RESEARCH-ARTICLE

From Menus to the Interactive Food-Ordering Systems

MIN-JI KIM, The Catholic University of Korea, Bucheon, Gyeonggi-do, South Korea

SEONG-JIN PARK, The Catholic University of Korea, Bucheon, Gyeonggi-do, South Korea

JAEHWAN HA

JU-WON SEO, The Catholic University of Korea, Bucheon, Gyeonggi-do, South Korea

DINARA ASYLKHANOVNA ALIYEVA, The University of North Carolina at Chapel Hill, Chapel Hill, NC, United States

KANG-MIN KIM, The Catholic University of Korea, Bucheon, Gyeonggi-do, South Korea

Open Access Support provided by:

The Catholic University of Korea

The University of North Carolina at Chapel Hill

PDF Download
3746252.3761001.pdf
17 February 2026
Total Citations: 0
Total Downloads: 143

Published: 10 November 2025

Citation in BibTeX format

CIKM '25: The 34th ACM International Conference on Information and Knowledge Management
November 10 - 14, 2025
Seoul, Republic of Korea

Conference Sponsors:
SIGWEB
SIGIR

CIKM '25: Proceedings of the 34th ACM International Conference on Information and Knowledge Management (November 2025)
https://doi.org/10.1145/3746252.3761001
ISBN: 9798400720406

From Menus to the Interactive Food-Ordering Systems

Min-Ji Kim[*] [†]
Department of Artificial Intelligence
The Catholic University of Korea
Bucheon, Republic of Korea
kimmin122@catholic.ac.kr

Seong-Jin Park[*]
Department of Artificial Intelligence
The Catholic University of Korea
Bucheon, Republic of Korea
sjpark@catholic.ac.kr

Jaehwan Ha
Beaverworks Inc.
Seoul, Republic of Korea
jaeh1115@beaverworksinc.com

Ju-Won Seo
Department of Artificial Intelligence
The Catholic University of Korea
Bucheon, Republic of Korea
tjwndnjs310@catholic.ac.kr

Dinara Aliyeva
Department of Computer Science
The University of North Carolina
at Chapel Hill
Chapel Hill, NC, USA
adinara@cs.unc.edu

Kang-Min Kim[‡]
Department of Artificial Intelligence
Department of Data Science
The Catholic University of Korea
Bucheon, Republic of Korea
kangmin89@catholic.ac.kr

Abstract

Conversational interfaces have emerged as an accessible and user-friendly alternative to traditional touch-based self-service kiosks in food-ordering systems. Despite their promise, building such systems remains challenging due to the need for costly data annotation, store-specific model adaptation, and scalable deployment. In this study, we propose a fully automated, end-to-end framework that transforms structured menu databases into high-quality annotated datasets and efficiently deploys store-specific conversational models using a parameter-efficient fine-tuning method. Our approach fine-tunes only 0.9% of the backbone model parameters per store, enabling cost-effective and plug-and-play deployment across diverse environments. To enhance robustness, we further integrate a recommendation module that suggests alternative items when requested menu options are unavailable. Experimental results on data from 27 stores in South Korea demonstrate that our framework consistently outperforms existing data generation baselines in intent classification and slot filling performance, while maintaining high annotation quality. Simulated real-world voice-ordering scenarios confirm the practicality of our framework for rapid, scalable, and accessible deployment in real-world environments.

CCS Concepts

• Human-centered computing → Systems and tools for interaction design; **Accessibility systems and tools**; **• Computing methodologies → Natural language processing**.

Keywords

Natural Language Understanding; Pre-trained Language Model; Automatic Framework; Conversational Interface; Food Ordering System; Accessibility Systems

ACM Reference Format:
Min-Ji Kim, Seong-Jin Park, Jaehwan Ha, Ju-Won Seo, Dinara Aliyeva, and Kang-Min Kim. 2025. From Menus to the Interactive Food-Ordering Systems. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management (CIKM '25), November 10–14, 2025, Seoul, Republic of Korea.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3746252.3761001

1 Introduction

The food service industry has undergone a significant transformation with the widespread adoption of contactless ordering systems, particularly in the form of self-service kiosks [1, 52]. This shift has been driven by rising labor costs [39, 47] and was further accelerated by the COVID-19 pandemic, which increased public preference for minimal-contact transactions [25, 58]. Most existing kiosks rely on touch-based interfaces [18], requiring users to navigate through visual menus and tap screens to place orders. While functional, this interaction interface presents accessibility barriers for certain user groups, such as the elderly or individuals with physical disabilities.

To address these limitations, voice-based conversational interfaces have emerged as a more accessible and user-friendly alternative [15, 43]. By simulating interactions with a human, these systems allow users to place orders via natural spoken language, reducing friction for those unfamiliar with digital interfaces [35, 37]. In South Korea, government policies mandating digital accessibility have further propelled the adoption of voice-ordering kiosks, highlighting the need for universally accessible systems [26, 38]. To support wide-scale deployment, voice-ordering kiosks should be cost-efficient and easy to maintain.

The core technology behind these systems is natural language understanding (NLU), which processes user utterances transcribed by a speech-to-text (STT) module. Since the emergence of pre-trained language models (PLMs), significant advances in NLU tasks have enabled the development of robust task-oriented dialogue systems [12, 28, 32, 60]. These systems typically consist of two main components: intent classification (IC) to determine the user's goal,

[*] These authors contributed equally to this work.
[†] Work done while author was an intern at Beaverworks Inc.
[‡] Corresponding author.

CIKM '25, Seoul, Republic of Korea.

