A Joint Learning Framework for Restaurant Survival Prediction and Explanation

Xin Li [1], Xiaojie Zhang [1], Jiahao Peng [1], Rui Mao [1], Mingyang Zhou [1], Xing Xie [2], Hao Liao [1*]
Shenzhen University, China[1]
Microsoft Research Asia[2]
{1910273046, 1800271040, 2070276145}@email.szu.edu.cn
{mao, zmy, haoliao}@szu.edu.cn
xing.xie@microsoft.com

Abstract

The bloom of the Internet and the recent breakthroughs in deep learning techniques open a new door to AI for E-commence, with a trend evolved from using a few financial factors such as liquidity and profitability to using more advanced AI techniques to process complex and multi-modal data. In this paper, we tackle the practical problem of restaurant survival prediction. We argue that traditional methods ignore two essential aspects, which are very helpful for the task: 1) modeling customer reviews and 2) jointly considering status prediction and result explanation. Thus, we propose a novel joint learning framework for explainable restaurant survival prediction based on the multi-modal data of user-restaurant interactions and users' textual reviews. Moreover, we design a graph neural network to capture the high-order interactions and design a co-attention mechanism to capture the most informative and meaningful signal from noisy textual reviews. Our results on two datasets show a significant and consistent improvement over the SOTA techniques (average 6.8% improvement in prediction and 45.3% improvement in explanation).

1 Introduction

Business survival prediction is a hot topic in management and finance literature. Traditional methods rely heavily on financial factors to research(e.g., liquidity, solvency, and profitability)(Ziman, 1991; Lussier, 1996; Pereira et al., 2020). However, there are two significant drawbacks: 1) the financial factors of a shop/company are hard to obtain due to privacy issues; 2) meanwhile, financial factors are macro indicators that reveal the status of a business only on a coarse level. With the development of information techniques, much restaurant-related data can be collected online. For example, people can post check-ins after consuming in a restaurant, and they can share reviews to show how/why they like the restaurant via an online review platform, such as Yelp.com. Moreover, reviews contain informative users' feedback on a fine-grain level. More importantly, the feedback which deeply reflects the restaurant's operating status, can in turn help to generate explainable prediction reasons. Some recent research works also verify this, and the use of online reviews to understand business performance is an emerging trend(Babić Rosario et al., 2016)(Kong et al., 2017).

Recent advances in deep learning have various models that research reviews and interactions for different kinds of tasks, such as recommendation (Wang et al., 2019), fake news detection (Potthast et al., 2018; Wang, 2017), rating prediction (Tay et al., 2018), but little attention has been paid to the application of restaurant survival analysis. In this paper, we propose a novel joint learning framework to tackle the challenging task of explainable restaurant survival prediction. Our model consists of two compulsory modules: the co-attention network for selecting valuable review texts and the graph neural network for learning high-order interactions on the user-restaurant graph. Specifically, the co-attention mechanism is used to select meaningful review text, which is a feature selection and learning process. The graph of user-item interactions could reveal the preference similarity between users (or items). Therefore, the construction of graph neural networks, on which we encode high-order relationships, can en- hance the representation of reputed users and high-quality restaurants by modeling the high-order interaction between user and restaurant, which is the key to our modeling exemplification.

Merely predicting the future status of restaurant survival is inadequate. It is also critical for businesses to understand why they will prosper or close in the future. Fortunately, we can leverage NLP models to encode the massive user reviews and output some explanations, just like a document summarization process. To this end, we jointly train the survival prediction and explanation task.

Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 3285 - 3297

The prediction task predicts the future status of the restaurant, and the explanation task generates some explainable texts to provide an informative summarization for the restaurant's management. We named the unified framework Restaurant Survival Prediction and Explanation (RSPE). Through experiments, we find that RSPE significantly improves the performance of both tasks compared with several competitive baselines.

The main contributions of our framework are as follows.

1) We propose a new joint learning framework for predicting the survival of restaurants and generating summarizing texts through reviews and interactions.

2) We design two key components in RSPE, i.e., the co-attention component, which mines high-quality and informative reviews, and the graph representation component to encode high-order interactions on the user-restaurant graph.

3) We conduct extensive experiments on Dianping and Yelp datasets. Our model outperforms all SOTA methods significantly on both prediction and explanation tasks, with an average improvement of up to 6.8% on the prediction task and 45.3% on the explanation task.

2 Related Work

Restaurant Survival Analysis: Store survival analysis is an essential and practical research topic in the financial and marketing field, which offers deep insights into stores' financial affairs, marketing strategies, and management (Parsa et al., 2005; Kim and Gu, 2006; Liang et al., 2016; Du Jardin, 2017). Traditionally, researchers usually leverage restaurant financial factors to build linear forecasting models, which are sensitive and hard to obtain. With the development of online services, researchers find that User Generated Content (UGC), such as textual reviews from Yelp.com or Dianping.com, contains massive information covering diverse aspects of stores (restaurants in this paper). Leveraging the heterogeneous UGC can effectively improve the performance of restaurant survival prediction models (Lian et al., 2017). However, the main weaknesses of this group of methods are tthreefold: 1) They used traditional NLP models such as LDA, bag-of-words, or word2vec; 2) they did not consider the interaction graph between customers and restaurants; 3) they did not explicitly reduce the noisy information from the raw UGC.

Pre-trained Model: The pre-trained model has been widely used in the field of NLP. It is trained on large-scale open-domain datasets with self-supervised learning tasks to encode common language knowledge into the model. The well-trained model can be fine-tuned with a small amount of labeled data to perform well on the given target task. For example, BERT (Devlin et al., 2019) is a multi-layer bidirectional Transformer encoder and uses Masked Language Model (MLM) and Next Sentence Prediction (NSP) to capture word and sentence-level representations. UniLM (Dong et al., 2019) is based on Bert, which achieved great success on NLP tasks such as unidirectional, bidirectional, and sequence-to-sequence prediction. Moreover, some studies (Qiu et al., 2020) have shown that the pre-trained model is capable of capturing hierarchy-sensitive and syntactic dependencies, which is beneficial to downstream NLP tasks.

Graph Representation: Graph Neural Network (GNN) is a key component in our framework. GNNs represent a node by fusing self-information with neighborhood information on the graph in a message-passing manner. For example, LightGCN (He et al., 2020) simplifies the classical GCN (Kipf and Welling, 2017) and NGCF (Wang et al., 2019) by removing the transformation layer and non-linear activation functions, and uses a mean pooling aggregator to fuse the neighborhood information. It handles the homogeneous graph. The Heterogeneous Graph Neural Networks model (HetGNN) (Zhang et al., 2019) considers heterogeneous structural (graph) information as well as heterogeneous contents information of each node. Several investigations (Battiston et al., 2021) have already shown that the presence of higher-order interactions may substantially impact the dynamics of networked systems. Thus, we argue that it is necessary to encode high-order interactions from the user-restaurant graph to better model user preference and restaurant status, which existing literature ignores.

