國立臺灣大學管理學院資訊管理學研究所

碩士論文

Department of Information Management
College of Management
National Taiwan University

Master's Thesis

運用深度學習從線上評論萃取顧客痛點

Why are Customers Unsatisfied: A Deep Learning Approach to Extract Customer Pain Points from Online Reviews

莊啟宏

Chi-Hung Chuang

指導教授：魏志平 博士

Advisor: Chih-Ping Wei, Ph.D.

中華民國 113 年 7 月

July 2024

doi:10.6342/NTU202402663

致謝

就讀碩士兩年期間，我能夠將過去大學所學知識理論進行實踐。從一開始大量閱讀他人論文到現在自己撰寫論文，整個過程包含主題發想都需要獨立進行研究，而這些事項都是大學時期的我難以達成的成就。雖說是獨立進行，但一路上也受到了許多幫助，特別是魏老師盡心盡力的支持與指導，讓我能夠在選擇有興趣的主題之下，找到一項有價值的研究。從問題定義、資料搜集、模型設計，到後面的論文內容呈現，老師都給予了極大的幫忙，特別是在最後論文內容的呈現上，有幾個夜晚老師陪著我一頁一頁的修改整個簡報架構，甚至是一字一句的修改用字遣詞。同時也感謝老師協助找了寒舍集團的領域專家幫助我進行論文主題的驗證以及資料的定義，還有即將進入碩士班的芷芊學妹也幫忙了進行許多資料標註。

另外也感謝實驗室的夥伴們，在論文口試前大家一起在實驗室撰寫論文、練習簡報、互相幫助。也非常感謝家人們給予空間與默默支持我就讀兩年的碩士以及論文的撰寫。謝謝實驗室學弟妹在我們忙著趕論文的過程，協助我們許多事項，也幫我們照顧與維修實驗室的機器。謝謝虹鈞在這兩年協助我們許多日常的事項。

經歷了人生中第一次也很可能是我唯一一次的論文訓練，相信未來遇到的任何困難都能迎刃而解。

莊啟宏謹啟

於國立臺灣大學資訊管理所

doi:10.6342/NTU202402663

摘要

在網際網路時代，顧客能夠在社交媒體以及線上評論中分享他們的感受。了解這些顧客需求，尤其是了解顧客不滿意的地方，對於服務改進與創新相當重要，因為這些不滿意的顧客指出了服務無法滿足他們期望的具體之處。顧客不滿意可以分為兩種類型：抱怨和痛點。痛點是帶有具體問題或是不滿意的抱怨，提供可以付諸行為的見解。進行痛點分析有助於公司作出明智的決策。

在過去的研究中，痛點被萃取為關鍵詞或是整個句子，可能導致語意上的模糊或是包含不相關的訊息。此外，只有少數研究包含了痛點分類，提供痛點在類別層面上的評估。

在本研究中，我們提出了一個兩階段的模型來預測顧客評論中的痛點，並將獲得的痛點分類至事先定義的類別之中。我們進一步在不同的領域測試了痛點萃取模型的預測能力。另外，我們採用特殊標記來表示整個評論以進行痛點分類。實驗結果顯示了我們所提出的痛點分析框架的有效性。

關鍵字：深度學習、機器學習、痛點分析、顧客需求、線上評論探勘

doi:10.6342/NTU202402663

Abstract

In the age of the Internet, customers can share their feelings on social media or through online reviews. Understanding these customer needs, especially customer dissatisfaction, is important for service improvement and innovation since unsatisfied customers highlight specific areas where services do not meet their expectations. There are two types of customer dissatisfaction: complaints and pain points. Pain points are complaints with specific problems or dissatisfactions, which provide actionable insights. Conducting pain point analysis assists companies in making informed decisions.

Pain points were extracted as keywords or entire sentences in previous studies, potentially leading to semantic ambiguity or the inclusion of irrelevant information. Additionally, only a few prior studies include pain points categorization, which enables evaluation of pain points at category level.

In this study, we propose a two-phase model to predict the pain point expressions in customer reviews and classify the obtained pain points into predefined categories. We further test the pain point extraction model across different domains. Besides, we adopt special tokens to represent entire reviews for pain point categorization. Experimental results show the effectiveness of our proposed framework for pain point analysis.

doi:10.6342/NTU202402663

Keywords: Deep learning, Machine Learning, Pain point analysis, Customer needs, Online review mining

doi:10.6342/NTU202402663

Table of Contents

doi:10.6342/NTU202402663

doi:10.6342/NTU202402663

List of Figures

doi:10.6342/NTU202402663

List of Tables

doi:10.6342/NTU202402663

Chapter 1 Introduction

1.1 Background

Customer feedback is important because it helps companies improve affective experiences and deliver better service (Khan and Fatma, 2022; Li et al., 2013). When customers experience services, products, dishes, and so on, they may compare them with their expectations. They will be satisfied when the experience meets their expectations and upset otherwise (Parasuraman et al., 1991; Chen and Tabari, 2017). When customers are dissatisfied, it is beneficial for companies to find the reasons behind their dissatisfaction and understand their behaviors (Mutlubaş, 2023). Therefore, it is important to know what customers are thinking.

There are many different methods to understand customer needs. Traditionally, companies have relied on various qualitative research methods to measure customer requirements. Some rely on the intuition and experience of business professionals, while others may try to collect needs directly from consumers through methods such as questionnaires, interviews, focus groups, workshops, consultations with field experts, quality function deployment, house of quality, or a combination of these approaches (Pacheco et al., 2018; Cai et al., 2021). These methods allow companies to directly gather customer needs and develop strategies to meet them. However, they have some

doi:10.6342/NTU202402663

drawbacks. These drawbacks include high costs, information overload, substantial time delays, and difficulties in eliciting responses (Yin et al., 2023). Moreover, the results obtained may only capture a partial perspective of customer needs. With the increase in information transparency nowadays, it is sometimes too late for companies to collect these needs from customers using traditional methods (Salminen et al., 2022).

Because of the rapid advancement of information technology, people are now able to share their thoughts and feelings, which may reflect their satisfaction, on social media or through company reviews (Liu et al., 2017). Before deciding on unfamiliar products or services, reading others' real experiences becomes a valuable reference (Chen and Tabari, 2017). Big data analytics has become a new technique for informed business decision-making. Since people often express their unsatisfied needs online, analyzing online reviews or user-generated content (UGC) offers insights into people's perceptions, needs, unmet expectations, and potential directions for improvement. Compared to traditional methods, exploring online reviews provides a more accurate way for service providers to understand customer satisfaction. To evaluate customer dissatisfaction, UGC serves as a resource for analyzing customer pain points and generating customer insights (Salminen et al., 2022). For example, hotels in many countries have utilized customer online reviews to improve service quality and customer loyalty (Piramanayagam and Kumar, 2020).

doi:10.6342/NTU202402663

Understanding unsatisfied customer needs is important for service improvement and service innovation (Tao et al., 2019; Zhang et al., 2021). These unsatisfied customers highlight specific areas where services do not meet their expectations (Parasuraman et al., 1991; Tao et al., 2019; Salminen et al., 2022; Mutlubaş, 2023). Additionally, it is sometimes more valuable for companies to focus on dissatisfied customers since they provide insights into product defects or service failures (Lee and Hu, 2005). Previous studies on unsatisfied customer needs involve the analysis of customer complaints (Chen and Tabari, 2017) and pain points (Wang et al., 2016; Tao et al., 2019; Salminen et al., 2022; Lee et al., 2023). Complaints refer to negative sentences customers leave, but not all negative sentences contain pain points. Pain points are specific problems or dissatisfactions encountered by customers (Salminen et al., 2022; Lee et al., 2023). For example, the sentence “The service is so bad” is a complaint since we do not know the specific deficiency in the service. In contrast, the sentence “The water in the pool is so cold” refers to a pain point since we know the problem is the temperature of the water. Pain points are more valuable than complaints since they contain actionable insights for potential improvement. These unsatisfied pain points may either represent individual customer needs or genuine product issues, offering companies insights for potential enhancements to current market offerings. Conducting pain point analysis assists companies in making informed decisions. It offers insights into customers’ critical issues,

doi:10.6342/NTU202402663

primary interests, and emerging demands for various offerings to achieve a higher degree of market orientation (Salminen et al., 2022). Pain point marketing (Tao et al., 2019) fills the gap between customer expectations and dissatisfaction, enabling companies to understand customer needs for actionable insights and higher customer satisfaction. There are four different applications of pain point analysis. The first one is service and product improvement. Pain points reflect deficiencies in services and products, suggesting areas for potential improvement for companies (Tao et al., 2019). Similarly, service and product innovation can also be achieved. Understanding unsatisfied customer needs provides direction for companies to innovate and develop new services or products that better address these needs and preferences (Zhang et al., 2021). The third application is market segmentation. Clustering enables companies to identify homogeneous subgroups with common characteristics related to pain points. This approach assists in effective market segmentation and customization of products or services tailored to individual customer needs (Wang et al., 2016). The last application is competitor analysis. By identifying and understanding competitors' pain points, companies can uncover their weaknesses. Analyzing these weaknesses can lead to greater differentiation from competitors (Tao et al., 2019; Salminen et al., 2022).