ACM ISBN 979-8-4007-2040-6/2025/11
https://doi.org/10.1145/3746252.3761001

and slot filling (SF) to extract relevant arguments [7, 8, 44, 56]. While recent studies have explored replacing NLU pipelines with large language models (LLMs) using natural language generation (NLG) [6, 29, 55, 64], NLG-based systems still face practical limitations such as hallucinations, high latency, and high inference costs [23, 57]. As a result, NLU-centric architectures remain more practical for real-world deployment in small-scale kiosk providers that operate under computational constraints.

Developing voice-ordering systems generally requires two stages: generating annotated datasets for IC and SF, and training and deploying models tailored to each store. However, for kiosk providers, particularly when addressing small or franchise stores[1] with frequent menu changes, manually building and updating such datasets to reflect menu changes is prohibitively time-consuming and costly. Although prior studies have proposed generating synthetic text from structured data [33, 34, 48], most focus on plain text and do not yield labeled data suitable for IC and SF. Moreover, training a single model on aggregated multi-store data often leads to degraded performance due to store-specific variation, while maintaining separate models for each store is computationally inefficient.

In this study, we present a novel end-to-end framework that enables the rapid development and deployment of voice-based conversational interfaces in food-ordering kiosks. Given only a structured menu database, our framework automatically generates annotated datasets for IC and SF, fine-tunes store-specific models using parameter-efficient adapters [45], and deploys them within a shared backbone architecture. We leverage predefined templates to ensure full slot coverage with natural and fluent utterances, and apply character-level perturbations and augmentation techniques to simulate realistic user variation. Each store-specific adapter is fine-tuned via multitask learning (i.e., IC and SF) and integrated into a unified model for seamless deployment. To further enhance user experience, we incorporate a recommendation module that suggests alternatives when unavailable menu items are requested.

We evaluate our framework using data from 27 stores in South Korea. Our system successfully generated high-quality datasets, maintained complete slot coverage, and outperformed various baselines in IC and SF performance. Store-specific adapters enabled plug-and-play deployment with only a 0.9% increase in backbone model's parameters, significantly reducing maintenance overhead. Furthermore, the model exhibited consistent performance across both synthetic and manually annotated datasets, demonstrating its effectiveness for rapid deployment. In real-world simulation experiments with STT module, we further validated the practicality and robustness of our approach in deployment environments. In addition, the item recommendation module achieved 88% Hit@1 and 96% Hit@5, demonstrating its effectiveness in supporting reliable and adaptive user interaction.

The contributions of our study are as follows:

- We propose a fully automated, end-to-end framework for building voice-ordering conversational interfaces that requires only a structured menu database as input.
- We enable efficient deployment across diverse store environments by fine-tuning only 0.9% of model parameters per store through adapter-based multitask learning.
- We demonstrate the effectiveness of our framework through extensive experiments, achieving high IC and SF performance while maintaining data quality and deployment scalability.
- We deliver a practical, robust, and fully integrated pipeline optimized for real-world deployment.

2 Related Works

2.1 Conversational Interfaces for Task-Oriented Systems

Early task-oriented conversational systems were rule-based [5, 14], but advancements in natural language processing (NLP) have gradually shifted the field toward data-driven and learning-based approaches [10, 16]. With the advent of deep learning, models utilizing NLU and NLG techniques have been applied to various domains such as restaurant reservations and music search [30, 40]. As task-oriented systems have expanded into domain-specific applications, research has explored the integration of NLU with structured services. For example, Yan et al. [62] employed convolutional neural networks (CNNs) to build conversational agents for e-commerce, enabling functionalities such as product search and recommendation. Similarly, Raux et al. [50] developed a pizza-ordering agent using CNNs to process order-related intents and slot values.

However, most existing systems are built for fixed domains or individual services, making them difficult to scale. Each new deployment typically requires manual reconstruction of training data and model configuration, limiting their adaptability to similar stores or services. In contrast, our framework is designed for broad applicability by automatically generating store-specific conversational interfaces from structured menu data, enabling scalable deployment without the need for task redefinition or manual annotation.

2.2 End-to-End Framework for Interface Development

Traditional pipeline-based dialog systems decompose functionality into discrete modules such as NLU, dialog state tracking, dialog policy, and NLG. While this modularity offers interpretability and control, it often leads to increased development complexity and error propagation between components [61]. To address these limitations, recent studies have proposed end-to-end architectures that unify these components within a single model. SimpleTOD [21] treats dialog modeling as a sequence generation problem using a causal language model. SPACE-3 [19] pretrains a multi-headed transformer model that simultaneously handles understanding, policy, and generation, while GALAXY [20] incorporates semi-supervised learning and dialog-act supervision to enhance robustness. For deployment scalability, parameter-efficient fine-tuning methods have gained attention. Task-specific adapters [3] and low-rank adaptation modules [54] are integrated into frozen backbone models, significantly reducing memory and compute overhead.

[1] In this paper, the term 'store' encompasses restaurants, cafes, and similar establishments.