3 Problem Statement

We first introduce some definitions and notations, then introduce the problem formulation.

User-Restaurant Interaction Graph: let $G = (U, V, E)$ represent the user-restaurant interaction graph, where $U = \{u_1, u_2..., u_n\}$ denotes the set of users, and $V = \{v_1, v_2..., v_m\}$ is a set of restaurants. $E = \{(u, v)|u \in U, v \in V\}$ denotes the

set of edges, where an edge $(u, v) \in E$ means that user u has reviewed restaurant v.

Reviews: the reviews of a restaurant are defined as $(R^{(V)}_{v,l_1}, ..., R^{(V)}_{v,l_v})$, where $R^{(V)}_{v,i}$ represents the i-th review of restaurant v and l_v is the number of reviews of restaurant v. Similarly, the reviews of the user u are defined as $(R^{(U)}_{u,l_1}, ..., R^{(U)}_{u,l_u})$. We further use $U_v = (u_1, u_2...)$ to denote the list of users who have reviewed restaurant v. The reviews of users related to restaurant v can be defined as $(R^{(U)}_{u_1,1}, ...R^{(U)}_{u_n,l_v}), u_i \in U_v$, where $R^{(V)}_{u_i,j}$ representing the j-th review comes from the reviews of user u_i.

Prediction Task: to predict the future status of the restaurant. This is a binary classification task, and 0 means the restaurant will be shut down and 1 means normal operation.

Explanation Task: besides the binary prediction task, the model also contains an explanation task, in which a sentence of summarization text $\boldsymbol{Y} = (w_1, \ldots, w_T)$ will be generated. **We invited 30 evaluators who are split into two groups to manually select a few sentences (about 30 words) from all the restaurant reviews to represent the key reasons for each restaurant's business prosperity, which will be used as ground-truth for training and evaluation**.

Problem Formulation: with the interaction graph G and review collections $R^{(U)}$ and $R^{(V)}$ as input data, we want to make predictions for a given restaurant regarding its future status and meanwhile generate an explaining text.

4 The Proposed Model

In this section, we introduce our model, a joint learning framework for restaurant survival prediction and explanation, which is illustrated in Figure1. There are four components in RSPE, including an input module, a co-attention module, a graph representation module and a joint learning module. We will introduce the details in the following sections.

4.1 Input Module

The function of the input module is to encode the input feature, and the input includes two types of sequences: the reviews of restaurants $(R^{(V)}_{v,1}, ..., R^{(V)}_{v,l})$, and related users' reviews $(R^{(U)}_{u_1,1}, ...R^{(U)}_{u_n,l}), u_i \in U_v$. Each sequence includes a list of reviews. This module encodes reviews to embedding representations. Each review is composed of a sequence of sentences. We use UniLM (Dong et al., 2019) to transform each sentence into a d-dimensional embedding representation $z \in \mathbb{R}^d$, because UniLM is pre-trained on a large-scale unsupervised dataset and through our experiments, we find that it is better than BERT. Given a review $R^{(V)}_{v,i}$, its embedding vector $\mathbf{r}_{u,i} = \sum_{\boldsymbol{z} \in \mathbb{R}_{u,i}} \boldsymbol{z}$ is represented by the average of sentence embeddings in the review. In addition, we use an embedding-lookup operation to get a trainable embedding vector representation for each user and restaurant from her/its ID, which will be used as the input for the graph representation module (will be introduced in Section 4.3).

4.2 Co-attention Module

The intuition is simple but powerful. Each user is represented by all reviews that he/she wrote, and the restaurant is represented by all reviews belonging to it. The goal of the co-attention module is to select high-quality reviews from the user/restaurant's review collection and finally merge reviews' embedding into one user/restaurant embedding.

Affinity Matrix: given user review embedding $\boldsymbol{a}_i (a_i \in \mathbb{R}^{l \times d})$ and restaurant review embedding $\boldsymbol{b}_j (b_j \in \mathbb{R}^{l \times d})$, the affinity matrix is calculated by :

$$\boldsymbol{M}_{i,j} = f\left(\boldsymbol{a}_i\right)^{\mathrm{T}} \mathbf{A} f\left(\boldsymbol{b}_j\right), \tag{1}$$

where $\mathbf{A} \in \mathbb{R}^{d \times d}$ is the weight matrix, and $f(.)$ is a feed-forward neural network.

Max Pooling Function: we use arg max to obtain the maximum value of each row and each column of the matrix, then weigh the review $\boldsymbol{a}_i$ and $\boldsymbol{b}_j$ respectively. The calculation process is as follows:

$$\boldsymbol{\zeta}_i = \left(Gumbel\left(\max_{col}(\boldsymbol{M})\right)\right)^{\top} \boldsymbol{a}_i, \tag{2}$$

$$\boldsymbol{\eta}_j = \left(Gumbel\left(\max_{row}(\boldsymbol{M})\right)\right)^{\top} \boldsymbol{b}_j, \tag{3}$$

where $\boldsymbol{\zeta}_i$ and $\boldsymbol{\eta}_j$ represent the co-attention embeddings of the user and the restaurant. $Gumbel()$ is Straight-Through Gumbel softmax (Jang et al., 2017), due to the arg max function is not differentiable, we use $Gumbel()$ to return a discrete vector and turn the unnormalized vectors $\boldsymbol{e} = (e_1, e_2, ..., e_d)$ into a probability distribution:

$$\boldsymbol{s_i} = \frac{\exp\left(\frac{\boldsymbol{e_i} + g_i}{\tau}\right)}{\sum_{j=1}^{d} \exp\left(\frac{\boldsymbol{e_j} + g_i}{\tau}\right)}, \tag{4}$$

where τ is a temperature parameter, and g_i is a Gumbel noise. In the feedforward process, $\boldsymbol{s_j}$ will