We develop a series of rules for the definition of pain points to provide valuable customer insights to companies. As previous studies have indicated, a pain point is not

doi:10.6342/NTU202402663

only a complaint but a sentence that contains actionable customer insights (Salminen et al., 2022; Lee et al., 2023). For example, “The waiter is bad” is not a pain point, while “The waiter is too mean” is a pain point since it indicates a bad service attitude. However, the sentence “The service attitude of the waiter is bad” is considered a pain point since it includes a specific aspect that can be improved. This informs service providers that the service attitude of the waiters needs improvement. In addition to customer dissatisfaction, we also include customer wish lists, as they convey actionable customer insights. For example, “I wish the bed could be larger” highlights a pain point regarding the bed size. Furthermore, since pain points are subjective, their feasibility and reasonableness may vary among different customers. We prioritize customers' feelings over the feasibility or reasonableness of pain points. For example, the sentence “There is no natural view outside the window in the room” is considered a pain point even if the hotel is located downtown with many buildings around. However, if a requirement does not cause dissatisfaction, it is not considered a pain point. For instance, the sentence “The breakfast is smaller than before, but I won’t be so full” mentions the decrease in meal size, but since the customer is satisfied, it is not a pain point. With this definition, we can ensure data quality for better model learning. We further illustrate this with some real examples from Chinese hotel reviews in Taiwan. For example, a Chinese review states, “一進房間還看到**大鏡子面對床 (room services) 早餐時段還說因人數過多需分流 (dining services)**. . . 真的

doi:10.6342/NTU202402663

完全沒有住大飯店的感覺," and the translated English version is, "As soon as you enter the room, you see **a large mirror facing the bed (room services)**. **During breakfast, they said that due to the large number of people, staggered dining is required (dining services)**... It really doesn't feel like staying in a luxury hotel at all." There are two pain point spans in the review: "a large mirror facing the bed," which belongs to room services, and "During breakfast, they said that due to the large number of people, staggered dining is required," which belongs to dining services. The pain point spans extracted are different from the aspects extracted in sentiment analysis. Our pain point spans consist of aspects as well as reasons for dissatisfaction. Therefore, a pain point span is a short sentence rather than just an aspect.

1.2 Research Motivation

In recent years, automated pain point analysis has addressed the shortcomings of traditional methods. Prior studies on automated pain point analysis mainly used text mining techniques. In these studies, pain points were extracted as keywords or entire sentences. However, keywords may not represent the whole pain points, leading to semantic ambiguity. Companies need more information to address these unsatisfied issues. Therefore, it is insufficient to extract only keywords as pain points. Moreover, even if entire sentences are extracted, there is often irrelevant information within these sentences. This requires further filtering for companies to accurately identify the pain

doi:10.6342/NTU202402663

points. Therefore, we aim to predict the pain point spans in each review. We use a sequence labeling model with BERT-BiLSTM-CRF layers to predict the spans using the {B, I, O} schemes.

Additionally, only some prior studies categorized pain points by type. Pain point categorization enables the evaluation of pain points at the category level. The category analysis can be compared across different companies, locations, times, and more, allowing companies to gain deeper insights into customer pain points. For example, the distribution of pain point categories in different companies can be compared for competitor analysis. In addition to cross-category analysis, companies can further enable different departments to focus on related issues by filtering specific pain point categories. As a result, we use a text classification model with BERT and Bi-LSTM layers to classify the category of each obtained pain point span.

1.3 Research Objective

In our study, we propose a Pain-point Extraction and Categorization (PEC) framework. There are two main objectives in the PEC framework. First, we aim to extract and highlight pain points from hotel reviews to enable a quick and better understanding of customer needs. Specifically, our focus is on identifying the shortest sub-sentences that include pain point expressions, which we refer to as "pain point spans" in our study. Second, we classify identified pain points into predefined categories corresponding to

doi:10.6342/NTU202402663

different areas. This classification allows for comparison across categories and helps in quickly identifying and resolving relevant issues.

doi:10.6342/NTU202402663

Chapter 2 Related work

2.1 Customer Pain Point Analysis

Previous studies on pain point analysis have defined pain points in various ways. According to Wang et al. (2016), pain points are not physical but emotional, arising from psychological gaps or dissatisfaction when customer expectations are not met. Pain points reflect customers' core concerns, primary interests, and emerging needs. In the studies by Salminen et al. (2022), pain points are identifiable problems that customers have experienced and that companies can address. They also offer actionable insights for companies. Additionally, in the studies by Lee et al. (2023), pain points stem from the emotions customers experience while using products or services. They indicate complaints that can be addressed through functional or procedural improvements, providing actionable insights.

When companies are aware of their customers' pain points, they can take specific actions to address them. Since pain points are based on real needs of consumers, the goal is to identify and meet these needs to gain their favor. By understanding customer needs, companies can identify deficiencies in services and products in the current market and make further improvements (Tao et al., 2019). Besides, these deficiencies, along with customer unsatisfied needs, provide direction for innovation in new services and products

doi:10.6342/NTU202402663

(Zhang et al., 2021). Moreover, by identifying customers with similar pain points, companies can conduct market segmentation to better address them (Wang et al., 2016). With these actions, companies can boost customer loyalty (Li et al., 2013; Mutlubaş, 2023), influence the purchase intentions of new customers through positive feedback and fewer negative reviews, and result in a better brand image and increased sales volume (Holjevac et al., 2010; Chen and Tabari, 2017). Additionally, companies can identify the pain points of their competitors to understand their weaknesses. By analyzing competitors' weaknesses, companies can develop services and products that are superior to those of their competitors, leading to greater differentiation. For example, Tao et al. (2019) compare the sentiment value and pain point indices, which represent user discomfort, across competitors. Salminen et al. (2022) conduct pain point profiling for each brand by calculating the frequency of pain points in different categories.

Traditional methods of identifying pain points include interviews, observations, focus groups, and cross-sectional surveys. These methods may face constraints related to budget, time limitations, data sample size, human biases, and challenges in illustrating needs (Salminen et al., 2022). Manual participation, which often contributes to these constraints, also results in high costs and low accuracy. Consequently, there is a growing recognition of the potential for automated methods to enhance the efficiency and effectiveness of customer pain point analysis (Ma and Sun, 2020).

doi:10.6342/NTU202402663

2.2 Previous Studies on Pain Point Analysis

There are four studies that have used automated text mining methods in pain point analysis. These studies all extracted pain points from customers for analysis. In the first study, Wang et al. (2016) extracted pain points from a poll. Customers participated in a poll to vote on various pain points related to products. The pain point choices listed in the poll were then abbreviated; for example, the pain point "USB connection goes wrong" was abbreviated as "USB." The data collected included each user's votes for multiple pain points. After the extraction of pain points, they conducted customer segmentation. A biclustering algorithm was applied to segment the customers. This process identified groups of homogeneous customers who exhibited similar characteristics regarding the sets of pain points. In the second study, Tao et al. (2019) extracted pain points from hotel reviews. They used TF-IDF to extract features from online comments. These features were then manually filtered to identify pain point-related features, and similar words were grouped together as pain points. They further calculated the score for obtained pain points. They adopted dependency parsing to identify pairs consisting of a feature word and an emotional word. Sentiment analysis was then used to calculate the sentiment value of each pain point. The pain point index was calculated by exponentiating the sentiment value; a higher pain point index indicated greater user discomfort with the feature. Finally, they conducted competitor analysis by comparing the sentiment values and pain point

doi:10.6342/NTU202402663

indices of different companies to identify relative strengths and weaknesses. In the third study, Salminen et al. (2022) extracted pain points from tweets. They used a binary classification model to identify sentences that included pain points. Then they classified identified pain points into predefined categories using a multi-class model. They also conducted competitor analysis, calculating the frequency of pain points by brand and category across different competitors as pain point profiling. The prevalence of various pain points among brands was then displayed. In the fourth study, Lee et al. (2023) focused on accurately identifying pain points using an unsupervised method. Their methods started with sentiment analysis and topic modeling. Pain points were analyzed within negative reviews, and topics within the reviews were identified through topic modeling. After that, gradient-based attribution was adopted to calculate token attribution for the previous two tasks. Finally, they conducted post-processing by using dependency parsing to find nouns associated with words with high attribution scores. The nouns identified were considered pain points.

We can categorize some of the methods used in these studies. In the methods of pain point extraction, the approaches include supervised learning (Salminen et al., 2022), unsupervised learning (Tao et al., 2019; Lee et al., 2023), and manual participation (Wang et al., 2016; Tao et al., 2019). Supervised learning can precisely extract pain points with consistent patterns, but this method requires labeled data, which needs human annotation

doi:10.6342/NTU202402663

if labels are not available. Pain points extracted from unsupervised learning are often unexpected and usually require multiple methods for more accurate extraction. To identify pain points that conform to a specific set of rules in our pain point definition, we adopt supervised learning in our study to ensure the model learns the definition of our pain points. In the type of pain point extracted, these studies use keywords (Wang et al., 2016; Tao et al., 2019; Lee et al., 2023) and whole sentences (Salminen et al., 2022). Pain points extracted through unsupervised learning or manual extraction often manifest as single keywords. However, these keywords can sometimes be ambiguous in meaning and lack actionable insights. For instance, consider the pain point "noise." Without additional context, it's difficult to determine the specific source of the noise—whether it originates from a particular location, person, or product. Even if we specify that the noise pertains to a product, we still don't know if it indicates a malfunction causing noise or simply that the product is too loud. In the previous study using supervised learning, pain points are extracted as whole sentences (Salminen et al., 2022). However, sentences often contain unrelated information. Therefore, the goal in our study is to extract the shortest expression within the sentences that captures the pain point. For example, from the sentence "The restaurant is very beautiful, the food inside is also delicious. But the waiters do not smile at all," the extracted pain point expression would be "the waiters do not smile at all." This concise expression allows us to grasp the core issue and derive actionable insights for

doi:10.6342/NTU202402663

improvement. In the categories of pain points, Tao et al. (2019) classified hotel pain points into hotel hygiene, cost performance, hotel location, hotel service, and noise insulation, while Salminen et al. (2022) classified pain points of brands into company image, customer service, product feature or quality, service quality or failure, and operational issues. Since pain point expressions used in our study are short sub-sentences, categorizing each pain point helps companies better understand performance across different categories. Moreover, it assists individuals in focusing on specific pain point categories through the use of category filtering. Table 1 shows the summary of previous studies on pain point analysis.