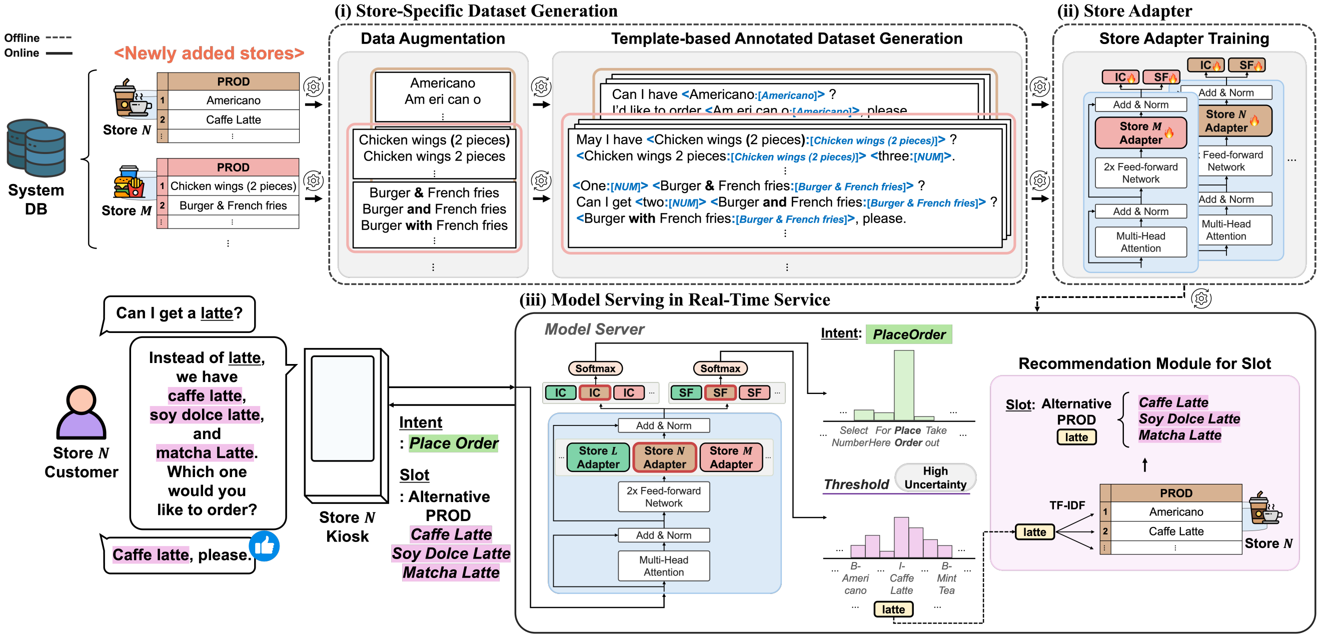

Figure 1: Overview of our proposed framework.

These approaches focus on general-purpose dialog systems or zero-shot transfer between domains. Our work differs in that it applies these principles to a practical domain (i.e., voice-ordering kiosks) by integrating modular fine-tuning into an end-to-end pipeline optimized for real-world deployment, data efficiency, and operational scalability.

2.3 Dataset Generation and Augmentation Method

Labeled data for IC and SF is essential but expensive to produce, especially in domain-specific or multilingual settings. To alleviate this bottleneck, recent work has explored automatic dataset generation and augmentation techniques. Template-based methods such as SynthDST [27] combine abstract dialog schemas with LLM-powered paraphrasing to create synthetic dialogs with labeled annotations. Paraphrasing-based approaches use LLMs to expand few-shot datasets by generating utterance variants [36, 63]. Other approaches enhance robustness through noise injection; DemoNSF [13] introduces controlled perturbations (e.g., typos, synonyms, structural edits) to train slot filling models via denoising objectives. Classical techniques, such as back-translation and slot value substitution, remain relevant for improving model generalization in low-resource settings [4].

While prior work often focuses on generating general-purpose or domain-transferable data, our method is specifically designed to generate annotated training data from structured menu databases. By leveraging predefined templates and character-level perturbations, we ensure complete slot coverage, maintain natural language fluency, and eliminate the need for manual labeling, which enables rapid and scalable deployment of store-specific models.

3 Methodology

In this section, we present the three main components of our proposed framework for building conversational interfaces in voice-ordering kiosks. Section 3.1 describes our method for efficiently generating high-quality, store-specific training datasets. Section 3.2 introduces our approach to model construction, where store-specific adapters are fine-tuned on a shared backbone model for efficient and scalable deployment. Finally, Section 3.3 describes our deployment strategy for conversational interfaces in real time and details the item recommendation module designed to improve system robustness and user experience.

3.1 Store-Specific Dataset Generation

To automatically construct a dataset for IC and SF from store-specific structured data, we propose a template-based generation approach with predefined intent labels and sequence-tagging slot labels. In this study, we define a total of 13 user utterance intents, including `Place Order`, `Cancel Menu`, and `Fallback`, and define slots for each intent. During the process of converting a structured menu database into ordering texts, we refine attribute expressions within the menu to mitigate errors caused by customer speech mistakes or inaccuracies in the STT module. Subsequently, we generate data by filling the attributes of the menu database into predefined intent-specific templates. For clarity, the top-left part of Figure 1 illustrates an example menu database from a store, and panel (i) visualizes the data generation process.

3.1.1 ***Enhancing Data Representation.*** We derive several insights from structured data and real-world ordering scenarios. Most structured product data in the menu database are written in natural language but often include special characters or notations with

Table 1: Overview of intents and their corresponding slots, templates, and phrases.

Intent	Description	# of Slots	Slots	# of Templates	# of Phrases
Fallback	Erroneous utterance unusable for ordering	0	-	0	403
Place Order	Utterance expressing intent for order placement	3	Menu name, Option name, Quantity	35	0
Cancel Menu	Utterance indicating a request to cancel an order	2	Menu name, Quantity	94	0
Select Number	Utterance selecting one of the displayed options	1	Number	0	500
Select Quantity	Utterance specifying the quantity of a product	1	Quantity	0	348
Phone Number	Utterance requesting point accumulation via phone number	1	Phone number	0	120
Point	Utterance requesting point redemption	1	Point amount	0	120
Pos. Response	Utterance affirming the kiosk's question	0	-	0	71
Neg. Response	Utterance negating the kiosk's question	0	-	0	72
Full Pay	Utterance requesting a one-time payment	1	Payment method	0	37
Installment Pay	Utterance requesting an installment payment	1	Installment period	0	69
For Here	Utterance requesting to dine in	0	-	0	58
Takeout	Utterance requesting takeout	0	-	0	54

implicit meanings (e.g., '&', '( )'). In addition, hesitation during ordering can result in fragmented speech, causing the STT module to transcribe incomplete product names. To address these issues, we preprocess special characters and notations.

