original sentiments to words more accurately than these methods. Many studies have assigned contextual word-level sentiments considering contexts in a document [31, 35, 36, 49, 62, 63]. However, they require specific knowledge of contexts. By contrast with these methods, the TVNN does not need any such specific knowledge.

Many previous works have visualized the sentiments [4, 13, 14, 29, 42, 68] and contents [5–7, 12, 28] of documents. However, there have been little works for visualizing both the aspect-based sentiment and sentiment influence at the same time.

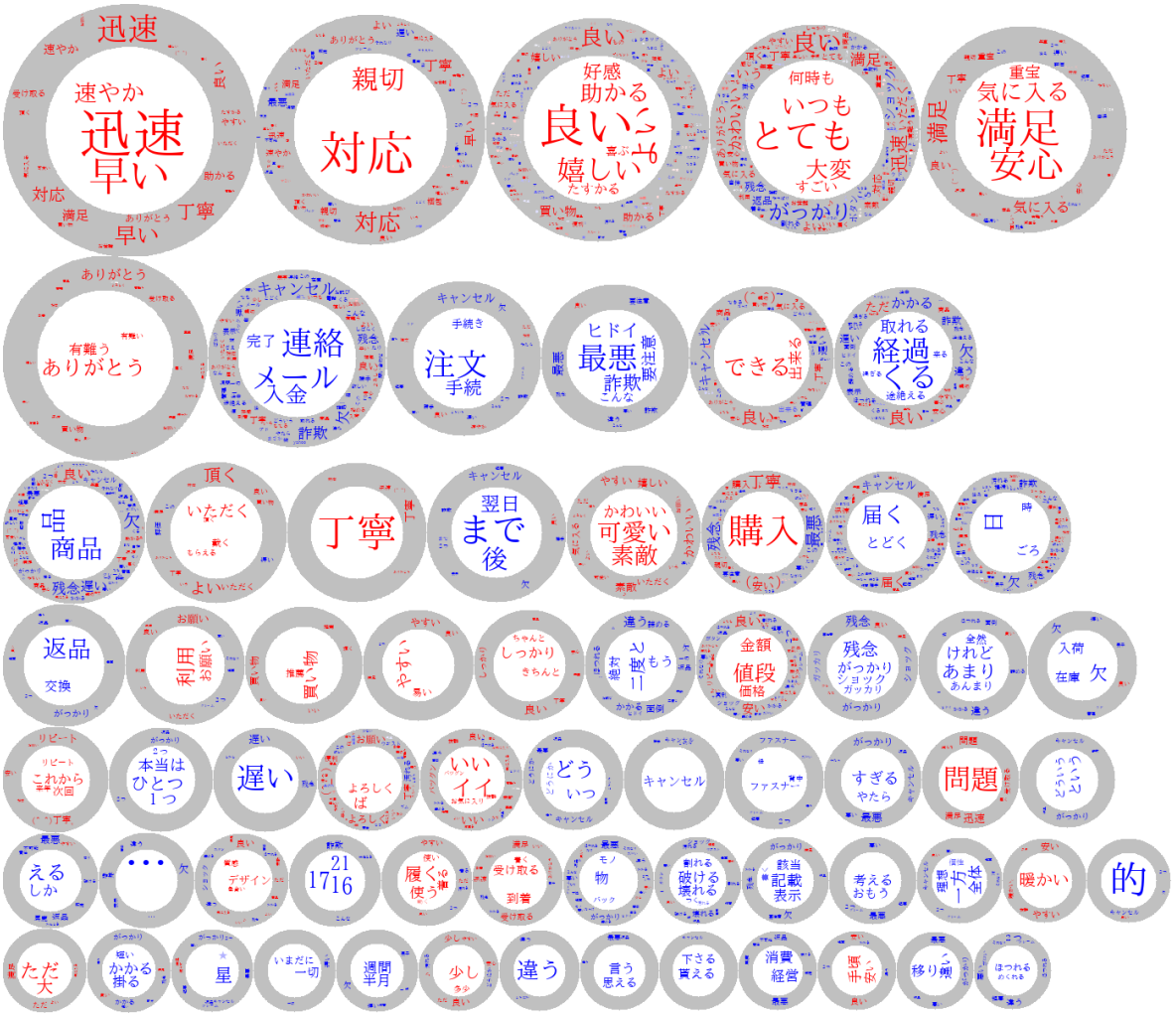

Figure 25: Text Visualization Example for reviews in the clothing shop X

8.7 Conclusion

This chapter proposed a novel text-visualization method for summarizing reviews called the CSCV as an application of our research. CSCV displays both the aspect-based sentiments and the sentiment influence of each term to each concept cluster in a user-friendly way. The CSCV can be realized using the interpretable neural network model called TVNN. We can use CSCV only review texts and their sentiment tags. We do not need any other knowledge such as aspect-based sentiment tags. This is the practical point in the CSCV.