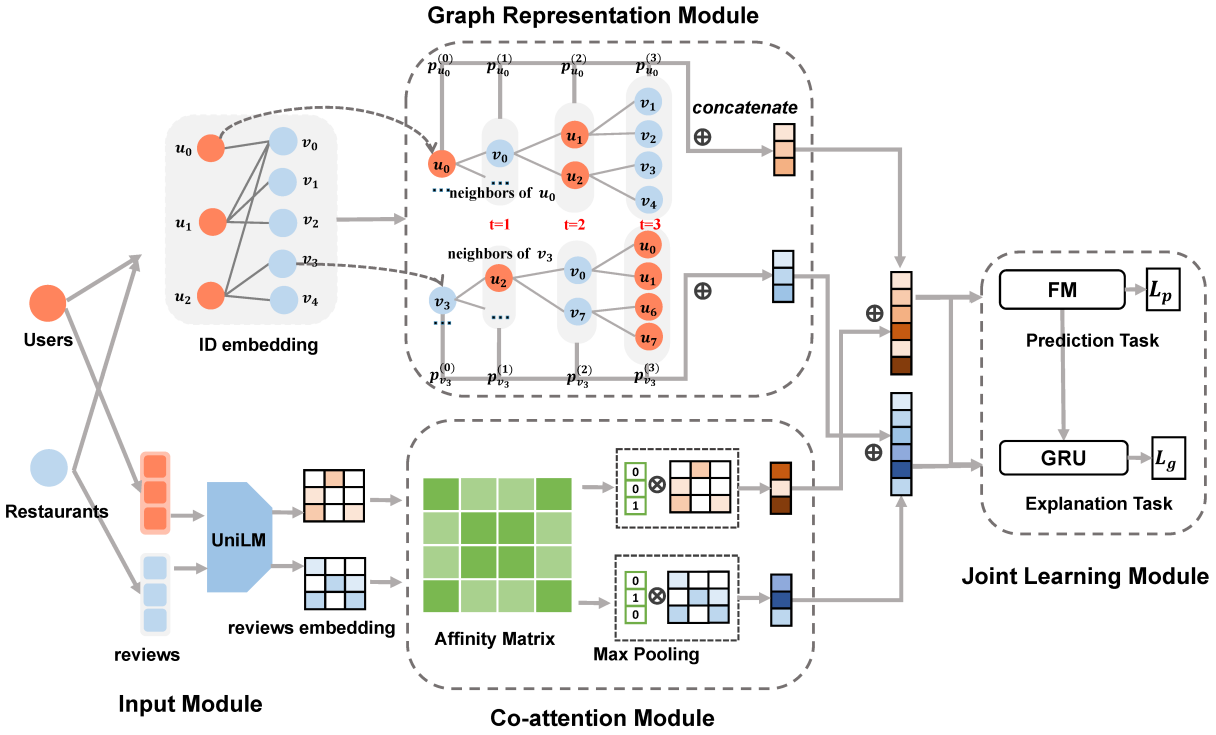

Figure 1: An overview of the RSPE framework.

be transformed into a one-hot vector $\boldsymbol{k_i}$, we denote this function as $Gumbel(\boldsymbol{s}) = \boldsymbol{k}$:

$$\boldsymbol{k_i} = \begin{cases} 1, & i = \arg\max_j (\boldsymbol{s_j}) \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

4.3 Graph Representation Module

Restaurants are not isolated. Sometimes we cannot understand why a restaurant becomes so popular if we only consider its review content. Many factors influence the business status of a restaurant, such as nearby competitors and the general social trend. In order to model the global context of restaurants, we construct a bipartite graph, on which the nodes are restaurants and users, and the edges are user-restaurant interactions. Since GNNs have demonstrated great superiority in learning useful information from graph-structure data (Veličković et al., 2018; Hamilton, 2020), in this section, we introduce a graph representation module to learn meaningful patterns on the user-restaurant interaction graph, so that restaurants' information are enhanced by their neighborhood.

The interaction graph G is illustrated in Figure 1. It stems from the idea that a specific interaction between the user and the restaurant can reveal the restaurant's survival.

Node Embedding: we obtain the trainable embedding vector by the user ID and the restaurant ID, denoted by $\boldsymbol{p}_u^{(0)}$ and $\boldsymbol{q}_v^{(0)}$ respectively.

High-order Neighbor Aggregation: the neighbour nodes embedded in the propagation layer of different orders have different effects on the target node. By stacking multiple propagation layers, we can explore high-level connectivity information and enhance the representation. According to the propagation rules, we obtain the neighbour nodes of the first-order, second-order, and third-order propagation layers adjacent to the target node, and the propagation layer embedding is calculated as follows:

$$\boldsymbol{p}_u^{(t+1)} = \sum_{v \in S_u} \frac{1}{\sqrt{|S_u|}\sqrt{|S_v|}} \boldsymbol{p}_v^{(t)}, \tag{6}$$

$$\boldsymbol{p}_v^{(t+1)} = \sum_{u \in S_v} \frac{1}{\sqrt{|S_v|}\sqrt{|S_u|}} \boldsymbol{p}_u^{(t)}, \tag{7}$$

where $\boldsymbol{p}_u^{(t)}$ and $\boldsymbol{p}_v^{(t)}$ represent the embeddings of user u and restaurant v after t^{th} layer propagation respectively, S_u and S_v represent the first-hop neighbors of user u and restaurant v.

To avoid the large embedding scale, each layer of convolution nodes needs to be regularized. Then, the obtained propagation embedding layer is aggregated to obtain the final target node embedding. The calculation process is as follows:

$$\boldsymbol{p}_u = \sum_{t=0}^{T} \boldsymbol{\alpha}_t \boldsymbol{p}_u^{(t)}, \boldsymbol{p}_v = \sum_{k=0}^{T} \boldsymbol{\alpha}_t \boldsymbol{p}_v^{(t)}, \tag{8}$$

where $\boldsymbol{\alpha_t}$ represents the weight of the T^{th} $(T = 0, 1, 2, 3)$ layer embedding.

For each restaurant v, there will be many user reviews. Therefore, we use a mean pooling to aggregate the vector representations of $u \in S_v$ of all users who have reviewed restaurant v, which is expressed as follows:

$$\boldsymbol{p}_{S_v} = \sum_{u \in S_v} \boldsymbol{p}_u \tag{9}$$

4.4 Joint Learning Module

Joint learning is an inductive transfer method to improve generalization by using the domain information in the training signals of related tasks as an inductive bias. Since the prediction and explanation tasks are associated, we jointly train them in a unified framework to make a better-generalized performance.