Certain special characters are pronounced according to their meaning, rather than read phonetically. For instance, '&' is generally spoken as 'and'. Based on natural speech patterns, we heuristically replace special characters with manually crafted verbal equivalents. Specifically, 'Burger & French fries' was rendered as 'Burger and French fries'.

We further introduce character-level perturbations by randomly inserting spaces into words, simulating disfluent speech in STT. Formally, for a word $w = (c_1, c_2, \ldots, c_n)$ consisting of n characters, we apply a perturbation function $\mathcal{F}$ that inserts a space with probability k between each character:

$$\mathcal{F}(w) = \bigoplus_{i=1}^{n-1} \begin{cases} c_i \,\|\, \text{` '} & \text{with probability } k \\ c_i & \text{otherwise} \end{cases} \| \, c_n. \qquad (1)$$

This simulates variations such as 'Bur ger an d Fre nch fri es', enhancing robustness in sub-utterance modeling. This process is illustrated in the left part of Figure 1 (i).

3.1.2 ***Template-Based Annotated Dataset Generation.*** We analyze customer utterances collected from real stores, along with data generated by human annotators, to identify and construct reusable templates for intent-slot annotation. For intents involving store-specific slots, such as `Place Order` and `Cancel Menu`, we define 35 and 94 templates, respectively. These intents require customized data generation due to differences in menu items, options, and quantities across stores. Accordingly, the templates are populated with store-specific attributes to produce tailored training instances.

In contrast, we use predefined fixed sentences (i.e., phrases) for store-independent intents, such as `Select Quantity`, to ensure consistency across deployments. These sentences remain constant across stores and are augmented with slot information as needed. The choice between templates and fixed sentences is determined by the degree of store-specific variability and utterance complexity. Template-based generation is used when diversity in input structure is required, while fixed phrases suffice for general-purpose, low-variance utterances. A summary of all defined intents, associated slots, and the number of templates and fixed sentences used per intent is provided in Table 1.

Using slot information, our templates automatically assign labels during the data generation process. Following the sequence labeling scheme used in the KLUE benchmark [41], we represent slot annotations within templates using the format '<Data:`[Slot]`>'. For instance, a phrase annotated as '<Burger and French fries:`[Burger & French fries]`>' is incorporated into a sentence such as 'Can I get <two:`[NUM]`> <Burger and French fries:`[Burger & French fries]`>?' (described in the right part of Figure 1 (i)).

In STT-based conversational interfaces, short utterances often result in transcription errors, particularly in unit nouns. For instance, the phrase 'Number two' intended for the `Select Number` intent may be misrecognized as 'No more two', leading to incorrect classification under the `Fallback` intent. To address this issue, we augment training data by incorporating phonetically similar variants associated with the numeric slot `NUM`, thereby improving robustness at the sub-utterance level. While such augmented phrases may sound unnatural to human evaluators, they yield strong performance on manually constructed test sets that simulate real-world STT errors.

3.2 Store-Specific IC & SF Model Training

To support multiple stores within a single backbone model, we adopt an adapter tuning strategy that trains only a small number of additional parameters per store, instead of fine-tuning the entire model. Once trained, store-specific adapters are integrated into a unified model for efficient deployment.

3.2.1 ***Store-Specific Adapters.*** Each store-specific adapter is trained independently while keeping the backbone model frozen. To maximize scalability, we employ the Pfeiffer adapter (P-Adapter) [45] rather than the Houlsby adapter (H-Adapter) [22]. The P-Adapter inserts a single adapter module after the feed-forward network in each Transformer encoder layer [53], whereas the H-Adapter inserts adapters after both the multi-head attention and feed-forward layers. This streamlined structure reduces overhead while maintaining adaptability. An illustration of this architecture is shown in Figure 1 (ii).

3.2.2 ***Adapter Integration into a Unified Model.*** After training, all store-specific adapters are sparsely integrated into a single

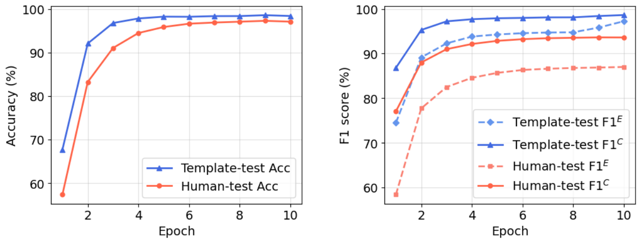

Figure 2: Comparison of performance trends between template-based and human-created test datasets on KLUE-RoBERTa$_{large}$.

backbone model. While the backbone parameters are shared across all stores, each adapter remains functionally independent, preserving store-specific behavior. This design enables plug-and-play extensibility: new store adapters can be added without retraining the entire model, and obsolete adapters can be removed without affecting overall system integrity.

3.3 Model Serving in Realistic Environments

To evaluate the deployability of our framework, we conduct end-to-end tests in a real-time inference environment using TorchServe[2], developed by PyTorch [42]. In this setting, the food-ordering system communicates with the model server via a REST API, sending payloads that include the user's transcribed utterance and system metadata. The server processes the input and returns results for both IC and SF. We also incorporate a recommendation module within the serving pipeline to provide alternative suggestions when users request unavailable items. The overall model serving workflow is illustrated in Figure 1 (iii).