In summary, previous studies extract pain points as keywords (Wang et al., 2016; Tao et al., 2019; Lee et al., 2023) or whole sentences (Salminen et al., 2022), which can be ambiguous or contain unrelated information. We extract concise pain point expressions that clearly convey the pain points. Additionally, we follow two of the studies to categorize pain points for cross-category analysis and allow filtering for individual-specific insights.

doi:10.6342/NTU202402663

Table 1: Summary of previous studies on pain point analysis

Studies	Methods of Pain Point Extraction	Use of Sentiment Analysis	Type of Pain Point Extracted	Classification of Pain Point Category
Wang et al. (2016)	Manual Participation	No	Keyword	No
Tao et al. (2019)	Unsupervised + Manual Participation	Yes	Keyword	Yes
Salminen et al. (2022)	Supervised	No	Whole sentence	Yes
Lee et al. (2023)	Unsupervised	Yes	Keyword	No
Our study	Supervised	No	Pain point expression	Yes

doi:10.6342/NTU202402663

Chapter 3 Methodology

3.1 Problem Formulation

Given a review $r_i = [w_{i1}, w_{i2}, \dots, w_{in}]$, our goal is to predict the pain point spans $Y_i = [y_{i1}, y_{i2}, \dots, y_{in}]$ where n is the number of tokens in review r_i and the k-th label $y_{ki} \in \{B-P, I-P, O\}$. Each pain point span starts with a B-P tag and is followed by I-P tags for subsequent tokens, while tokens not part of a pain point are denoted by O tags.

For all extracted pain point spans $P_i = [p_1, p_2, \dots, p_l]$ in review r_i, we predict the corresponding categories $C_i = [c_1, c_2, \dots, c_l]$. The k-th pain point span $p_k = [w_1, w_2, \dots, w_m]$ consists of m tokens (where $1 \leq m$), and l denotes the total number of pain point spans in review r_i (where $0 \leq l$).

3.2 Overview of the PEC Framework

We propose a **P**ain-point **E**xtraction and **C**ategorization (PEC) framework. The objective of the PEC framework is to identify customers' unsatisfied pain points and their corresponding categories. We construct a complete workflow for pain point analysis, comprising two phases: Pain Point Extraction (PPE) and Pain Point Categorization (PPC). In the pain point extraction phase, we use sequence labeling to extract pain point spans from online reviews. In the pain point categorization phase, we use a multi-class classification model to predict the predefined categories of each extracted pain point.

doi:10.6342/NTU202402663

Figure 1 provides a pipeline for the PEC framework.

We apply the pretrained language model RoBERTa (Liu et al., 2019) for Chinese (chinese-roberta-wwm-ext) to acquire representations of each token in reviews. Given a review $r_i = [w_{i1}, w_{i2}, \dots, w_{in}]$, we obtain hidden states h_{ik}^{BERT} as review embeddings for each token w_{ik} in review r_i. In the pain point extraction phase, the inputs are the review embeddings from the BERT model. The outputs are sequence labels for each token w_{ik} in review r_i. For sequence labeling, we adopt the {B, I, O} scheme to represent the locations of pain point spans, indicating whether a token belongs to the Beginning, Inside, or Outside of pain point spans. Each pain point span starts with a B-P tag, followed by I-P tags for the rest of the tokens, while non-pain point tokens are represented by O tags. In the pain point categorization phase, we train another multi-class classification model independently to predict the categories of the obtained pain point spans from the previous phase. During training, since the categories of pain points are related to the entire context of the reviews, we input the whole review instead of each separate pain point span. We add special tokens $[Pl_{start}]$ and $[Pl_{end}]$ at the beginning and end of the l-th pain point span in review r_i, and extract the hidden states of the$[Pl_{start}]$ as the representations of the l -th pain point span for further classification. The idea of adding special tokens to represent each pain point span is inspired by Soares et al. (2019). All obtained pain points are specific and contain actionable insights, implying potential areas for improvement.

doi:10.6342/NTU202402663

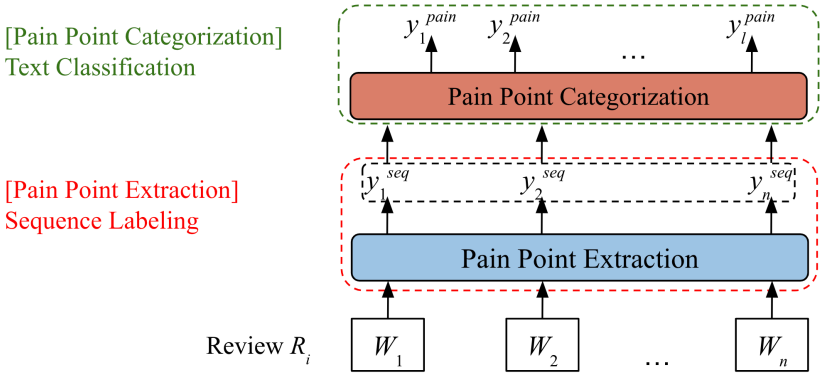


Figure 1: Pipeline of the PEC framework

3.3 Pain Point Extraction (PPE)

The objective of PPE is to use a sequence labeling model to find the locations of all pain point spans in reviews. Each review may contain zero to many pain point spans. Each pain point span is the shortest sub-sentence while retaining the meaning of the whole pain point. We adopt the {B, I, O} scheme to represent pain point spans. A pain point span starts with a B-P token and is followed by continuous I-P tokens. O represents non-pain point tokens. For example, Table 2 shows the labels of each token in the sentence "The soup is salty, and the waiter is mean."

Table 2: An example of sequence labels

Label	B-P	I-P	I-P	I-P	O	O	B-P	I-P	I-P	I-P	O
Token	The	soup	is	salty	,	and	the	waiter	is	mean	.

We take the hidden states h_{BERT} from the BERT model as the input for the bi-directional Long Short-Term Memory (Bi-LSTM) layer. The output hidden states from all time steps of the Bi-LSTM are then passed to the Conditional Random Field (CRF)

doi:10.6342/NTU202402663

layer (Sutton and McCallum, 2007). Finally, we obtain the output sequence $Y_i = [y_{i1}, y_{i2}, \ldots, y_{in}]$ in review r_i, where the k-th label $y_{ik} \in \{B-P,\ I-P,\ O\}$. Figure 2 shows the model architecture of our sequence labeling model.

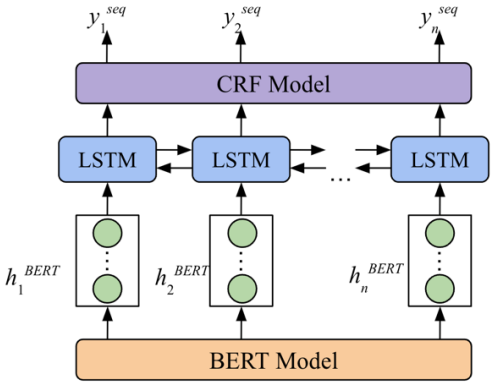


Figure 2: Model architecture of our sequence labeling model

We adopt the Bi-LSTM model to acquire long-term contextual information for each review. LSTM networks are a type of recurrent neural network (RNN) that can capture temporal dependencies and long-term information in sequential data. Additionally, LSTMs mitigate the problem of vanishing gradients in traditional RNNs. Suppose there are h hidden units, the batch size is n, and the number of inputs is d. The LSTM unit contains a memory cell $C_t \in R^{n\times h}$ at time t, which can maintain the information over time. Besides, there are also three gates in the LSTM unit, including the input gate $I_t \in R^{n\times h}$, forget gate $F_t \in R^{n\times h}$, and output gate $O_t \in R^{n\times h}$. The input is $X_t \in R^{n\times d}$, and the hidden state of the previous time step is $H_{t-1} \in R^{n\times h}$. The flow of information is:

$$I_t = \sigma(H_{t-1}W_{hi} + X_tW_{xi} + b_i)$$

$$F_t = \sigma\left(H_{t-1}W_{hf} + X_tW_{xf} + b_f\right)$$

doi:10.6342/NTU202402663

$$O_t = \sigma(H_{t-1}W_{ho} + X_tW_{xo} + b_o)$$

$$C_t = F_t \odot C_{t-1} + I_t \odot \tanh(H_{t-1}W_{hc} + X_tW_{xc} + b_c)$$

$$H_t = O_t \odot tanh(C_t)$$

where W are model parameters and b are bias parameters.

A CRF is a probabilistic graphical model often used in Natural Language Processing (NLP). It is a variant of the Hidden Markov Model (HMM) but makes independence assumptions among y instead of x. For the sequence labeling task, we adopt a linear-chain CRF rather than a general CRF since the data is sequence with the input length equal to the output length. A CRF models the conditional probability of a label sequence y given an input sequence x with the following formulation:

$$p(y|x) = \frac{1}{Z(x)} exp\{\sum_{k=1}^{K} \lambda_k f_k(y_t, y_{t-1}, x_t)\}$$

$$Z(x) = \sum_{y} exp\{\sum_{k=1}^{K} \lambda_k f_k(y_t, y_{t-1}, x_t)\}$$

where $Z(x)$ is the normalization factor, λ_k are model parameters, f_k are feature functions capturing the input and output sequence, k is the feature function index, and K is the number of feature functions.