We aggregate the embeddings of users and restaurants in the co-attention module and graph representation module. The formula is as follows:

$$\boldsymbol{q}_u = \boldsymbol{\zeta}_i + \boldsymbol{p}_{S_v}, \boldsymbol{q}_v = \boldsymbol{\eta}_j + \boldsymbol{p}_v. \tag{10}$$

Prediction Task: the factorization machine (Rendle, 2010) helps extract the most essential latent or hidden features, which can solve the classification problem. The formula is as follows:

$$f(\boldsymbol{q}) = \boldsymbol{b} + \sum_{i=1}^{n} \boldsymbol{w}_i \boldsymbol{q}_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} \langle \boldsymbol{h}_i, \boldsymbol{h}_j \rangle \boldsymbol{q}_i \boldsymbol{q}_j, \tag{11}$$

where $\boldsymbol{q}_i \in \mathbb{R}^d$ is the i^{th} entry of $\boldsymbol{q} = [\boldsymbol{q}_u, \boldsymbol{q}_v]$, $\boldsymbol{b} \in \mathbb{R}^d$ is the bias, $\boldsymbol{w}_i \in \mathbb{R}^d$ and $\boldsymbol{h}_i \in \mathbb{R}^k$ are parameters to be learned. The loss function uses sigmod cross entropy:

$$L_p = \frac{1}{2|\boldsymbol{\Theta}|} \sum_{(u,v) \in \boldsymbol{\Theta}} (-[\boldsymbol{y} \log \hat{\boldsymbol{y}} + (1-\boldsymbol{y}) \log(1-\hat{\boldsymbol{y}})]), \tag{12}$$

where $\boldsymbol{y}$ is truth label and $\boldsymbol{\Theta}$ represents the training set.

Explanation Task: since the Gated Recurrent Unit(GRU) (Cho et al., 2014) performs well in the generation, we choose it for the explanation task. The details of GRU are as follows. First, calculate the initial hidden state $\boldsymbol{h}_0$:

$$\boldsymbol{h_0} = \tanh\left(\boldsymbol{w}^1 \boldsymbol{q}_u + \boldsymbol{w}^2 \boldsymbol{q}_v + \boldsymbol{w}^3 \hat{\boldsymbol{y}} + \boldsymbol{b}_e\right), \tag{13}$$

where $\boldsymbol{w}^1$, $\boldsymbol{w}^2$ and $\boldsymbol{w}^3$ are parameters to learned. $\boldsymbol{b}_e$ is the bias.

The current t state is related to the last $t-1$ state:

$$\boldsymbol{h}_t = GRU\left(\boldsymbol{h}_{t-1}, \boldsymbol{w}_t\right), \tag{14}$$

where $\boldsymbol{w}_t$ is the word generated at time t.

The final output layer generates the distribution $\mathbf{d}_t$ of words from the hidden state at time t:

$$\boldsymbol{\eta}_t = O\left(\boldsymbol{w}^4 \boldsymbol{h}_{t-1} + \boldsymbol{b}_r\right), \tag{15}$$

where $\boldsymbol{w}^4$ are parameters to learned and $\boldsymbol{b}_r \in \mathbb{R}^{|\mathcal{V}| \times l_d}$ is the bias. $|\mathcal{V}|$ is the vocabulary size and $O()$ is the softmax function. Then, we use beam search to select the best text generated $\boldsymbol{Y}$.

We expect to maximize the probability of the ground-truth text. Thus, the loss function for the explanation task is:

$$L_g = \frac{1}{|\boldsymbol{\Theta}|} \sum_{(u,v) \in \boldsymbol{\Theta}} \sum_{t=1}^{T} \left(-\log \boldsymbol{\eta}_{t,\hat{l_t}}\right), \tag{16}$$

where $\hat{l}_t$ is the word of ground-truth text at time t.

Multi-task Loss: by sharing the representations between related tasks, we aggregate the three loss functions of the two tasks for optimization:

$$\mathcal{L} = \lambda_1 L_p + \lambda_2 L_g + \lambda_3 \|\boldsymbol{\Psi}\|_2^2, \tag{17}$$

where $\lambda_\xi (\xi = 1, 2, 3)$ are hyper-parameters that control the weight of different loss functions. $\boldsymbol{\Psi}$ denotes the set of trainable parameters. For more details on the setting of hyper-parameters, please refer to the appendix.

5 Experiments

5.1 Datasets

We experiment with two public datasets, the basic statistics are listed in Table 1:

Dianping[1]**:** it is the largest consumer review site in China. This dataset records reviews from Jan.2011 to Dec.2011 and restaurants' status in Dec.2011 as the binary label. In the Dianping dataset, the top 3 popular cities are used in the experiments: Shanghai (**SH**), Beijing (**BJ**) and Guangzhou (**GZ**).

Yelp[2]**:** it is the largest review site for business. We use the latest restaurant records reviews from Jan.2019 to Dec.2019 and restaurants' status in Dec.2019 as the binary label. In the Yelp dataset, the top 3 popular states are Nevada (**NV**), Arizona (**AZ**) in the United States, and Ontario (**ON**) in Canada.

Due to space limitations, for more details on data processing, please refer to the appendix.

[1] http://yongfeng.me/dataset/

[2] https://www.kaggle.com/yelp-dataset/yelp-dataset

Table 1: Statistics of clean datasets in the experiments from Dianping and Yelp

dataset		#restaurant	#closure restaurant	#closure ratio
Dian Ping	SH	10251	3312	32.31%
	BJ	5067	1308	25.81%
	GZ	1932	509	26.34%
Yelp	NV	4764	223	4.68%
	AZ	6623	258	3.90%
	ON	5688	209	3.67%

5.2 Metrics

In our experiments, we use **AUC** (Hanley and McNeil, 1982) to evaluate the prediction task. **BLEU** (Papineni et al., 2002) and **ROUGE** (Lin, 2004) are used to evaluate the explanation task. ROUGE's evaluation is based on the co-occurrence information of n-grams in the text. ROUGE-N (N=1,2) mainly counts on the N-grams. ROUGE-L is calculated by matching the longest common subsequence. ROUGE-SU4 is calculated by the skip-gram strategy, when generating explanation text and ground-truth text for matching, which does not require that the words must be continuous, and several words could be "skipped". The larger value of BLEU and ROUGE indicates better explainability.

5.3 Performance Evaluation

To evaluate the prediction task, we compare the RSPE with two groups of baselines which perform binary classification tasks as our prediction module:

Traditional Machine Learning: we take the heterogeneous information obtained by encoding reviews through Word2Vec (Church, 2017) and Bag of Word as input features for traditional machine learning methods, including: **1) LR** (Cortes and Vapnik, 1995). **2) SVM** (Cortes and Vapnik, 1995). **3) GBDT** (Friedman, 2001).