3.3.1 ***Recommendation Module for Unavailable Items.*** When a user attempts to order an item that is not present in the store's menu, the model often exhibits high uncertainty, as such inputs were not observed during training. To detect low-confidence predictions, we apply the softmax function to the slot logits $\mathbf{z}$, where each element z_i represents the logit score for the i-th slot label:

$$\mathbf{p} = \text{softmax}(\mathbf{z}) \quad \text{where} \quad p_i = \frac{e^{z_i}}{\sum_j e^{z_j}}. \tag{2}$$

Here, p_i is the softmax probability assigned to the i-th slot label, and the index j iterates over all possible slot labels.

If the maximum confidence score $\max(\mathbf{p})$ falls below a predefined threshold τ_{conf}, we interpret the model's prediction as uncertain. We then compute the cosine similarity between the term frequency-inverse document frequency (TF-IDF) [51] vector of the user's utterance $\mathbf{u}$ and the TF-IDF vectors of all menu items $\mathbf{m}_j$:

$$\text{sim}(\mathbf{u}, \mathbf{m}_j) = \frac{\mathbf{u} \cdot \mathbf{m}_j}{\|\mathbf{u}\| \, \|\mathbf{m}_j\|}. \tag{3}$$

Here, the index j refers to the j-th menu item in the store's menu. If the highest similarity score $\max_j \text{sim}(\mathbf{u}, \mathbf{m}_j)$ exceeds a second threshold τ_{tfidf}, the corresponding menu item is retrieved and returned as a recommendation. Both τ_{conf} and τ_{tfidf} are empirically determined through validation experiments.

Table 2: Category, number of menu items, and average number of options per store used in this study.

Store ID	Category	# of Menus	Avg. # of Options per Menu
1	Café	30	5.06
2	Café	47	4.49
3	Italian	43	6.58
4	Burger	94	3.88
5	Snack	57	2.39
6	Japanese	61	23.08
7	Café	220	3.71
8	Italian	52	4.44
9	Café	36	9.67
10	Korean	53	28.3
11	Japanese	78	2.17
12	Korean	31	3.84
13	Korean	36	8.22
14	Café	44	5.66
15	Café	148	3.53
16	Asian	13	2.31
17	Sandwich	52	1.12
18	Japanese	11	1.82
19	Café	67	2.34
20	Café	36	3.28
21	Korean	28	4.57
22	Korean	34	3.0
23	Italian	29	1.38
24	Korean	93	2.88
25	Café	46	4.13
26	Salad	38	2.66
27	Burger	47	5.96

3.3.2 ***End-to-End Adapter Update for New Stores.*** When a new store is introduced or an existing menu is updated, our framework automatically generates training and validation datasets from the structured menu database and fine-tunes a store-specific adapter accordingly. The newly trained adapter is then integrated into the unified model and made available to the system with minimal overhead. While template-based data generation ensures scalability, it may raise concerns about generalizability in real-world scenarios. To assess this, we compared model performance on the `Place Order` intent using both human-created and template-generated test sets. The results showed consistent performance trends across both datasets (described in Figure 2), suggesting that our template-based method provides sufficient diversity and captures real-world user inputs.

4 Experiments

4.1 Experimental Setup

4.1.1 ***Store Information.*** We conduct our experiments using menu databases from 27 stores in South Korea. Each store belongs to one of nine categories and is associated with a distinct brand, resulting in unique distributions of menu items and configurable options across stores. Table 2 summarizes the category, number

[2] https://pytorch.org/serve/

of menu items, and average number of options per item for each store. On average, the stores contain 56.44 menu items, while the number of options per item varies widely across stores, ranging from 1.12 to 23.08. We generate the training data using the actual menu structures and option configurations for each store.

4.1.2 ***Test Datasets.*** Due to its complexity and central importance, we manually create 100 test samples per store for the `Place Order` intent, covering all products across the 27 stores. For the remaining intents, we construct test datasets using predefined templates, ensuring no overlap with the training data used in either the baseline methods or our proposed approach. In total, we construct 6,989 test instances. Twelve human annotators with domain expertise in the food service industry create the ordering-intent test data, ensuring realistic and high-quality samples.

4.1.3 ***Data Generation Baselines.*** We compare our proposed method against two baseline approaches for annotated dataset generation, categorized as follows. First, we include a vanilla template-based method (TUDA) [48], which directly populates predefined templates without leveraging language model-based inference. Second, we experiment with an open-source Korean LLM based on Llama 3_{8B} [17], known as Bllossom [9]. Motivated by recent studies that demonstrate the sequence labeling capabilities of large language models through in-context learning [2, 57], we adopt a 3-shot prompting strategy with Bllossom to generate slot-annotated utterances.

4.1.4 ***Models.*** To evaluate our framework, we conduct experiments using three types of pre-trained models from the KLUE benchmark, along with XLM-RoBERTa$_{large}$ [11]. For both IC and SF, we test four model adaptation strategies: full fine-tuning (Full), H-Adapter, P-Adapter, and classifier-only fine-tuning.

4.1.5 ***Evaluation Metrics.*** We evaluate the quality of the generated datasets through both human and automatic evaluations, following established methodologies in prior work [33, 34]. For human evaluation, we use a 3-point Likert scale [31] to assess four dimensions: plausibility, fluency, concept coverage, and overall quality. Three annotators with domain expertise in the food service industry independently assess the samples. For automatic evaluation, we evaluate two key aspects: effectiveness and efficiency. To assess effectiveness, we count the number of dropped slots, following the approach of Luo et al. [34]. To measure efficiency, we record the processing time in milliseconds. We assess model performance on IC and SF using the evaluation metrics defined in the KLUE benchmark: accuracy (Acc) for IC, and both entity-level (F1^{E}) and character-level (F1^{C}) micro-F1 scores for SF. Finally, we evaluate the performance of the recommendation module using the Hit rate at top-k (Hit@k) metric [24].