The architecture of Bi-LSTM layers followed by CRF layers for sequence tagging was first introduced by Huang et al. (2015). The Bi-LSTM-CRF model can efficiently use past and future input features with a Bi-LSTM layer and sequence-level tag

doi:10.6342/NTU202402663

information with a CRF layer. After the introduction of the BERT model, BERT-CRF and BERT-LSTM-CRF models have been adopted in studies to acquire the information of sequence labels (Jiang et al., 2019; Souza et al., 2019). For the loss function, we use negative log-likelihood (NLL) for the model with CRF layers, and cross-entropy for the model without CRF layers. The formula for NLL is calculated as follows:

$$NLL\ loss = -(\sum_{t=1}^{n} U(y_t|x) + \sum_{t=1}^{n} T(y_t|y_{t-1}) - logZ(x))$$

$$Z(x) = \sum_{y'} \exp\left(\sum_{t=1}^{T} U(y'_t|x) + \sum_{t=1}^{T-1} T(y'_{t+1}|y'_t)\right)$$

where $\sum_{t=1}^{T} U(y'_t|x)$ means the emission score, $\sum_{t=1}^{n} T(y_t|y_{t-1})$ means the transition score, and $logZ(x)$ means the normalization factor. Each emission score is a score indicating how likely to assign the label y_t at position t given the input sequence x. Each transition score captures the probability of transitioning for moving from the label y_{t-1} to the label y_t across the sequence. The partition function ensures the probabilities of all possible sequences sum to one.

In order to ensure the output format of label Y conforms to the rule of {B, I, O} schemes, where a span starts with a B-P token and is followed by I-P tokens, we further adopt post-processing for label Y. There are four rules in our post-processing. First, since a span must start with B-P, then (1) if a span starts with I-P, convert the first token to B-P. Second, a single I-P token is not allowed, but a single B-P token is allowed, then (2) a

doi:10.6342/NTU202402663

single I-P token is converted to O. Third, continuous B-P tokens are not allowed, then (3) the second continuous B-P token is converted to I-P, also (4) continuous spans are allowed. Table 3 shows the post-processing result using these four rules. By this method, we can know if the model is well trained with the rule of {B, I, O} schemes. For those well-trained models in sequence labeling, the performance will be the same after post-processing. On the contrary, we can further improve the performance if the data is not enough or the model is not well trained. Figure 3 shows the overall architecture of our pain point extraction model.

Table 3: An example of post-processing

Rule	(1)				(3)				(4)		(2)	
Before	I-P	I-P	O	B-P	B-P	O	B-P	I-P	B-P	O	I-P	O
After	B-P	I-P	O	B-P	I-P	O	B-P	I-P	B-P	O	O	O

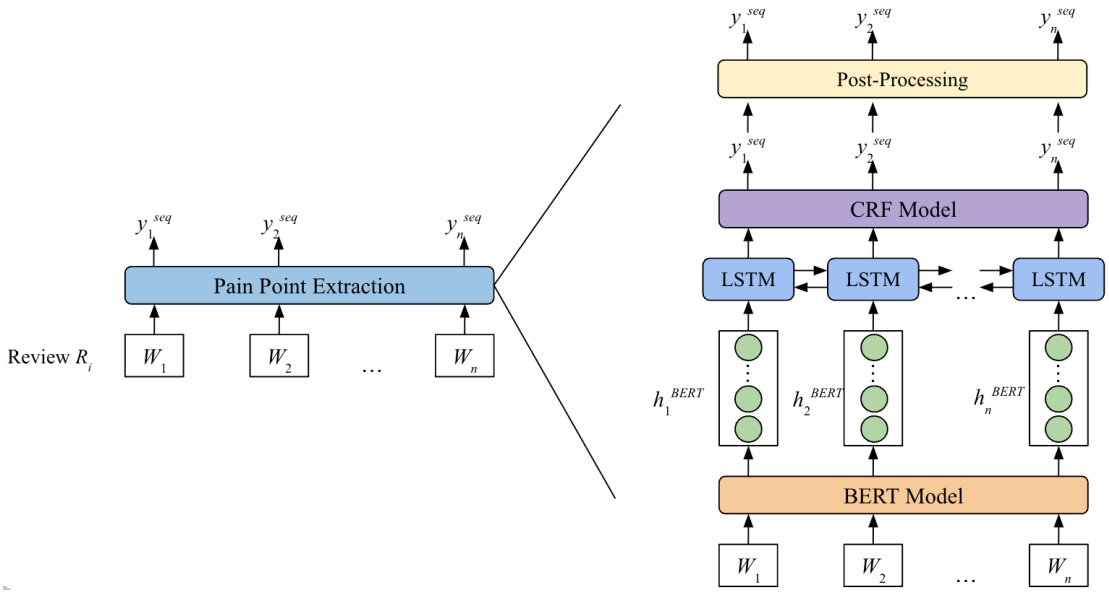


Figure 3: Overall model architecture of our PPE model

doi:10.6342/NTU202402663

3.4 Pain Point Categorization (PPC)

The objective of PPC is to use a multi-class classification model to classify each identified pain point into predefined categories. The model in this phase is trained independently of the models used in the pain point extraction phase, and the data is human-labeled instead of identified pain point spans. This setup allows us to predict each identified pain point into one of the predefined categories, forming a data pipeline for pain point analysis. Since there are zero to many pain point spans in a review, we predict one to many categories in each review. We do not predict any categories if there are no pain point spans.

We use similar model architecture in PPE, but without the CRF layers. The model architecture includes BERT layers and Bi-LSTM layers. We use cross-entropy as the loss function for the text classification. The inputs to the model are pain point spans, while the outputs are their corresponding categories. Traditionally, text classification is performed on each sentence separately, as shown in Figure 4.

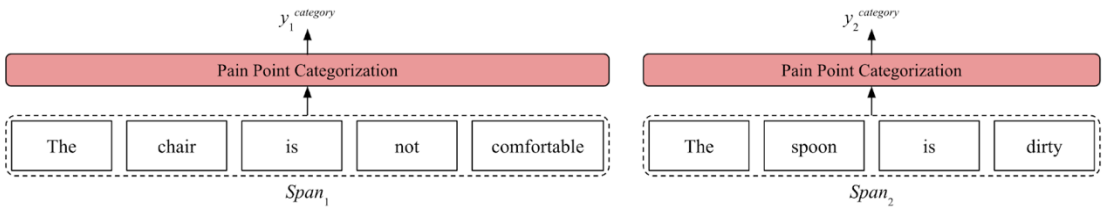


Figure 4: An example of traditional text classification

doi:10.6342/NTU202402663

However, since pain point categories are context-dependent, we should consider the whole context of reviews. For example, the category of the sentence "The chair is not comfortable" cannot be determined without knowing the whole context. It could be referring to a chair in a room, a restaurant, or any other place in a hotel. Thus, we pass the entire reviews into the model rather than just pain point spans. To obtain the representation of each pain point span, special tokens $[Pl_{start}]$ and $[Pl_{end}]$ are added at the beginning and end of the l-th pain point for BERT embedding, with the hidden states of the $[Pl_{start}]$ token used as the representation of the l-th pain point. With the use of special tokens, we can input the whole review and then get the representation of each pain point span in the review. Figure 5 shows the process of using special tokens.

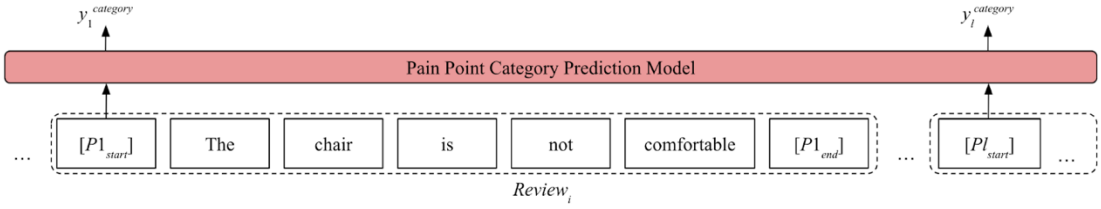


Figure 5: An example of text classification using special tokens

Finally, Figure 6 shows the entire model architecture for pain point categorization. We illustrate that if the second and third tokens are predicted as a pain point span, we add a start token and an end token as the first and fourth tokens. Then we can get the hidden state of the first token, which is a start token, as the representation of the pain point span for classification.

doi:10.6342/NTU202402663

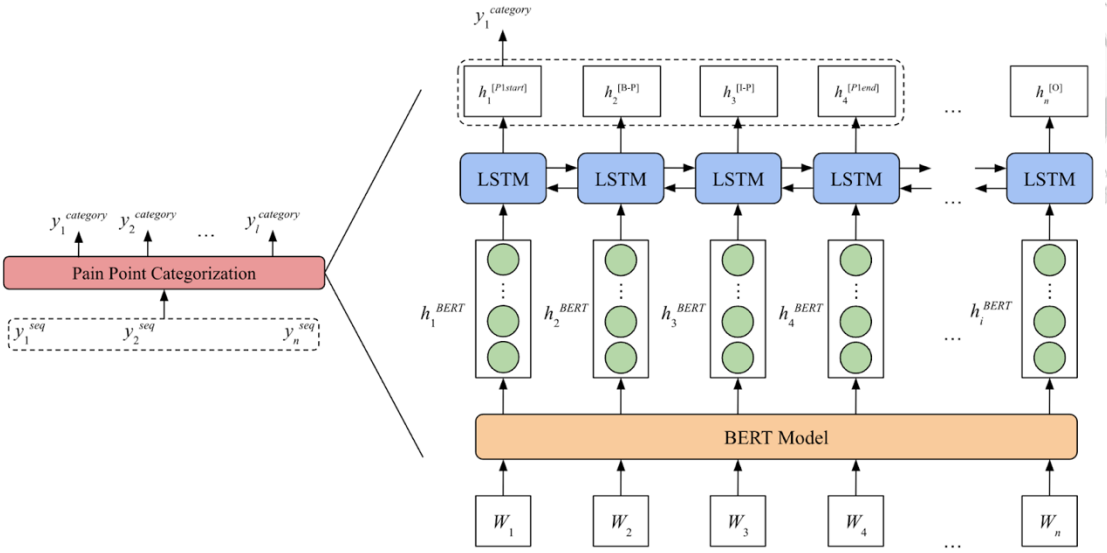


Figure 6: Overall model architecture of our PPC model

doi:10.6342/NTU202402663

Chapter 4 Empirical Experiments

4.1 Data Collection

We collected customer reviews for five high-class hotels under a large group in Taiwan, using Google reviews as our training data, which include reviews from Tripadvisor. The reason for choosing hotel reviews is the diverse range of services offered, including rooms, food, swimming pools, and more. Consequently, customer pain points vary significantly within hotels. For example, Li et al. (2013) found that customers in China were most dissatisfied with the bed, reception, room size, and decoration. In our study, we classify pain points into seven categories, corresponding to different areas within hotels. These predefined categories allow each department to filter and address their related pain points, resulting in more efficient issue resolution and improved customer satisfaction. The categories include (1) Room Services, which encompasses all items and services related to the room; (2) Check-In/Out Services, which covers check-in/out processes, reception, valet parking, and interactions with lobby staff; (3) Dining Services, which includes dishes in the restaurant, interactions with service staff, phone services, breakfast, dinner, bar services, and weddings; (4) Recreational Facilities and Services, which involves facilities and interactions with service staff in recreational areas such as the spa, gym, and swimming pool; (5) Public Facilities and Services, which covers