Using real textual datasets and real user response, we demonstrated the usefulness of the CSCV. The proposed CSCV outperformed the baseline methods in both response quality and response speed. Using the CSCV, we could catch-up on the contents of long reviews faster than reading the review texts. This work should have a high impact on the industry. Moreover, this application of our learning theory into the development of *Conceptual Sentiment Cloud*

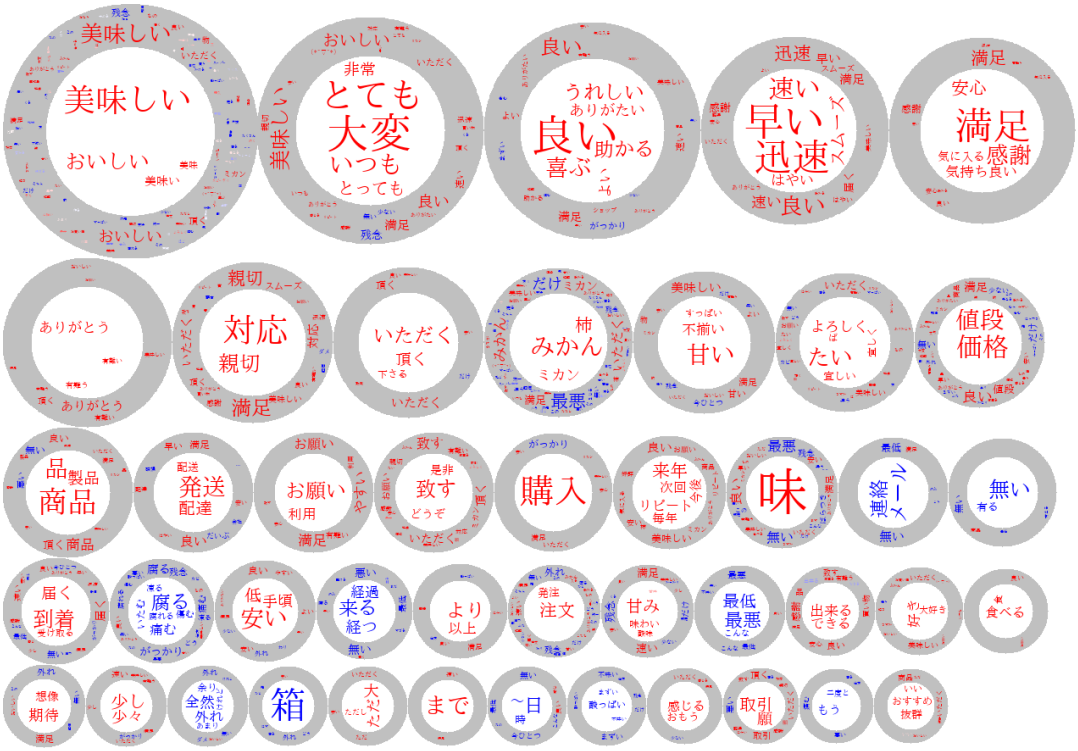

Figure 26: Text Visualization Example for reviews in the food shop Y

Visualization demonstrates that our study can be applied to several real-world issues.

It should be noted that this trial is incomplete and ongoing. First, the comparison with other text visualization strategies such as text visualization using word-level original sentiments or word frequency has not been done. Second, discussion with users has not been done. Therefore, the modification of the text visualization design considering the real demands of users is one of the important issues in this study.

Part IV

Conclusion and Appendix

Chapter 9

Conclusion and Future Work

9.1 Conclusion

This study addresses the issue of developing interpretable neural networks that can analyze sentiment by explaining its prediction results in a form that humans feel natural and agreeable. To address this problem, we first derive and discuss the basic learning theory, conditions, and assumptions that are required to realize the interpretability of layers in NNs. We then practically develop four types of interpretable neural networks, namely, SINN, SSNN, GINN, and CSNN, and experimentally evaluate them using several datasets including Japanese and English datasets. The interpretable NNs developed with the proposed approaches had both the high prediction ability and high explanation ability. They outperformed some DNNs in a document-level sentiment analysis task, whereas the interpretability of each layer in each of them was sufficiently valid.

As an application of this study, we propose a of novel text-visualization framework called *Conceptual Sentiment Cloud Visualization (CSCV)*. CSCV should be valuable in a situation where users want to catch up a large volume of reviews for a certain product or shop.