4.1.6 ***Implementation Details.*** We implement the model training framework using the Hugging Face Transformers [59] and adapter transformers [46] libraries. We initialize the training process with three different random seeds and report the average performance. We employ Bllossom with temperature set to 0.8 and nucleus (top-p) sampling set to 0.85. For PLM training, we trained the models for 10 epochs with a batch size of 32 and a weight decay of 1e-2, sweeping learning rates from 1e-4, 3e-4, 5e-4. The adapter configuration is set with a reduction factor of 16 and ReLU as the activation function. All experiments are conducted on three NVIDIA A100 and four NVIDIA RTX 3090 GPUs.

Table 3: Number of training samples by dataset generation method.

Method	# of Training Data
TUDA	52,143
Bllossom	39,954
Ours (w/o augmentation)	52,143
Ours	281,497

4.1.7 ***Details for Real-World Performance Evaluation.*** We simulate real-world conditions using STT processing. For each of the 27 stores, we randomly sample 12 `Place Order` intent sentences from those originally created by human annotators, which results in a total of 324 test samples. Three different speakers read each sentence aloud, and we transcribe the recordings using the open-source STT model Whisper-large-v3[3] [49]. We then feed the transcribed utterances into the trained KLUE-RoBERTa$_{large}$ model to perform IC and SF.

To avoid the overhead of re-annotating STT-transcribed texts with new slot labels, we evaluate slot-level accuracy by comparing the predicted slot values from the transcribed text with those from the original sentence. For example, if a sentence contains three menu items and the model correctly identifies two of them after STT, we calculate slot accuracy as 66.67%. We adopt this evaluation strategy because STT errors (e.g., spacing inconsistencies and recognition mistakes) pose challenges to the direct reuse of original slot annotations. Finally, we aggregate the performance results across the three speakers and 12 utterances per store, and report the average scores across all 27 stores.

4.2 Data Generation Details

Table 3 presents the size of the training datasets generated by each methodology. Although all methods use the same underlying menu data, the number of generated samples differs due to variations in augmentation and generation strategies. Below, we detail the generation process for each methodology.

4.2.1 ***TUDA.*** The original TUDA framework proposed a template-based data generation approach designed primarily for NLG tasks. To adapt this method for our NLU-oriented setting, we apply TUDA's template-based generation while modifying it to include slot annotations. Specifically, we automatically assign labels (e.g., menu and option names) as slot values within the generated utterances. This modification enables TUDA to produce datasets suitable for IC and SF tasks.

4.2.2 ***Ours.*** Building on the modified TUDA approach, our method introduces representation augmentation techniques to improve robustness against STT errors. We normalize special characters (e.g.,

[3] https://huggingface.co/openai/whisper-large-v3

Table 4: The results of automatic and human evaluations for the generated datasets. DS refers to dropped slots and P, F, C, O refer to plausibility, fluency, concepts, and overall, respectively. The best results are highlighted in bold and the second best results are underlined.

Generation Method	Automatic Evaluation		Human Evaluation			
	DS	Processing Time (ms)	P	F	C	O
Human	1.1	32,640 (↓ 0.88M×)	**2.97**	**2.91**	**2.93**	**2.96**
TUDA	**0.0**	**0.007** (↑ 0.19×)	2.79	2.41	2.85	2.48
Bllossom	3.8	4,630 (↓ 0.12M×)	2.85	2.62	2.86	2.69
Ours	**0.0**	0.037 (-)	2.77	2.4	2.75	2.59

replacing '&' with 'and') and randomly insert spaces between characters with a probability of 10%. This augmentation process generates multiple surface variants of each utterance while preserving the underlying semantic content. As a result, although we use the same templates and menu databases as TUDA, our method generates a significantly larger and syntactically more diverse dataset. Without representation augmentation, the dataset size would be identical to that of TUDA.

To ensure coverage, we apply at least one template to each menu item and its corresponding options. In some cases, multiple templates are applied to a single menu item. We randomly select templates using Python's built-in `random` package to introduce further variation.

4.2.3 ***Bllossom.*** We provide three few-shot examples per intent within the prompt and instruct the model to generate both utterances and corresponding slot annotations based on the menu database. To enhance diversity, we apply sampling techniques during generation and iteratively refine prompts with input from NLP experts. However, we heuristically observe that generating a large number of samples from a single menu often reduces diversity and degrades performance.

We attribute this limitation to the relatively small parameter size (8B) of the model, which may constrain its generative capabilities. Given the practical goal of supporting rapid deployment using only a new store's menu, we consider scaling to larger LLMs impractical. Therefore, we limit the number of generated samples per menu to a predefined threshold.

4.3 Experimental Results

4.3.1 ***Annotated Dataset Generation.*** Table 4 presents the evaluation results of our dataset generation method in comparison with the baseline approaches. Our method generates annotated datasets efficiently and achieves quality that is comparable to both human-written and LLM-generated data. As shown in Table 5, our method yields stronger performance in the slot filling task. Specifically, it outperforms the TUDA baseline by an average of 13.68% in F1^{E} and 3.78% in F1^{C}, while maintaining similar scores in human evaluation. Although human annotators report slightly lower fluency in the generated sentences, mainly due to the use of shuffled slot templates and representation augmentation techniques, yet the consistently strong results in IC and SF indicate that our method produces robust and reliable training data.

Table 5: The IC and SF results using different dataset generation methods, averaged across all 27 stores. All models are fine-tuned using P-Adapter. The best results for each model are highlighted in bold, and the second-best results are underlined.