doi:10.6342/NTU202402663

the lobby, public areas, elevators, smoking areas, parking lots, and interactions with service staff in these areas; (6) Customer and Reservation Services, which includes phone and online customer service, reservation services, and the online website; (7) Others, which covers areas not included in the above categories, such as staff training, pricing, lost-and-found management, surrounding environment, and transportation.

To obtain the pain point spans and their corresponding categories, a team of two people manually labeled the data in two steps. For pain point extraction, we identified the pain points in each review, extracting the shortest spans that fully express the customers' unsatisfied needs. There can be zero to many pain point spans in a single review. For pain point categorization, each identified pain point span was classified into one of the predefined seven categories. The second rater joined after the first rater had labeled more than 2,000 reviews. We provided the second rater with a guideline for data annotation. The guideline includes the definition of pain points, the definition of pain point categories, the rules of annotation, and examples of pain points and non-pain points.

To ensure the label consistency, we applied Cohen's kappa score (McHugh, 2012) to measure interrater agreement. The formula for Cohen's kappa score (κ) is:

$$\kappa = \frac{P_O - P_E}{1 - P_E}$$

where P_O is the observed agreement among raters, and P_E is the expected agreement by chance. The formulas for P_O and P_E are:

doi:10.6342/NTU202402663

$$P_O = \frac{\sum_{i=1}^{k} n_{ii}}{N}$$

$$P_E = \sum_{i=1}^{k} p_i \cdot q_i$$

where n_{ii} is the number of data points for which each rater assigns the same category i, k is the number of categories, N is the total number of data points rated, p_i is the proportion of all assignments made by the first rater to category i, and q_i is the proportion of all assignments made by the second rater to category i.

Since the identified pain points themselves may differ among raters, we do not measure the agreement between pain point categories from identified pain point spans. We only measured the agreement on the extraction of pain point spans to ensure both raters had the same definition of pain points. The evaluation of pain point spans is at the token level. It is challenging to evaluate at the span level since the identified pain points may not be identical. We evaluated interrater reliability three times. For each evaluation, the second rater labeled 10, 11, and 54 reviews respectively. After each measurement, we assessed the agreement on the same data between the two raters before proceeding to the next evaluation. Cohen's kappa scores for the three evaluations were 0.4448, 0.6938, and 0.7376 respectively. According to McHugh (2012), a score between 0.61 and 0.8 is considered substantial, and a score greater than 0.8 is considered perfect agreement. Additionally, since the pain point spans are long sub-sentences rather than short entities,

doi:10.6342/NTU202402663

there can be differing opinions on sentence breaks. There is no absolute answer as to what is correct, so our goal is to maintain consistency in labeling. Although the score does not reach 0.8, it is acceptable to us.

Finally, we collected 5,586 reviews. Table 4 shows the number of reviews among different ratings, and Table 5 shows the number of pain points in each category. Since our goal is to identify customer pain points, we believe that reviews with high ratings are likely to contain fewer pain points. Therefore, we only collected a limited number of 5-star reviews. Figure 7 shows the distribution of collected review lengths in the training data. Table 6 shows the statistics of review lengths, pain point span lengths, and the number of pain point spans in a review in the training data. The review lengths range from 1 to 499, with an average length of 90.9 and a standard deviation of 103.3. This indicates that the review lengths vary significantly, which may increase the difficulty of model learning in sequence labeling. However, the lengths of the pain point spans do not vary as much, which may increase the stability of the model.

Table 4: Number of reviews among different ratings

Rating	Number of Reviews
1 - 1.9	1157
2 - 2.9	621
3 - 3.9	1482
4 - 4.9	1942
5	384
Total	5586

doi:10.6342/NTU202402663

Table 5: Number of pain points among different categories

Category	Number of pain points
Room service	2080
Check-in/out service	1368
Dining services	3285
Recreational facilities and services	612
Public facilities and services	649
Customer and reservation services	262
Others	813
Total	9069

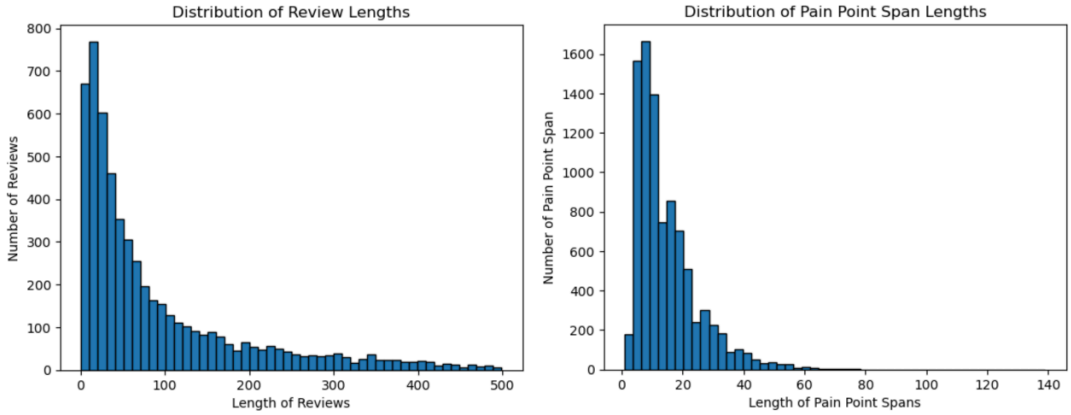


Figure 7: Distribution of review lengths and pain point span lengths

Table 6: Statistics of review and pain point span

Statistics	Review Length	Pain Point Span Length	# of Pain Point Spans in a Review
Mean	90.9	14.8	1.6
Standard Deviation	103.3	10.5	2
Minimum	1	1	0
Median	48.5	12	1
Maximum	499	139	16

doi:10.6342/NTU202402663

4.2 Evaluation Metrics

As our model consists of two phases, we evaluate each phase separately. Typically, sequence labeling tasks are evaluated using sequence evaluation, which is called exact match in our study. In this method, for each predicted entity or span, we consider it a true positive if every token in the entity or span matches the ground truth. For named entity recognition (NER) tasks, where common entities include time, person, organization, and location, the exact match is appropriate since the entities are often short. However, since we aim to predict pain point spans in long reviews, the pain point spans are short sentences with longer lengths than common entities. Additionally, different people may not have the same understanding of the breaks in the pain point spans. Thus, evaluating with only the common sequence evaluation is not sufficient. We adopt a method used by Jiang (2022), evaluating the correct ratio in each span, which is called span match in our study, to better fit our scenario. Similar to exact match, we calculate precision and recall for each span. However, we do not require the model to predict all tokens correctly. Instead, we calculate the correct ratio in each predicted/true span and then average these ratios by the number of predicted/true spans. We also calculate the ratio of spans with zero precision/recall, which are termed the false span ratio and missing span ratio, respectively. Table 7 shows an example of a sequence with true labels and predicted labels.

doi:10.6342/NTU202402663

Table 7: An example of a sequence with true labels and predicted labels

True	B-P	I-P	O	B-P	I-P	I-P	O	B-P	O	B-P
Predict	B-P	I-P	O	B-P	I-P	O	B-P	I-P	I-P	O

Assuming B-P as the beginning of a pain point span and I-P representing the tokens in the rest of the pain point span, there are four true pain point spans and three predicted pain point spans. For overall precision, we average the precision of each predicted span, and for overall recall, we average the recall of each true span. Therefore, we can calculate the span precision, false span ratio, span recall, and missing span ratio as follows:

$$precision = \frac{1 + 1 + \frac{1}{3}}{3} = \frac{7}{9} \cong 0.77$$

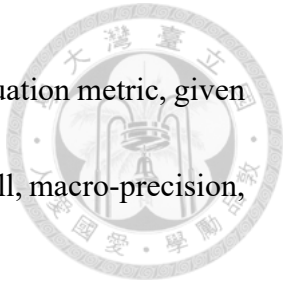
$$false\ span\ ratio = \frac{0}{3} = 0$$

$$recall = \frac{1 + \frac{2}{3} + 1 + 0}{4} = \frac{2}{3} \cong 0.66$$

$$missing\ span\ ratio = \frac{1}{4}$$

Additionally, we also loosen the rule of exact match by allowing for mistakes of one or two tokens at the front and end of a span, a method called fuzzy match. We refer to these as "fuzzy match one word" for allowing mistakes of one token and "fuzzy match two words" for allowing mistakes of two tokens. Therefore, we have exact match, fuzzy match, and span match for span-level evaluation. We also calculate token-level evaluation, which assesses only the correctness of individual tokens.

doi:10.6342/NTU202402663



For pain point categorization, we use macro-F1 as our main evaluation metric, given that it is a multi-class classification task. We also calculate macro-recall, macro-precision, and accuracy.