Deep Learning: we also compare with several competitive deep learning based methods, including: **1) text-CNN** (Kim, 2014): a modified convolutional neural networks model. **2) text-RNN** (Lai et al., 2015): a modified long short-term memory model. **3) MPCN** (Tay et al., 2018): a review-based attention network model that combines multi-pointer for recommendations. **4) HetGNN** (Zhang et al., 2019): a heterogeneous graph neural network for various graph mining tasks by aggregating different types of nodes. **5) DCA** (Liao et al., 2020): a review based attention neural model for data augmentation by selecting concepts. **6) HGAT** (Li et al., 2020): a hierarchical graph attention network to accomplish the semi-supervised node classification tasks.

Table 2: Results of the prediction task

Method	Dianping			Yelp		
	SH	BJ	GZ	NV	AZ	ON
LR	0.7203	0.7081	0.7103	0.5812	0.6747	0.6111
SVM	0.7097	0.7049	0.6518	0.5391	0.6092	0.561
GBDT	0.573	0.6003	0.5932	0.645	0.7135	0.653
text-CNN	0.5647	0.5537	0.5604	0.5896	0.5558	0.5414
text-RNN	0.5683	0.5656	0.5675	0.5535	0.5317	0.5456
MPCN	0.5573	0.6783	0.6782	0.6972	0.7672	0.7563
HetGNN	0.5107	0.5165	0.499	0.6234	0.6711	0.6122
HGAT	0.7753	0.7463	0.7598	0.7718	0.7582	0.7825
DCA	0.8412	0.8612	0.8379	0.9014	0.8752	0.8856
RSPE	**0.8994**	**0.9073**	**0.9096**	**0.9521**	**0.9171**	**0.9379**
Improvement	6.91%	5.35%	8.56%	5.63%	4.80%	5.91%

To evaluate the explanation task, we compare RSPE with two groups of baselines that both perform well on text generation.

Generative-based Methods: NRT (Li et al., 2017) is a framework based on user review information, which generates abstractive text with good linguistic quality for prediction explanation. **DCA** (Liao et al., 2020) is a framework based on attention neural, which generates diverse texts through a large amount of text learning. **PETER** (Li et al., 2021): a personalized Transformer that shows good performance in text generation tasks.

Retrieval-based Method: the retrieval method selects the most important text from reviews as explanation sentence. **Lexrank** (Erkan and Radev, 2004) is an unsupervised text summarization method based on graph-based lexical centrality, which generates summary text by reviews.

5.4 Implementation Details

In our experiments, we randomize the datasets into a training set (70%), validation set (15%), and test set (15%). We follow the corresponding papers to adjust the baselines to ensure the best results. The hyperparameter settings and implementation details are listed in the appendix.

5.5 Results on the Prediction Task

The overall prediction results are shown in Table 2. Our model's improvement over the best baseline is quite significant. For example, a performance gain up to 6.9%/5.4%/8.6% on the Dianping dataset of city SH/BJ/GZ, and 4.8%/9.5%/5.9% on the Yelp dataset on state AZ/NV/ON, which demonstrates the effectiveness of our model.

In addition, we have the following 4 observations about the results. First, the MPCN, DCA, and HGAT are generally better than the traditional methods. Those methods use an attention mechanism to build their model. HGAT also considers heterogeneous graph convolution, demonstrating

Table 3: Results of the explanation task

Method	BLEU	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-SU4
Dianping-SH					
NRT	1.34	2.7	0.78	2.67	0.94
LexRank	1.49	4.85	1.04	4.83	1.14
DCA	1.50	6.07	1.65	5.17	1.76
PETER	0.97	5.72	1.09	4.89	1.45
RSPE	**2.19**	**7.15**	**1.79**	**7.12**	**1.98**
Improvement	46.0%	17.8%	8.5%	37.7%	12.5%
Dianping-BJ					
NRT	1.19	2.39	0.62	2.393	0.127
LexRank	1.65	4.39	0.98	4.39	0.98
DCA	1.77	4.98	1.20	4.93	1.28
PETER	1.04	4.53	1.22	4.58	1.35
RSPE	**2.90**	**6.87**	**1.94**	**6.85**	**2.02**
Improvement	63.2%	38.0%	59.0%	38.9%	57.7%
Dianping-GZ					
NRT	1.31	5.46	0.57	0.546	0.78
LexRank	1.74	5.28	0.90	5.27	0.86
DCA	1.79	7.68	2.55	7.42	2.66
PETER	1.03	12.00	0.59	12.43	1.25
RSPE	**3.28**	**13.23**	**3.74**	**13.07**	**3.85**
Improvement	83.2%	10.3%	46.7%	5.1%	43.8%
Yelp-NV					
NRT	1.31	21.08	8.86	16.45	10.94
LexRank	1.33	15.95	3.45	12.3	5.21
DCA	2.09	28.51	12.08	23.21	13.53
PETER	1.06	17.69	2.72	9.07	7.39
RSPE	**2.66**	**30.15**	**13.48**	**24.20**	**14.69**
Improvement	27.4%	5.7%	11.5%	4.2%	8.6%
Yelp-AZ					
NRT	1.64	21.67	8.60	16.57	10.90
LexRank	1.69	19.35	5.31	15.08	7.01
DCA	2.54	30.04	13.45	24.12	15.02
PETER	1.36	14.02	0.19	9.33	10.4
RSPE	**3.15**	**32.50**	**15.58**	**26.66**	**17.25**
Improvement	24.2%	8.2%	15.8%	10.5%	14.8%
Yelp-ON					
NRT	1.29	24.5	12.42	19.63	14.53
LexRank	1.1	16.17	4.21	12.74	5.63
DCA	1.67	27.36	10.20	20.31	12.31
PETER	0.94	23.19	2.12	13.1	9.87
RSPE	**2.13**	**31.01**	**14.74**	**24.83**	**16.43**
Improvement	27.7%	13.3%	44.5%	22.2%	33.4%

that the information of the heterogeneous graph and attention mechanism may contribute to the model performance. Second, a simple graph structure cannot perform well in the prediction task, such as HetGNN. Third, our model performs well on Dianping and Yelp datasets, demonstrating that our model is robust across different datasets. Fourth, our model achieves better performance. Our model can not only automatically dig important information in massive reviews through the co-attention module but also combine the interaction information between users and restaurants to capture the most informative and meaningful signal from noisy textual reviews.