Model *Parameter Size*	Generation Method	Intent	Slot	
		Acc	F1^{E}	F1^{C}
KLUE-BERT$_{base}$ *111M*	TUDA	96.86	79.83	91.07
	Bllossom	93.24	73.12	86.35
	Ours	**97.93**	**88.17**	**94.37**
KLUE-RoBERTa$_{base}$ *111M*	TUDA	93.76	75.28	87.79
	Bllossom	88.77	65.82	79.24
	Ours	**96.87**	**87.95**	**94.27**
KLUE-RoBERTa$_{large}$ *337M*	TUDA	**97.54**	82.02	91.61
	Bllossom	93.44	77.39	88.24
	Ours	97.52	**89.22**	**94.62**
XLM-RoBERTa$_{large}$ *560M*	TUDA	88.87	81.96	92.3
	Bllossom	87.63	74.21	87.78
	Ours	**96.73**	**97.27**	**93.05**

One notable observation is that our method generates data much more quickly than both human annotators and LLM-based methods such as Bllossom. In addition, it avoids slot omission during generation. Even when compared to TUDA, which produces data at a similar speed and without dropped slots, our method shows clearly better performance in both IC and SF tasks. These results support the overall effectiveness of our proposed template-based dataset generation approach and highlight its practicality for real-world deployment.

4.3.2 ***Intent Classification and Slot Filling.*** Table 6 presents the performance of various models on IC and SF tasks using several fine-tuning strategies. Among the evaluated models, XLM-RoBERTa$_{large}$ achieves the highest overall performance across different model sizes and methods. KLUE-RoBERTa$_{large}$ also performs competitively, offering a practical trade-off between accuracy and model size.

Our approach, which fine-tunes store-specific P-Adapters, shows better performance than full fine-tuning. When applied to KLUE-RoBERTa$_{large}$, this method achieves 97.52% accuracy in IC, 89.22% in F1^{E}, and 94.62% in F1^{C} in SF, while updating only 0.9% of the model parameters (described in Table 7). These results demonstrate that store-specific adapters offer a scalable and parameter-efficient solution for multi-store deployment. In line with findings from prior work [45], we observe that fine-tuning only the final classifier leads to substantially lower performance compared to methods that involve deeper parameter adaptation.

4.3.3 ***Effectiveness of Store-Specific Models.*** We validate the effectiveness of store-specific models for IC and SF. To this end, we compare two approaches: (1) training a separate model for each of the 27 stores using store-specific data, and (2) training a single unified model using the combined dataset from all stores. We generate the training datasets based on each store's menu using

Table 6: The IC and SF results using different training methods, averaged across all 27 stores. The best results for each model are highlighted in bold and the second best results are underlined.

Model	FT Method	Intent	Slot	
		Acc	F1^E	F1^C
KLUE-BERT$_{base}$	Full	96.24	**89.28**	**95.65**
	H-Adapter	97.64	88.07	94.32
	P-Adapter	**97.93**	88.17	94.37
	Classifier	91.38	74.88	88.93
KLUE-RoBERTa$_{base}$	Full	95.34	**88.6**	94.24
	H-Adapter	96.59	87.79	94.09
	P-Adapter	**96.87**	87.95	**94.27**
	Classifier	35.75	62.56	81.63
KLUE-RoBERTa$_{large}$	Full	92.09	84.95	92.16
	H-Adapter	97.23	88.92	94.52
	P-Adapter	**97.52**	**89.22**	**94.62**
	Classifier	42.04	65.44	83.23
XLM-RoBERTa$_{large}$	Full	86.29	82.44	91.53
	H-Adapter	**97.39**	93.18	**97.26**
	P-Adapter	96.73	**97.27**	93.05
	Classifier	36.73	61.41	72.99

Table 7: Comparison of the fine-tuning methods in terms of trainable parameters on KLUE-RoBERTa$_{large}$. The best results are highlighted in bold.

FT Method	Trainable Params per Store	Total Params for N Stores
Full	100%	100% × N
H-Adapter	1.8%	100% + (1.8% × N)
P-Adapter	**0.9%**	100% + **(0.9% × N)**

our proposed methodology, and evaluate the models using the same test datasets described earlier in this paper. All experiments use KLUE-RoBERTa$_{large}$ as the backbone model.

Table 8 presents the performance results for models trained using full fine-tuning and the P-Adapter method. Under both training strategies, store-specific models achieve better performance in slot filling compared to the single unified model. For IC, the unified model shows higher accuracy when trained with full fine-tuning, whereas store-specific models perform better when trained with the P-Adapter. Each store's dataset typically contains only a limited number of training samples, often ranging from a few dozen to a few hundred. In this context, the P-Adapter performs particularly well, as it is known to be effective in low-resource settings. Based on these observations, we design our framework to train P-Adapters that are specialized for each store, aiming to improve both efficiency and scalability.

4.3.4 ***Item Recommendation Module.*** We evaluate the recommendation module by generating 199 non-existent product names for each store. These names are partially matched variants of actual menu items, designed to simulate common user errors or unavailable requests. For each case, we manually define a set of appropriate alternative products. The module retrieves recommendations using TF-IDF-based similarity between the user input and available menu items. It achieves a Hit@1 of 88% and a Hit@5 of 96%, demonstrating strong retrieval performance. Table 9 and Table 10 show example outputs at each hit threshold.

Table 8: Comparison of fine-tuning a single model on all store datasets combined and on a single store dataset.

FT Method	Dataset	Intent	Slot	
		Acc	F1^E	F1^C
Full	All Stores	**92.43**	52.6	74.24
	Single Store	92.09	**84.95**	**92.16**
P-Adapter	All Stores	93.92	63.7	82.01
	Single Store	**97.52**	**89.22**	**94.62**

Table 9: Examples of item recommendation results at Hit@1. The original Korean input is shown in parentheses alongside its English translation. Correct predictions are highlighted in bold and color.