4.3 Experimental Procedure

For all experiments, we conduct 10-fold cross-validation to obtain more reliable results. Our experimental settings include linear learning rate scheduling with a 10% warm-up for better stability. In the sequence labeling model, we use different learning rates for BERT, Bi-LSTM, and CRF layers. Since fine-tuning BERT layers requires a small learning rate (Sun et al., 2019), and CRF layers need a much higher learning rate for convergence (Su, 2020), we use a learning rate of 2e-5 for the BERT layers, 1e-4 for the Bi-LSTM layers, and 2e-3 for the CRF layers. For pain point extraction, we use a learning rate of 3e-5. Other settings include training the model for a maximum of 20 epochs, using the AdamW optimizer, a batch size of 4, a BERT embedding dimension of 768, a Bi-LSTM hidden dimension of 768, a review representation length of 512, and a dropout rate of 0.1.

4.4 Evaluation of Pain Point Extraction

Table 8 shows a comparison of different model architectures for pain point extraction. The results indicate that the model with only BERT layers performs well, and adding additional layers further improves performance. Specifically, models with CRF

doi:10.6342/NTU202402663

layers perform significantly better than those without them. In addition, models with Bi-GRU or Bi-LSTM layers show slight differences in performance, but both perform better in span F1 compared to the model with only BERT layers. Our proposed PPE model with BERT-BiLSTM-CRF layers achieves the best span F1 score of 0.8037. The result indicates accurate identification of approximately 80% of the words within each pain point span. Models with CRF layers do not require post-processing since they effectively adhere to the {B, I, O} tagging scheme rules. Our post-processing method provides marginal improvements in models without CRF layers. Furthermore, the exact match F1 score is 0.4659, indicating that nearly half of the pain point spans are predicted exactly with all words correctly identified. The model performance improves when we allow for mistakes of one or two words at the beginning and the end. But the improvement in fuzzy F1 compared to exact F1 is not significant, suggesting that incorrect predictions are not solely due to minor issues like sentence breaks. Different model architectures show similar missing span ratios but vary a lot in false span ratios. It shows the importance of reducing the false span ratio for improving model performance.

doi:10.6342/NTU202402663

Table 8: Evaluation results of the pain point extraction model

Model	Token F1	Exact Match F1	Fuzzy F1 one word	Fuzzy F1 two words	Span Precisi-on	False Span Ratio	Span Recall	Missing Span Ratio	Span F1 w/o post proce-ssing	Span F1
BERT	0.7646	0.4009	0.4421	0.4756	0.7517	19.66%	0.8080	**11.40%**	0.7715	0.7786
BERT+ Bi-GRU	0.7662	0.4297	0.4632	0.4979	0.7752	16.66%	0.8060	12.26%	0.7876	0.7902
BERT+ Bi-LSTM	0.7656	0.4199	0.4538	0.4862	0.7677	17.13%	0.8109	12.04%	0.7864	0.7885
BERT+ CRF	0.7710	0.4543	0.4760	0.5045	0.7751	15.32%	0.8277	11.74%	0.8001	0.8001
BERT+ Bi-GRU+ CRF	0.7725	**0.4659**	**0.4901**	**0.5198**	0.7791	15.05%	0.8281	11.71%	0.8027	0.8027
BERT+ Bi-LSTM+ CRF **(PPE)**	**0.7744**	0.4643	0.4873	0.5164	**0.7809**	**14.60%**	**0.8284**	11.61%	**0.8037**	**0.8037**

Additionally, we provide an example of a predicted pain point span from a Chinese review: "地點方便，門口即捷運。但**大廳燈光暗，給人昏昏的感覺 (public facilities and services)**。**浴室稍小了點 (room services)**。以性價比上勉強可以," with the English translation: "The location is convenient, right at the MRT entrance. However, **the lobby lighting is dim, giving a gloomy feeling (public facilities and services)**. **The bathroom is a bit small (room services)**. Overall, it's acceptable in terms of cost-performance ratio." The predicted example shows two pain point spans: "the lobby lighting is dim, giving a gloomy feeling" and "the bathroom is a bit small." This result demonstrates that our predicted pain point spans are short sentences rather than just

doi:10.6342/NTU202402663

aspects. The identified pain point spans also convey the complete meaning of customer dissatisfaction and the reasons behind it.

4.5 Effects of Multi-Task Learning in Pain Point Extraction

4.5.1 Multi-task Learning Architecture

According to Singh et al. (2022), adopting multi-task learning with an auxiliary task of sentiment analysis enhances the performance of the main task of complaint classification. The main task in their study is a binary classification of complaint, while the auxiliary task is a multi-class classification of sentiment (positive, neutral, negative). To acquire additional information related to pain points in our pain point extraction model, we integrate two auxiliary tasks into the sequence labeling model. With these auxiliary tasks, the model becomes a multi-task learning model. The auxiliary tasks are rating prediction and pain point existence classification. Our aim is for the model to learn the relationship between sentences and their corresponding ratings to identify pain points associated with different ratings. For example, reviews with 1-star ratings generally contain more pain points. Besides, if the model can determine whether a sub-sentence contains pain points, it can make more accurate sequence predictions. These additional information may help the learning process of pain point extraction, and further enhance the model performance.

doi:10.6342/NTU202402663

For rating prediction, we share the same BERT and Bi-LSTM layers with the other two tasks. We use the hidden states h^{BERT} from the BERT model, pass them through the Bi-LSTM layer, and apply pooling methods for the final rating prediction. Figure 8 shows the model architecture of rating prediction. There are two output formats for ratings. One is the actual number, and the task becomes a regression problem. The other one is the predefined categories for ratings. Ratings can be predicted in two formats: as an actual number (regression problem) or as predefined categories (multi-class classification problem). The first group contain ratings where $1 \leq rating \leq 2.5$. The second group contain ratings where $2.5 < rating \leq 3.5$. The third group contain ratings where $3.5 < rating \leq 5$. We classified these three groups and made the problem a sentiment analysis problem, where the first group is negative, the second group is neutral, and the third group is positive. Regardless of the format, this allows the model to learn the pain points corresponding to ratings. For instance, reviews with lower ratings are likely to contain more pain points, whereas higher-rated reviews are likely to have fewer pain points.

doi:10.6342/NTU202402663

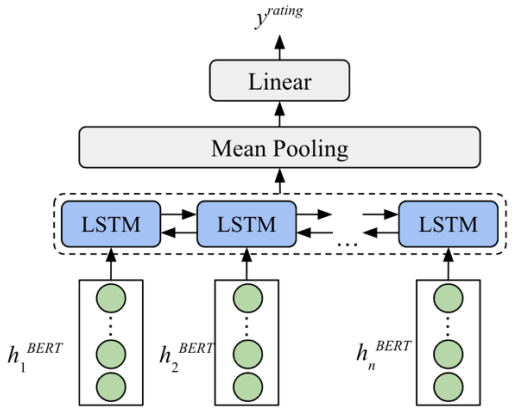


Figure 8: Model architecture of rating prediction

For pain point existence classification, we share the same BERT and Bi-LSTM layers with the other two tasks. Inputs are the hidden states h^{BERT} from the BERT model. Since most reviews have multiple sub-sentences and contain at least one pain point span, predicting the presence of pain points in entire reviews is unnecessary. Instead, we predict pain points in each sub-sentence. Special tokens $[Sj_{start}]$ and $[Sj_{end}]$ are added at the beginning and end of sub-sentence j for BERT embedding. The hidden states of the $[Sj_{start}]$ token, after the Bi-LSTM layers, are used as the representation for each sub-sentence. This representation is passed through a fully connected layer to predict the existence of pain points. If the model predicts no pain point in a sub-sentence, it labels all tokens in that sub-sentence with O tags. Figure 9 shows the model architecture of pain point existence classification.

doi:10.6342/NTU202402663

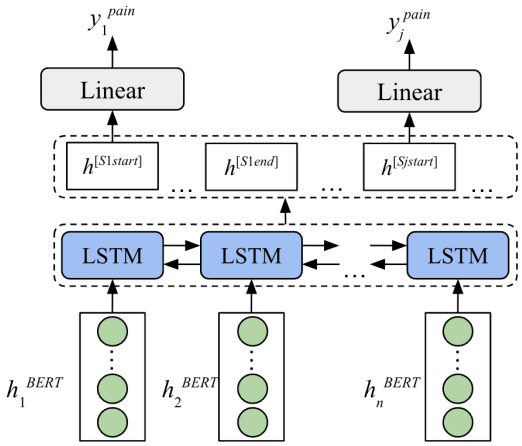


Figure 9: Model architecture of pain point existence classification

Figure 10 shows the complete architecture of the multi-task learning model with parameters shared in the BERT embedding and Bi-LSTM layers.