5.6 Results on the Explanation Task

The detailed results are shown in Table 3. First, the performance of our model on the explanation task is significantly better than the SOTA methods. Take the BLEU metric as an example, RSPE achieves an improvement of 46.0%/63.2%/83.2%/27.4%/24.2%/27.7% in SH/BJ/GZ/NV/AZ/ON, and the average improvement of 45.3%. The ROUGE (ROUGE-1/2/L/SU4) indicator mainly considers overall accuracy, RSPE achieves an improvement of 49.1% in BJ city, and the average improvement in all datasets is as high as 23.8%. Second, in 6 cities/states, NRT's explanation performance is not good because it is based on historical records to learn the latent factors and can only output some general-purpose expressions. Third, the retrieval method Lexrank does not perform well because it focuses on similarity matching while lacking personalized expression. Because the framework of DCA is too complex, its feature selection ability is insufficient, so the overall performance is lower than our model. Although PETER proposes a new Transformer structure to generate text, the results show that its performance improvement is modest. At last, our RSPE performs significantly better in the text of both Chinese and English datasets because we leverage a graph convolutional neural network to enhance hidden collaborative signals modeling from the user-restaurant interaction, which enables the model to learn the reputed reviews to improve the quality of explanation text. This observation is in line with the results mentioned above. It further verifies that by including graph structure in the modeling process, our model can learn the interaction information between user and restaurant and thus generate informative textual expressions for the restaurant survival.

Table 4: Explanations generated by RSPE and Baseline.

Case 1	**Delicious**! The customer **service** is pretty **good** and the open all the way to 3 am In the morning. The prime burgers are **excellent**!
Lexrank	The customer **service** is pretty **good**!
NRT	Best!
DCA	The customer **service** is pretty **good**!
RSPE	**Delicious**! The customer **service** is pretty **good**! The only issue was the front of the **best** ! It 's a lot of what is some of the **best**.
Case 2	The **environment** is not good, the **service** is not good, and the main **dishes** are **terrible**. After several times of food, the boss has always been very **disdainful**. Noodles with soybean paste is much more expensive than before, it is far from before. Anyway, I won't go again...
Lexrank	The **taste** is **OK**, the **environment** is just so-so, noodles with soybean paste is much more expensive than before.
NRT	The **taste** is **good**, the **environment** is **bad**, and the **service** is not **good**.
DCA	The **taste** is **good**, the **environment** is **bad**, need to line up and wait every time, the **price** is much higher than before.
RSPE	The **taste** is **good**, the **environment** is not **good**, the **service** is not **good**, and the **dishes** are **poor**. The **price** is much higher than before, in short, it is not **recommended**.
Case 3	The **ostentation** is **huge**, and the dining **environment** is also **good**. Unfortunately, the most important food was **terrible**. The ingredients were not **fresh**, and the taste was not **good** enough. I would not care about it any more.
Lexrank	The **environment** is **good**, the **service** is **good**.
NRT	The **environment** is **good**, the **dishes** are not **good**.
DCA	he **environment** is **good**, the **service** is **good**, but the food is too **bad**. The ingredients were not **fresh**, that's too **bed**.
RSPE	The **ostentation** and **environment** are very **good**, and the **service** is also very **good**, but the food is too **bad**. The ingredients are not **fresh**, so I won't go there any more.

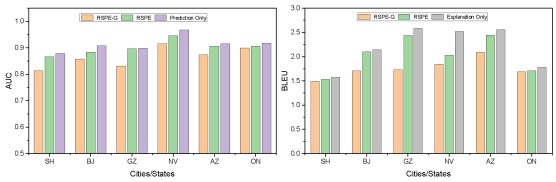

Figure 2: RSPE ablation analysis in AUC and BLEU

Table 5: Results on the fluency evaluation.

Measures	NRT	LexRank	DCA	Our
Fluency	2.98	3.24	3.46	**3.75**
(Kappa)	(0.76)	(0.73)	(0.74)	(0.8)

5.7 Ablation Analysis

In order to study the effectiveness of joint learning in the model, we performed an independent task experiment for prediction and explanation respectively, denoted as "Prediction Only" and "Explanation Only". Additionally, we remove the graph representation module from the model, denoted as "RSPE-G". The results are shown in Figure 2. It has been proved that independent tasks can achieve better results, but RSPE could achieve a balance between prediction accuracy and interpretation ability through the joint learning framework. In addition, it is clear that the graph representation module indeed plays a significant contribution. This proves again that the graph of high-order interaction enhances the power to capture the most informative and meaningful signal from noisy textual reviews, thus more accurate prediction and more reasonable explanations.

5.8 Case Analysis

We take three cases generated from LexRank, NRT, DCA, and RSPE as examples, which are shown in Table 4. We bold the frequent adjective and nouns in the reviews as keywords, and the cases of Dianping are transformed from Chinese to English. This table shows that: 1) The explanation words generated by RSPE are more comprehensive and cover many important factors such as environment, service, taste, and price. Meanwhile, the generated content is highly consistent with the ground-truth text. 2) RSPE has a powerful generalizing ability to summarize relevant sentences, such as *The ostentation and environment are very good* in Case 3. 3) RSPE can generate personalized language expressions, such as *The only issue was the front of the best* in Case 1 and *in short, it is not recommended* in Case 2.

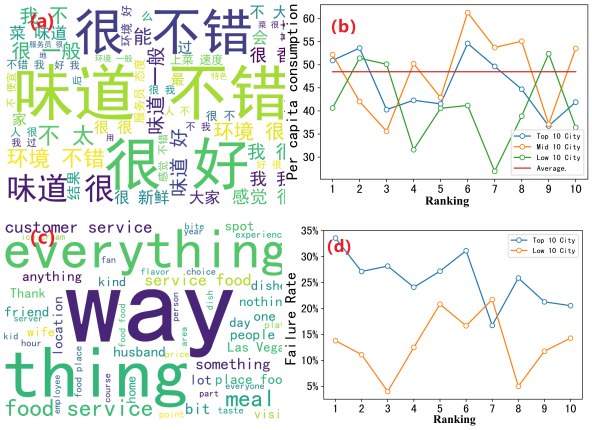

Figure 3: Review word cloud and failure rate statistics.

5.9 Fluency Evaluation

Next, we evaluate the model's usefulness in improving the fluency of the generated sentences. The fluency evaluation experiment is done by human judgment. We randomly selected 100 samples and invited 5 annotators to assign scores. Five points mean very satisfied, and 1 point means very bad. The human evaluation results are reported in Table 5. Results demonstrate that our model outperforms the other three methods on Fluency and Kappa (Li et al., 2019) metrics.

5.10 Survival Discussions

A restaurant's survival is not only related to user reviews but also affected by many off-site factors, such as financial breakdown and competitive pressure. Therefore, we hope to mine some instructive explanations for the sustainable development of the restaurant industry through some data analysis.