9.2 Future Work

There has been some limitation in the proposed approach. First, our approach requires a large volume of document-level sentiment tags for a specific domain. Second, our approach can not consider the domain of documents. As a result, a model developed with LEXIL or JSP learning can address only single domain texts. Considering the above limitations, the extension of CSNN or SINN into the multi-domain sentiment analysis model is possible as one of the future directions. This type of extension has two strong merits. The first merit is that this type of model can address multi-domain types of texts. In addition, it can be possible that we can develop a fine model in a situation where we have several small datasets and the total volume of them is large. As for another direction, the usage of unlabeled text corpus in a semi-supervised way can be possible.

In addition, it should be noted that this study is still theoretical and basic. Therefore, the application of our approach to more industrial problems or other types of classification tasks is one of the future works.

Acknowledgement

This work was supported in part by JSPS KAKENHI Grant Number JP17J04768. We thank the five financial professionals including Yuichi Sugihara (Nomura Asset Management) and the other two anonymous professionals, who support this thesis in a form of personal communication. We thank the Kota Tsubouchi and Tatuso Yamashita, who are the co-researchers in Yahoo Japan Corporation.

References

[1] L. Arras, G. Montavon, K. R. Muller, and W. Samek, *Explaining recurrent neural network predictions in sentiment analysis*, in EMNLP Workshop, 2017.

[2] S. Bach, A. Binder, G. Montavon, F. Klauschen, K. R. Muller, and W. Samek, *On pixel-wise explanations for nonlinear classifier decisions by layer-wise relevance propagation*, PLOS ONE, 10 (2017), pp. 1–46.

[3] A. Bai, H. Hammer, A. Yazidi, and P. Engelstad, *Constructing sentiment lexicons in norwegian from a large text corpus*, in 2014 IEEE 17th International Conference on Computational Science and Engineering, Dec 2014, pp. 231–237.

[4] A. Brew, D. Greene, D. Archambault, and P. Cunningham, *Deriving insights from national happiness indices*, in IEEE ICDMW 2011, 2011, pp. 53–60.

[5] M. Burch, S. Lohmann, F. Beck, N. Rodriguez, L. DiSilvestro, and D. Weiskopf, *Radcloud: visualizing multiple texts with merged word clouds*, in IV 2014, 2014, pp. 108–113.

[6] M. Burch, S. Lohmann, D. Pompe, and D. Weiskopf, *Prefix tag clouds*, in IV 2013, 2013, pp. 45–50.

[7] C. Collins, F. B. Viegas, and M. Wattenberg, *Parallel tag clouds to explore and analyze faceted text corpora*, in IEEE VAST 2009, 2009, pp. 91–98.

[8] J. L. B. D. P. Kingma, *Adam: A method for stochastic optimization*, arXiv:1412.6980, (2014).

[9] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, *BERT: Pre-training of deep bidirectional transformers for language understanding*, in Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Minneapolis, Minnesota, 2019, Association for Computational Linguistics, pp. 4171–4186.

[10] Y. Dong, H. Su, J. Zhu, and B. Zhang, *Improving interpretability of deep neural networks with semantic information*, in CVPR, 2017.

[11] F. Fancellu, A. Lopez, and B. Webber, *Neural networks for negation scope detection*, in ACL 2016, 2016.

[12] P. Gambette and J. Veronis, *Gam: Visualising a text with a tree cloud*, in Classification as a Tool for Research, Springer, Berlin, 2010, pp. 561–569.