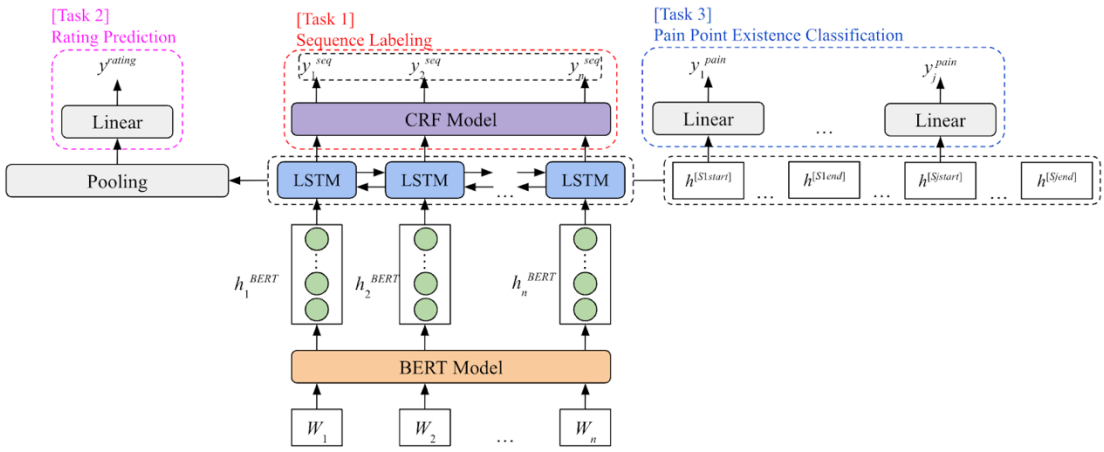


Figure 10: Model architecture of the multi-task learning model

The loss function for the sequence labeling task is negative log-likelihood (NLL) used in CRF layers. For rating prediction, we use cross-entropy for multi-class classification and mean squared error (MSE) for regression. Binary cross-entropy is used for the pain point existence classification. The combined loss function is:

doi:10.6342/NTU202402663

$$Total\ loss = loss_s \times weight_s + loss_r \times weight_r + loss_p \times weight_p$$

where s stands for the sequence labeling model, r represents the rating prediction model, and p is the pain point existence classification model. We set $weight_s$ as 0.7, and $weight_r$ and $weight_p$ are both set at 0.15.

4.5.2 Evaluation Results of Multi-task Learning

First, we compare the model performance in the main task (sequence labeling) using different prediction strategies, regression and classification, in rating prediction. The model architecture for sequence labeling is the same as in PPE, which includes BERT-BiLSTM-CRF layers. As we have found, post-processing is not needed for models with CRF layers, so we do not include a comparison on post-processing in the result. Table 9 shows that using regression achieves higher scores across all metrics except for span recall and missing span ratio. With the result, we can infer that the three rating categories oversimplify the task, and thus decrease the model performance. Therefore, we use regression for rating prediction in subsequent experiments.

Table 9: Evaluation results of tasks for rating prediction in multi-task learning

Model	Token F1	Exact Match F1	Fuzzy F1 one word	Fuzzy F1 two words	Span Precisi-on	False Span Ratio	Span Recall	Missing Span Ratio	Span F1
Classification	0.7502	0.4320	0.4572	0.4851	0.7335	16.74%	**0.8457**	**11.79%**	0.7857
Regression	**0.7594**	**0.4502**	**0.4713**	**0.5011**	**0.7885**	**14.32%**	0.7993	13.79%	**0.7939**

doi:10.6342/NTU202402663

To test the effectiveness of adopting multi-task learning, we conduct ablation experiments. Table 10 shows the results of removing one auxiliary task at a time. The results indicate that the single-task sequence labeling model (PPE) achieves the best results across all metrics except for fuzzy F1. Although incorporating pain point existence classification as auxiliary task further improves fuzzy F1, the model performance in other metrics decreases, especially span F1. Additionally, the multi-task learning model with both auxiliary tasks has the lowest span F1 score. This suggests that the single-task sequence labeling model (PPE) is sufficient, and the auxiliary tasks do not improve performance.

Table 10: Evaluation results of ablation experiments in multi-task learning

Model	Token F1	Exact Match F1	Fuzzy F1 one word	Fuzzy F1 two words	Span Precisi-on	False Span Ratio	Span Recall	Missing Span Ratio	Span F1
Single task **(PPE)**	**0.7744**	**0.4643**	0.4873	0.5164	**0.7809**	**14.60%**	**0.8284**	**11.61%**	**0.8037**
+ Pain point existence classification	0.7638	0.4584	**0.4897**	**0.5217**	0.7775	15.75%	0.8070	13.48%	0.7916
+ Rating prediciton	0.7619	0.4460	0.4688	0.4980	0.7657	15.94%	0.8240	12.00%	0.7936
+ All auxiliary tasks	0.7633	0.4556	0.4863	0.5160	0.7757	15.89%	0.8033	13.92%	0.7890

Although the auxiliary tasks provide additional information on pain points, the model performance decreases. One of the potential reasons is the different levels of

doi:10.6342/NTU202402663

granularity in these tasks. The main task sequence labeling operates at the token level, and focuses on detailed annotations within sentences. The auxiliary task rating prediction operates at the review level, and predicts overall ratings based on aggregated information. The other auxiliary task pain point existence prediction operates at the sub-sentence level to determine the presence of pain points in segmented text. As the levels of granularity differ, the three tasks may not synergize well. The multi-task learning model in Singh et al. (2022) had tasks at the same level, and both of the tasks aimed to classify categories. In our study, given that the main task is trained at the smallest level (token level), integrating additional information from higher levels (review and sub-sentence) can pose challenges. These differences in granularity show the need for design in multi-task learning frameworks to effectively leverage diverse types of information across different levels of text analysis.

4.6 Power of the PPE Model for Cross-Domain Inference

Since pain points may differ across domains, we evaluated our PPE model, trained on hotel reviews, on data from other domains. For example, pain points in restaurants typically relate to food, service, or other aspects, while product reviews usually focus on product defects. To this end, we collected 500 product reviews from MOMO, an online shopping site in Taiwan, and 300 restaurant reviews from Din Tai Fung, a well-known restaurant in Taiwan. For the product reviews, we gathered 100 reviews each for five

doi:10.6342/NTU202402663

keywords: earphones, power banks, seat cushions, car accessories, and razors. For the restaurant reviews, we selected two branches of Din Tai Fung in Taipei. Table 11 displays the number of product reviews by rating, while Table 12 shows the number of restaurant reviews by rating. The distribution of review lengths is shown in Figure 11. The length distribution of restaurant reviews is similar to that of the hotel reviews in the training data, while product reviews are generally shorter.

Table 11: Number of product reviews among different ratings

Rating	# of Product Reviews
1-1.9	150
2-2.9	141
3-3.9	199
4-5	10
total	500

Table 12: Number of restaurant reviews among different ratings

Rating	# of Restaurant Reviews
1	60
2	60
3	60
4	60
5	60
total	300

doi:10.6342/NTU202402663

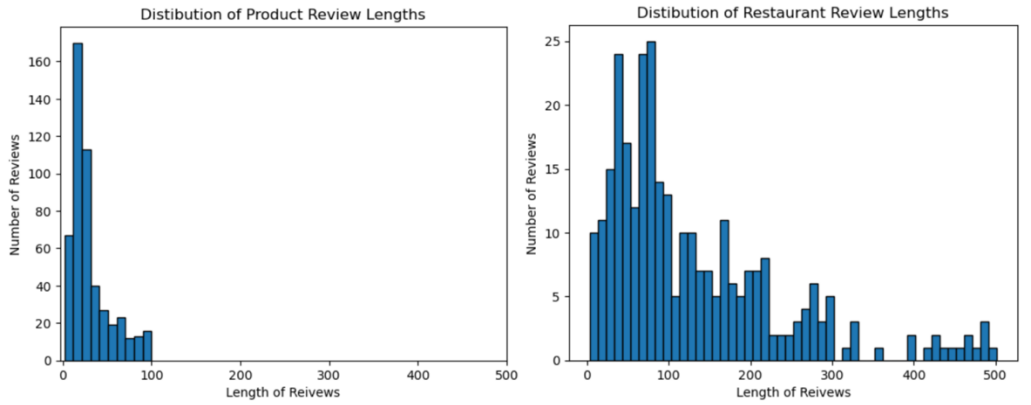


Figure 11: Distribution of product and restaurant review lengths

We evaluate the model trained on hotel reviews with a span F1 score of 0.8008 on the other two domains. The evaluation results are shown in Table 13. The span F1 score for product reviews was 0.8362. It exceeds the original hotel review performance. However, the span F1 score for restaurant reviews is lower, at 0.7192. The higher performance on product reviews may be attributed to the shorter and more concise sentences and pain point spans in these reviews, which makes it easier for the model to identify pain point spans. Conversely, although hotels also include restaurants, the lower performance on restaurant reviews may be due to the longer sentences and pain point spans in these reviews. As shown in Table 14, the difference in average pain point span length between the labeled data and predicted data is similar for product reviews but larger different for restaurant reviews. The lower performance is attributed to the larger difference in pain point span length.

doi:10.6342/NTU202402663

Table 13: Evaluation results of different domains for the PPE model

Domain	Token F1	Exact Match F1	Fuzzy F1 one word	Fuzzy F1 two words	Span Precisi-on	False Span Ratio	Span Recall	Missing Span Ratio	Span F1
Product	0.8263	0.4855	0.5070	0.5402	0.8359	11.59%	0.8364	9.45%	0.8362
Restaurant	0.7058	0.3247	0.3436	0.3746	0.7487	21.18%	0.6912	17.72%	0.7192
Hotel	0.7597	0.4515	0.4735	0.5028	0.7622	16.86%	0.8437	9.16%	0.8008

Table 14: Statistics of labeled and predicted pain point span

Domain	Product		Restaurant	
	Labeled Pain Point Span Length	Predicted Pain Point Span Length	Labeled Pain Point Span Length	Predicted Pain Point Span Length
Count	605	815	538	326
Mean	13.37	10.34	18.76	11.57
Standard Deviation	9.24	7.14	13.11	9.48
Minimum	2	1	1	1
Median	11	9	15	9
Maximum	73	57	85	61

4.7 Evaluation of Pain Point Categorization

Table 15 presents a comparison of different inputs and model architectures for pain point categorization. Among models using only pain point spans, the BERT model achieves the best macro-F1 score of 0.7759. For models using special tokens, our proposed PPC model achieves the best macro-F1 score of 0.8143. The two results show that incorporating full-context information through special tokens improves model performance. Among the two model architectures, adding Bi-LSTM layers results in a macro-F1 score 0.0005 lower for the input with only pain point spans and 0.0009 higher

doi:10.6342/NTU202402663

for the input with special tokens. These results suggest that although adopting Bi-LSTM layers achieves the best result, the improvement is marginal. The model using only BERT layers can perform similarly to the PPC model.