As shown in Figure 3 (a) and (c), users of Dianping pay more attention to taste (taste, good, fresh). In contrast, users of Yelp are more concerned about the environment and service (service, place, way, location). As shown in Figure 3 (c) and (d), the per capita consumption of medium cities is generally higher than that of big and small cities, and the failure rate of restaurants in small cities is much lower than that in big cities. We found that we could explore the restaurant's survival from a more fine-grained perspective, which to mine the rules, and helped adjust their strategies to promote business.

6 Conclusion

In this paper, we tackle the problem of restaurant survival, which is an essential task for social good. Unlike traditional methods, which highly rely on sensitive financial indicators, we use deep learning techniques to mine useful signals from massive

UGC. We are the first to conduct both future status prediction and explanation simultaneously as a joint framework. Our model has two key components, i.e., the graph representation module and the co-attention module. We conduct extensive experiments on two datasets. Results demonstrate that our proposed model achieves the SOTA performance on both prediction and explanation tasks.

7 Limitations

Current limitations of this paper are threefold. First, a limited set of features are used in this paper. Whether a restaurant can survive is influenced by many factors, such as finances, social circumstances (such as Covid-19), and other issues that can exacerbate a restaurant's survival. In this paper, we can't fully explain those additional factors out of the review text. We just took a new perspective on the restaurant survival prediction task from NLP. Second, the model structure is not lightweight enough, and there is still room for model simplification, such as the combination of attention mechanisms and graph neural networks. Third, the data application scope of the model is not large enough. Currently, only two datasets have been tested in 6 cities/states. We do not test the model on data samples on more different online service platforms.

8 Acknowledgments

Hao Liao is the corresponding author. Thanks a lot for Dr. Jianxun Lian and Dr. Xiting Wang's valuable suggestions and help. This work was supported by the Natural Science Foundation of China under Grant no. 62276171 and 62072311, the Natural Science Foundation of Guangdong Province of China under Grant Nos. 2019A1515011173 and 2019A1515011064, the Shenzhen Fundamental Research-General Project under Grant No. JCYJ20190808162601658, CCF-Baidu Open Fund , NSF-SZU and Tencent-SZU fund.

References

Ana Babić Rosario, Francesca Sotgiu, Kristine De Valck, and Tammo HA Bijmolt. 2016. The effect of electronic word of mouth on sales: A meta-analytic review of platform, product, and metric factors. *Journal of Marketing Research*, 53(3):297–318.

Federico Battiston, Enrico Amico, Alain Barrat, Ginestra Bianconi, Guilherme Ferraz de Arruda, Benedetta Franceschiello, Iacopo Iacopini, Sonia Kéfi, Vito Latora, Yamir Moreno, et al. 2021. The physics of higher-order interactions in complex systems. *Nature Physics*, 17(10):1093–1098.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1724–1734.

Kenneth Ward Church. 2017. Word2vec. *Natural Language Engineering*, 23(1):155–162.

Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine learning*, 20(3):273–297.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, pages 13063–13075.

Philippe Du Jardin. 2017. Dynamics of firm financial evolution and bankruptcy prediction. *Expert Systems with Applications*, 75:25–43.

Günes Erkan and Dragomir R Radev. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of artificial intelligence research*, 22:457–479.

Jerome H Friedman. 2001. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232.

William L Hamilton. 2020. Graph representation learning. *Synthesis Lectures on Artifical Intelligence and Machine Learning*, 14(3):1–159.

James A Hanley and Barbara J McNeil. 1982. The meaning and use of the area under a receiver operating characteristic (roc) curve. *Radiology*, 143(1):29–36.

Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. 2020. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 639–648.

Eric Jang, Shixiang Gu, and Ben Poole. 2017. Categorical reparameterization with gumbel-softmax. In *International Conference on Learning Representations(Poster)*, page 4.

Hyunjoon Kim and Zheng Gu. 2006. A logistic regression analysis for predicting bankruptcy in the hospitality industry. *The Journal of Hospitality Financial Management*, 14(1):17–34.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1746–1751.

Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, pages 1–14.

Grace Kong, Jennifer Unger, Lourdes Baezconde-Garbanati, and Steve Sussman. 2017. The associations between yelp online reviews and vape shops closing or remaining open one year later. *Tobacco prevention & cessation*, 2(Suppl).

Siwei Lai, Liheng Xu, Kang Liu, and Jun Zhao. 2015. Recurrent convolutional neural networks for text classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29, pages 2267–2273.

Junyi Li, Wayne Xin Zhao, Ji-Rong Wen, and Yang Song. 2019. Generating long and informative reviews with aspect-aware coarse-to-fine decoding. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1969–1979.

Kangjie Li, Yixiong Feng, Yicong Gao, and Jian Qiu. 2020. Hierarchical graph attention networks for semi-supervised node classification. *Applied Intelligence*, 50(10):3441–3451.

Lei Li, Yongfeng Zhang, and Li Chen. 2021. Personalized transformer for explainable recommendation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4947–4957.

Piji Li, Zihao Wang, Zhaochun Ren, Lidong Bing, and Wai Lam. 2017. Neural rating regression with abstractive tips generation for recommendation. In *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 345–354.

Jianxun Lian, Fuzheng Zhang, Xing Xie, and Guangzhong Sun. 2017. Restaurant survival analysis with heterogeneous information. In *Proceedings of the 26th International Conference on World Wide Web Companion*, pages 993–1002.

Deron Liang, Chia-Chi Lu, Chih-Fong Tsai, and Guan-An Shih. 2016. Financial ratios and corporate governance indicators in bankruptcy prediction: A comprehensive study. *European Journal of Operational Research*, 252(2):561–572.

Hao Liao, Xiaojie Zhang, Xin Li, Mingyang Zhou, Alexandre Vidmer, and Rui Mao. 2020. A deep concept-aware model for predicting and explaining restaurant future status. In *2020 IEEE International Conference on Web Services*, pages 559–567.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Workshop on Text Summarization Branches Out, Post-Conference Workshop of ACL 2004*, pages 74–81.

Robert N Lussier. 1996. A startup business success versus failure prediction model for the retail industry. *The Mid-Atlantic Journal of Business*, 32(2):79.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

HG Parsa, John T Self, David Njite, and Tiffany King. 2005. Why restaurants fail. *Cornell Hotel and Restaurant Administration Quarterly*, 46(3):304–322.

José Manuel Pereira, Humberto Ribeiro, Amélia Silva, and Sandra Raquel Alves. 2020. *To Fail or Not to Fail: An Algorithm for SME Survival Prediction Using Accounting Data*. Springer International Publishing, Cham.