[13] E. Guzman, *Visualizing emotions in software development projects*, IEEE VISSOFT 2013, (2013), pp. 1–4.
[14] M. C. Hao, C. Rohrdantz, H. Janetzko, D. A. Keim, and etal, *Visual sentiment analysis of customer feedback streams using geo temporal term associations*, in Information Visualization, 2013.
[15] Y. Hechtlinger, *Interpretation of prediction models using the input gradient*, in arXiv:1611.07634, 2016.
[16] Q. Hu, J. Zhou, Q. Chen, and L. He, *Snnn: Promoting word sentiment and negation in neural sentiment classification*, in AAAI 2018, 2018.
[17] C. Hutto and E. Gilbert, *Vader: A parsimonious rule-based model for sentiment analysis of social media text*, in ICWSM-14, 2014.
[18] D. Ikeda, H. Takamura, and M. Okumura, *Learning to shift the polarity of words for sentiment classification*, in IJCNLP 2008, 2008, pp. 50–57.
[19] R. Ito, K. Izumi, H. Sakaji, and S. Suda, *Lexicon creation for financial sentiment analysis using network embedding*, Journal of Mathematical Finance, 7 (2017), pp. 896–907.
[20] T. Ito, K. Izumi, K. Tsubouchi, and T. Yamashita, *Polarity propagation of financial terms for market trend analyses using news articles*, in 2016 IEEE Congress on Evolutionary Computation (CEC), July 2016, pp. 3477–3482.
[21] T. Ito, H. Sakaji, K. Tsubouchi, K. Izumi, and T. Yamashita, *Text-visualizing neural network model: Understanding online financial textual data*, in PAKDD 2018, 2018.
[22] T. Ito, K. Tsubouchi, H. Sakaji, T. Yamashita, and K. Izumi, *Csnn: Contextual sentiment neural network*, in IEEE ICDM 2019, 2019.
[23] T. Ito, K. Tsubouchi, H. Sakaji, T. Yamashita, and K. Izumi, *Ssnn: Sentiment shift neural network*, in SDM 2020, 2020.
[24] T. Ito, K. Tsubouchi, H. Sakaji, T. Yamashita, and K. Izumi, *Word-level contextual sentiment analysis with interpretability*, in AAAI 2020, 2020.
[25] S. Jain and B. C. Wallace, *Attention is not Explanation*, in Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Minneapolis, Minnesota, June 2019, Association for Computational Linguistics, pp. 3543–3556.
[26] M. K. K. Hornik, I. Feinerer and C. Buchta, *Spherical k-means clustering*, Journal of Statistical Software, 50 (2012), pp. 1–22.
[27] S. Karen, V. Andrea, and A. Zisserman, *Deep inside convolutional networks: Visualising image classification models and saliency maps*, arXiv:1312.6034, (2013).
[28] O. Kaser and D. Lemire, *Tag-cloud: drawing algorithms for cloud visualization*, in arXiv: cs/0703109, 2007.
[29] R. Kempter, V. Sintsova, C. Musat, and P. Pu, *Emotion watch: visualizing fine-grained emotions in event-related tweets*, in ICWSM 2014, 2014.
[30] Y. Kim, *Convolutional neural networks for sentence classification*, in EMNLP 2014, 2014.
[31] S. Kiritchenko and S. M. Mohammad, *The effect of negators, modals, and degree adverbs on sentiment composition*, in NAACL-HLT 2016, 2016, pp. 43–52.
[32] T. Kudo, K. Yamamoto, and Y. Matsumoto, *Applying conditional random fields to japanese morphological analysis*, in EMNLP, 2004, pp. 230–237.
[33] K. Labille, S. Alfarhood, and S. Gauch, *Estimating sentiment via probability and information theory*, in KDIR 2016, 2016, pp. 121–129.
[34] Q. Li, *Learning stock market sentiment lexicon and sentiment-oriented word vector from stocktwits*, in CoNLL 2017, 2017, pp. 301–310.