Table 15: Evaluation results of different inputs for the PPC model

Inputs	Model	Macro Precision	Macro Recall	Macro F1	Accuracy
Input with only pain point spans	BERT	0.7790	0.7744	0.7759	0.8185
	BERT+ Bi-LSTM	0.7846	0.7690	0.7754	0.8183
Use of special tokens	BERT	0.8195	**0.8121**	0.8134	0.8630
	BERT+ Bi-LSTM **(PPC)**	**0.8241**	0.8097	**0.8143**	**0.8643**

doi:10.6342/NTU202402663

Chapter 5 Conclusion

5.1 Contribution

We propose the PEC framework, a comprehensive pain point analysis flow designed for companies to identify pain point expressions from customer reviews and assign corresponding categories to each identified expression. We utilize a sequence labeling model for pain point extraction, a novel approach not previously employed in this context. Unlike previous studies that have used keywords (Wang et al., 2016; Tao et al., 2019; Lee et al., 2023) or whole sentences (Salminen et al., 2022) for pain point extraction, our method captures pain point expressions, preserving the complete information while excluding unrelated details. We explore the potential of multi-task learning in pain point extraction. Although multi-task learning did not improve results in our study, it remains a promising avenue, especially when tasks are aligned at the same level. Additionally, we demonstrate effective results in the hospitality industry and show the potential for applying the PPE model trained on the hotel domain to other domains, including products and restaurants. This indicates the model's potential for broader applications beyond the hospitality sector. For pain point categorization, each identified pain point is classified into predefined categories corresponding to different hotel areas. This categorization allows hotel departments to focus on pain points relevant to their specific functions. We

doi:10.6342/NTU202402663

employ special tokens to integrate the entire review context into each pain point span representation, which enhances model performance.

5.2 Limitations and Future Work

However, our dataset is limited to high-class hotels. To test our model more broadly, we should include reviews from a wider range of hotel types. Different types of hotels, such as budget accommodations, may have pain points not covered in our current dataset. For instance, Sann et al. (2020) found that first-class hotel customers prioritize service and value, while lower-class hotel guests focus more on cleanliness, room quality, sleep quality, and location. Moreover, feasibility and reasonableness of pain points were not addressed in our study. There are situations where customers suggest impractical or unrealistic ideas, leading to pain points with low feasibility or reasonability. Feasibility is affected by the subjective nature of customer expressions, leading to situations where customers suggest impractical or impossible ideas. For example, a customer may request an early morning check-in, but this is often infeasible as previous customers may still be occupying the room and cleaning staff need time to prepare the room. For reasonability, customers might not always be aware of a product's positioning or may forget it when expressing pain points. Complaints about the lack of extra services in a low-end hotel may be unreasonable due to the lower cost. Conversely, it may be unreasonable for a customer to complain about the price of a luxury hotel, which is positioned and priced

doi:10.6342/NTU202402663

accordingly. Therefore, a model prioritizing feasibility and reasonability could be developed in the future. Besides, in the multi-task learning model, the granularity in the auxiliary tasks should align more closely with that of the main task (i.e. token level). We do not apply auxiliary tasks operating at the token level because we want to avoid additional data annotation solely for auxiliary tasks. In the future, additional pain point-related auxiliary tasks with finer granularity should be developed.

doi:10.6342/NTU202402663

References

Cai, M., Tan, Y., Ge, B., Dou, Y., Huang, G., & Du, Y. (2021). PURA: A product-and-user oriented approach for requirement analysis from online reviews. *IEEE Systems Journal*, 16(1), 566-577.

Chen, W., & Tabari, S. (2017). A study of negative customer online reviews and managerial responses on social media—case study of the Marriott Hotel Group in Beijing. *Journal of Marketing and Consumer Research*, 41, 53-64.

Holjevac, I. A., Marković, S., & Raspor, S. (2010, June). Customer satisfaction measurement in hotel industry: Content analysis study. In *Proceedings of the 4th International Scientific Conference on Planning for the Future Learning from the Past: Contemporary Developments in Tourism, Travel & Hospitality*.

Huang, Z., Xu, W., & Yu, K. (2015). Bidirectional LSTM-CRF models for sequence tagging. *arXiv preprint arXiv:1508.01991*.

Jiang, H. (2022). A multi-task deep neural network method for sentiment lexicon extraction. Unpublished Master Thesis. Department of Information Management, National Taiwan University, Taipei, Taiwan, ROC.

Jiang, S., Zhao, S., Hou, K., Liu, Y., & Zhang, L. (2019, October). A BERT-BiLSTM-CRF model for Chinese electronic medical records named entity recognition. In

doi:10.6342/NTU202402663

Proceedings of 12th International Conference on Intelligent Computation Technology and Automation (ICICTA) (pp. 166-169). IEEE.

Khan, I., & Fatma, M. (2022). Using netnography to understand customer experience towards hotel brands. *Sustainability*, 15(1), 279.

Lee, C. C., & Hu, C. (2005). Analyzing hotel customers' e-complaints from an internet complaint forum. *Journal of Travel & Tourism Marketing*, 17(2-3), 167-181.

Lee, Y., Kim, J., Kim, D., Kho, Y., Kim, Y., & Kang, P. (2023). Painsight: An extendable opinion mining framework for detecting pain points based on online customer reviews. *arXiv preprint arXiv:2306.02043*.

Li, H., Ye, Q., & Law, R. (2013). Determinants of customer satisfaction in the hotel industry: An application of online review analysis. *Asia Pacific Journal of Tourism Research*, 18(7), 784-802.

Liu, X., Burns, A. C., & Hou, Y. (2017). An investigation of brand-related user-generated content on Twitter. *Journal of Advertising*, 46(2), 236-247.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V. (2019). Roberta: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*.

doi:10.6342/NTU202402663

Ma, L., & Sun, B. (2020). Machine learning and AI in marketing–Connecting computing power to human insights. *International Journal of Research in Marketing*, 37(3), 481-504.

McHugh, M. L. (2012). Interrater reliability: The kappa statistic. *Biochemia Medica*, 22(3), 276-282.

Miloslavić, M. (2019). "The customer is not always right": Frontline employees' perspective and coping with illegitimate customer complaints. Doctoral Dissertation. Rochester Institute of Technology, Croatia.

Mutlubaş, I. (2023). Evaluation of online customer complaints for hotel businesses in terms of expectation management and behavioral intention. *Journal of Tourism & Gastronomy Studies*, 11(2), 1416-1432.

Parasuraman, A., Berry, L. L., & Zeithaml, V. A. (1991). Understanding customer expectations of service. *MIT Sloan Management Review*, 32(3), 39-48.

Pacheco, C., García, I., & Reyes, M. (2018). Requirements elicitation techniques: a systematic literature review based on the maturity of the techniques. *IET Software*, 12(4), 365-378.

Piramanayagam, S., & Kumar, S. (2020). Determinants of customer's dissatisfaction: A content analysis of negative online customer reviews on budget segment hotels in India. *African Journal of Hospitality, Tourism and Leisure*, 9(1), 1-9.

doi:10.6342/NTU202402663

Salminen, J., Mustak, M., Corporan, J., Jung, S. G., & Jansen, B. J. (2022). Detecting pain points from user-generated social media posts using machine learning. *Journal of Interactive Marketing*, 57(3), 517-539.

Sann, R., Lai, P. C., & Liaw, S. Y. (2020). Online complaining behavior: Does cultural background and hotel class matter?. *Journal of Hospitality and Tourism Management*, 43, 80-90.

Singh, A., Saha, S., Hasanuzzaman, M., & Dey, K. (2022). Multitask learning for complaint identification and sentiment analysis. *Cognitive Computation*, 14(1), 212-227.

Soares, L. B., FitzGerald, N., Ling, J., & Kwiatkowski, T. (2019). Matching the blanks: Distributional similarity for relation learning. *arXiv preprint arXiv:1906.03158*.

Souza, F., Nogueira, R., & Lotufo, R. (2019). Portuguese named entity recognition using BERT-CRF. *arXiv preprint arXiv:1909.10649*.

Su, J. (2020, February 7). 你的 CRF 层的学习率可能不够大. [Blog post]. Retrieved from https://kexue.fm/archives/7196

Sun, C., Qiu, X., Xu, Y., & Huang, X. (2019). How to fine-tune BERT for text classification?. In *Proceedings of the 18th China National Conference on Chinese Computational Linguistics*, Kunming, China (pp. 194-206). Springer International Publishing.

doi:10.6342/NTU202402663

Sutton, C., & McCallum, A. (2007). An introduction to conditional random fields for relational learning. *Introduction to Statistical Relational Learning*. MIT Press: Cambridge, MA.

Tao, W., Zhang, Q., Zhang, M., & Li, Y. (2019, May). Mining pain points from hotel online comments based on sentiment analysis. In *Proceedings of 2019 IEEE Joint International Information Technology and Artificial Intelligence Conference (ITAIC)* (pp. 1672-1677). IEEE.

Wang, B., Miao, Y., Zhao, H., Jin, J., & Chen, Y. (2016). A biclustering-based method for market segmentation using customer pain points. *Engineering Applications of Artificial Intelligence*, 47, 101-109.

Yin, C., Jiang, C., Jain, H. K., Liu, Y., & Chen, B. (2023). Capturing product/service improvement ideas from social media based on lead user theory. *Journal of Product Innovation Management*, 40(5), 630-656.

Zhang, M., Fan, B., Zhang, N., Wang, W., & Fan, W. (2021). Mining product innovation ideas from online reviews. *Information Processing & Management*, 58(1), 102389.

doi:10.6342/NTU202402663