Martin Potthast, Johannes Kiesel, Kevin Reinartz, Janek Bevendorff, and Benno Stein. 2018. A stylometric inquiry into hyperpartisan and fake news. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 231–240.

Xipeng Qiu, Tianxiang Sun, Yige Xu, Yunfan Shao, Ning Dai, and Xuanjing Huang. 2020. Pre-trained models for natural language processing: A survey. *Science China Technological Sciences*, 63:1872–1897.

Steffen Rendle. 2010. Factorization machines. In *2010 IEEE International Conference on Data Mining*, pages 995–1000.

Yi Tay, Anh Tuan Luu, and Siu Cheung Hui. 2018. Multi-pointer co-attention networks for recommendation. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2309–2318.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. *International Conference on Learning Representations*, pages 1–12.

William Yang Wang. 2017. "liar, liar pants on fire": A new benchmark dataset for fake news detection. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 422–426.

Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural graph collaborative filtering. In *Proceedings of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval*, pages 165–174.

Chuxu Zhang, Dongjin Song, Chao Huang, Ananthram Swami, and Nitesh V Chawla. 2019. Heterogeneous graph neural network. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 793–803.

John M Ziman. 1991. *Reliable knowledge: An exploration of the grounds for belief in science*. Cambridge University Press.

A Appendix On Reproducibility

A.1 Experimental Environment

This experiment runs on GPU V100 and CentOS 7 servers. The code is implemented with Tensorflow.

A.2 Reproducibility

A.2.1 Code Resources

We compared the proposed framework, RSPE, with 11 baseline methods discussed in Section 5.3, the prediction task methods including LR, SVM, GBDT, text-CNN, text-RNN, MPCN, HetGNN, HGAT, DCA and the explanation task methods including Lexrank, NRT and DCA. Our proposed framework, RSPE's code that we have implemented are available through the following link: https://github.com/Complex-data/RSPE. Other codes were obtained as follows:

- **LR, SVM, GBDT:** we used the scikit-learn, which is a publicly machine learning project at: https://scikit-learn.org/stable/index.html
- **text-CNN:** we used the publicly available implementation at: https://github.com/FinIoT/text_cnn
- **text-RNN:** we used the publicly available implementation at: https://github.com/luchi007/RNN_Text_Classify
- **MPCN:** we used the publicly available implementation at: https://github.com/vanzytay/KDD2018_MPCN
- **HetGNN:** we used the publicly available implementation at: https://github.com/ chuxuzhang/ KDD2019_HetGNN
- **HGAT:** we used the publicly available implementation at: https://github.com/BUPT-GAMMA/HGAT
- **DCA:** we used the publicly available implementation at: https://github.com/Complex-data/
- **Lexrank:** we used the publicly available implementation at: https://github.com/ crabcamp/lexrank
- **NRT:** we used the publicly available implementation at: https://github.com/lipiji/NRT-theano

A.2.2 Data Processing

We can download dataset from DianPing[3] and Yelp[4]. Because of the contents of the datasets are different, we conduct data processing for these two datasets respectively.

Dianping: the download content includes two files. One is *checkins.json*, and the other one is *business.json*. *checkins.json* includes all user and shop review records, while *business.json* includes all records about shops. The data processing steps are as follows: 1) Read checkins and business data, and merge these according to restId. 2) Filter non-restaurant data. 3) Filter cities, in our experiment, we used Beijing, Shanghai and Guangzhou data. 4) Filter out 10% of users and restaurants with few reviews. 5) Select the attributes required for the experiment: userid, restId, review, label.

Yelp: the download content include two files. One is *review.json*, and the other one is *business.json*. *review.json* includes all user and shop review records, while *business.json* includes all records about shops. The data processing steps are as follows: 1) Read review and business data, and merge these according to restId. 2) Filter Year, we only use data from 2019. 3) Since Yelp data doesn't have a detailed survival status, we determined restaurants by determining whether *RestaurantsReservations* exist. If this field exists, it means that the store is a restaurant. 4) Filter states, in our experiment, we used Nevada, Arizona and Ontario. 5) Filter out 10% of users and restaurants with few reviews. 6) Select the attributes required for the experiment: userid, restId, review, label

A.2.3 Pre-trained Model

We encode words and sentences by UniLM model. First, we need to download UniLM model from https://github.com/microsoft/ unilm. Then, we use Tensorflow to load the UniLM model, which provides that have been trained. Then, we add our training data to continue training. Finally, we can get a semantic vector representation for each sentence through this pretrained model.

A.2.4 Hyperparameter Setting

For hyperparameter settings for RSPE, we introduce the details of major hyperparameter setting as shown in Table 6. In our experiments, we set λ_1 (pred_lambda=1) by default, and then tune the model by adjusting λ_2 (gen_lambda). The descriptions of the major hyperparameter are as follows:

- gen_lambda: the threshold to control the generating loss weight.
- rnn_type: the threshold to control the compositional model name.

[3] http://yongfeng.me/dataset/

[4] https://www.kaggle.com/yelp-dataset/yelp-dataset

Table 6: The details of the parameters of RSPE

Parameter	Set	Parameter	Set
rnn_type	RSPE	l2_reg	1.00E-06
opt	Adam	len_penalty	2
emb_size	50	implicit	1
rnn_size	30	att_pool	MAX
rnn_dim	400	dmax	50
use_lower	1	beam_size	12
dropout	0.8	init_type	xavier
gen_lambda ($\lambda 2$)	0.01	beam_number	4
rnn_dropout	0.8	emb_dropout	0.8
lr	0.001	epochs	50
att_reuse	0	rnn_layers	1
pred_lambda ($\lambda 1$)	1		

- emb_size: the threshold to control the embeddings dimension.
- rnn_size: the threshold to control the model-specific dimension.
- epoch: the threshold to control the number of epochs.
- lr: the threshold to control the learning rate.
- att_pool: the threshold to control the pooling type for attention.
- dmax: the threshold to control the max number of reviews.
- beam_size: the threshold to control the beam search size.
- pred_lambda: the threshold to control the weight of prediction task

A.2.5 Evaluation

- **Results on the Prediction Task:** we use **AUC** to evaluate the prediction task, and execute test_RSPE.py to get the accuracy in the test set.
- **Results on the Explanation Task: BLEU** and **ROUGE** are used to evaluate the explanation task. For BLEU metrics, we execute evaluate/ compute_bleu.py to get the result score. For ROUGE metrics, we used the publicly available implementation at: https ://github.com/ kavgan/ ROUGE-2.0.