[35] S. Li, Z. Wang, S. Y. M. Lee, and C.-R. Huang, *Sentiment classification with polarity shifting detection*, in IALP 2013, 2013, pp. 129–132.
[36] S. Li, S. Yat, M. Lee, Y. Chen, C. R. Huang, and G. Wang, *Sentiment classification and polarity shifting*, in COLING 2010, 2010, pp. 635–643.
[37] T. Loughran and B. McDonald.
[38] Y. Ma, H. Peng, and E. Cambria, *Targeted aspect-based sentiment analysis via embedding commonsense knowledge into an attentive lstm*, in AAAI 2018, 2018.
[39] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, *Distributed representations of words and phrases and their compositionality*, in NIPS 2013, 2013.
[40] S. Mohammad, S. Kiritchenko, and X. D. Zhu, *Nrc-canada: Building the state-of-the-art in sentiment analysis of tweets*, in SemEval-2013, 2013.
[41] P. Nakov, S. Rosenthal, . Kozareva, V. Stoyanov, A. Ritter, and T. Wilson, *Semeval-2013 task 2: Sentiment analysis in twitter*, in SemEval 2013, 2013.
[42] C. Nan and W. Cui, *Introduction to text visualization*, in Atlantis Briefs in Artificial Intelligence, 2016.
[43] K. A. Nguyen, S. Schulte im Walde, and N. T. Vu, *Integrating distributional lexical contrast into word embeddings for antonym-synonym distinction*, in Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, 2016, pp. 454–459.
[44] Z. Quanshi, Y. N. Wu, and S. C. Zhu, *Interpretable convolutional neural networks*, in CVPR 2018, 2018.
[45] K. Ravi and V. Ravi, *A survey on opinion mining and sentiment analysis: Tasks, approaches and applications*, Knowledge-Based Systems, 89 (2015), pp. 14–46.
[46] M. T. Ribeiro, S. Singh, and C. Guestrin, *"why should i trust you?" explaining the predictions of any classifier*, in KDD, 2016.
[47] S. Rosenthal, P. Nakov, A. Ritter, and V. Stoyanov, in SemEval 2014, 2014.
[48] M. Schulder, M. Wiegand, J. Ruppenhofer, and S. Köser, *Introducing a Lexicon of Verbal Polarity Shifters for English*, in Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC), N. Calzolari (Conference chair), K. Choukri, C. Cieri, T. Declerck, S. Goggi, K. Hasida, H. Isahara, B. Maegaard, J. Mariani, H. Mazo, A. Moreno, J. Odijk, S. Piperidis, and T. Tokunaga, eds., vol. 1, Miyazaki, Japan, May 2018, European Language Resources Association (ELRA), pp. 1393–1397.
[49] M. Schulder, M. Wiegand, J. Ruppenhofer, and B. Roth, *Towards bootstrapping a polarity shifter lexicon using linguistic features*, in IJCNLP 2017, 2017, pp. 624–633.
[50] M. Schuster and K. Paliwal, *Bidirectional recurrent neural networks*, IEEE Transactions on Signal Processing, 45 (1997), pp. 2673–2681.
[51] S. Serrano and N. A. Smith, *Is attention interpretable?*, in Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, July 2019, Association for Computational Linguistics, pp. 2931–2951.
[52] A. Shrikumar, P. Greenside, and A. Kundaje, *Learning important features through propagating activation differences*, in ICML, 2017.
[53] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Ng, and C. Potts, *Recursive deep models for semantic compositionality over a sentiment treebank*, in Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, Seattle, Washington, USA, Oct. 2013, Association for Computational Linguistics, pp. 1631–1642.
[54] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. A. Riedmiller, in Striving for simplicity: The all convolutional net, ICLR Workshop, 2015.
[55] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov,

Dropout: A simple way to prevent neural networks from overfitting, JMLR, 15 (2014), pp. 1929–1958.

[56] M. Sundararajan, A. Taly, and Q. Yan, *Axiomatic attribution for deep networks*, in ICML, 2017.

[57] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, *Attention is all you need*, in NIPS 2017, 2016.

[58] D. T. Vo and Y. Zhang, *Don't count, predict! an automatic approach to learning sentiment lexicons for short text*, in ACL 2016, 2016, pp. 219–224.

[59] C.-J. Wang, M.-F. Tsai, T. Liu, and C.-T. Chang, *Financial sentiment analysis for risk prediction*, in Proceedings of the Sixth International Joint Conference on Natural Language Processing, Oct. 2013, pp. 802–808.

[60] W. Wang, N. Yang, F. Wei, B. Chang, and M. Zhou, *Gated self-matching networks for reading comprehension and question answering*, in ACL 2017, 2017.

[61] Y. Wang, M. Huang, x. zhu, and L. Zhao, *Attention-based lstm for aspect-level sentiment classification*, in EMNLP 2016, 2016.

[62] T. Wilson, J. Wiebe, and P. Hoffman, *Recognizing contextual polarity in phrase level sentiment analysis*, in EMNLP 2005, 2005, pp. 347–354.

[63] R. Xia, F. Xu, J. Yu, Y. Qi, and E. Cambria, *Polarity shift detection, elimination and ensemble: A three-stage model for document-level sentiment analysis*, Information Processing and Management, 52 (2016), pp. 36–45.

[64] Q. Z. X. H. Y. Zou, T. Gui, *A lexicon-based supervised attention model for neural sentiment analysis*, in COLING 2018, 2018.

[65] T. Yanase, K. Yanai, M. Sato, T. Miyoshi, and Y. Niwa, *Neural and syntactic models of entity-attribute relationship for aspect-based sentiment analysis*, in SemEval-2016, 2016.

[66] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy, *Hierarchical attention networks for document classification*, in NAACL 2016, 2016.

[67] L. Young and S. Soroka, *Affective news: The automated coding of sentiment in political texts*, Political Communication, 29 (2012), pp. 205–231.

[68] J. Zhao, L. Gou, F. Wang, and M. Zhou, *Pearl: an interactive visual analytic tool for understanding personal emotion style derived from social media*, in IEEE VAST 2014, 2014, pp. 203–212.

[69] P. Zhao and T. Zhang, *Accelerating minibatch stochastic gradient descent using stratified sampling*, arXiv:1405.3080v1, (2014).