Partial Product Name	Alternative Products	Recommended Products
Ice Cream (아이스크림)	Strawberry sundae ice cream (딸기 선데이 아이스크림), Strawberry ice cream shake (딸기 아이스크림 쉐이크), Mango sundae ice cream (망고 선데이 아이스크림), Mango ice cream shake (망고 아이스크림 쉐이크), Ice cream grape smoothie (아이스크림 포도 스무디), **Ice cream latte (아이스크림 라떼)**	**Ice cream latte (아이스크림 라떼)**
Rosé Meal (로제 세트)	Black Tobiko Shrimp Rosé Meal (블랙 날치알 쉬림프 로제 세트), **Steaky Rosé Meal (스테이키 로제 세트)**	**Steaky Rosé Meal (스테이키 로제 세트)**

Instead of returning an error when a product is unavailable, the system suggests plausible alternatives that align with the user's intent. This improves the user experience, especially in voice-based ordering where recognition errors are more common. By providing relevant fallback options in real time, the module enhances the robustness and usability of the conversational interface. It helps reduce user frustration and supports higher order completion rates, contributing to the practicality of voice-ordering systems in real-world deployments.

4.3.5 ***Real-World Performance Evaluation.*** We evaluate the effectiveness of our framework in a simulated voice-ordering kiosk environment. For each of the 27 stores, we randomly sample 12 ordering instances from the test dataset. Three human participants read these sentences aloud, and we transcribe the audio using the Whisper model. For IC and SF, we use KLUE-RoBERTa$_{large}$, which offers a good balance between performance and model size. We assess accuracy by comparing the recognized intents and extracted slots from the transcribed utterances with those from the original

Table 10: Examples of item recommendation results at Hit@5. The original Korean input is shown in parentheses alongside its English translation. Correct predictions are highlighted in bold and color.

Partial Product Name	Alternative Products	Recommended Products
Ice Cream (아이스크림)	**Strawberry sundae ice cream (딸기 선데이 아이스크림),** **Strawberry ice cream shake (딸기 아이스크림 쉐이크),** **Mango sundae ice cream (망고 선데이 아이스크림),** **Mango ice cream shake (망고 아이스크림 쉐이크),** Ice cream grape smoothie (아이스크림 포도 스무디), **Ice cream latte (아이스크림 라떼)**	**Ice cream latte (아이스크림 라떼),** **Strawberry sundae ice cream (딸기 선데이 아이스크림),** **Mango sundae ice cream (망고 선데이 아이스크림),** **Strawberry ice cream shake (딸기 아이스크림 쉐이크),** **Mango ice cream shake (망고 아이스크림 쉐이크)**
Rosé Meal (로제 세트)	**Black Tobiko Shrimp Rosé Meal (블랙 날치알 쉬림프 로제 세트),** **Steaky Rosé Meal (스테이키 로제 세트)**	**Steaky Rosé Meal (스테이키 로제 세트),** Steaky Rosé (스테이키 로제), **Black Tobiko Shrimp Rosé Meal (블랙 날치알 쉬림프 로제 세트),** Steaky Rosé Gold Label (스테이키 로제 골드라벨), Carbonara Meal (까르보나라 세트)

text. Our framework achieves an IC accuracy of 96.11% and an SF accuracy of 84.06% in this real-world simulation, demonstrating its practical applicability for deployment in voice-based ordering systems.

5 Conclusion and Discussion

In this study, we present a fully automated end-to-end framework for developing conversational interfaces in food-ordering kiosks. Our framework covers the entire pipeline, from store-specific dataset generation to lightweight model deployment, requiring no manual intervention beyond the initial design of reusable templates. By significantly lowering the cost and complexity of data annotation and system integration, our approach addresses a key societal challenge: improving accessibility by enabling even small or independent stores to adopt conversational interfaces efficiently and affordably.

Through extensive experiments across 27 real-world stores, we demonstrate the effectiveness of our approach in generating high-quality annotated datasets, achieving robust performance in IC and SF. Notably, our adapter-based strategy fine-tunes only 0.9% of model parameters per store, ensuring scalability while preserving performance. Furthermore, our real-world evaluation using STT transcripts validates the system's practical applicability, demonstrating strong performance under realistic input conditions.

In addition, we integrate an item recommendation module to enhance system robustness. This module recovers from failed slot predictions when users mention unavailable items and provides alternative suggestions, significantly improving the user experience. With Hit@1 and Hit@5 scores of 88% and 96%, respectively, the module demonstrates practical reliability in conversational settings. Overall, our framework offers a cost-efficient, scalable, and highly accessible solution for deploying voice-based ordering systems in real-world environments.

5.1 Limitations and Future Work

Despite the strengths of our framework, several limitations remain, which we outline below along with potential future directions:

- **Data Fluency vs. Robustness**: Our template-based method with character-level augmentation improves robustness to STT errors but may reduce fluency in generated utterances. While human evaluations showed minor fluency degradation, future work may explore hybrid approaches that combine template generation with LLM-based paraphrasing to balance naturalness and resilience.
- **Model Scope**: We evaluate our framework on four PLM-based backbones, including the XLM-RoBERTa and KLUE family. Although these models are competitive in the Korean NLP landscape, extending the framework to support LLMs or multilingual settings would further demonstrate its generality.
- **Language Coverage**: Our current implementation is limited to Korean due to dataset availability. However, the proposed methodology is language-agnostic and can be extended to other languages using multilingual templates and models. Future work will validate this generalization in cross-lingual environments.
- **Template Authoring Effort**: While our approach avoids store-specific annotation, the initial creation of intent-specific templates still requires domain knowledge. Nonetheless, once defined, these templates are reusable across stores and only require minimal adjustments, supporting sustainable scaling. Semi-automated template induction based on existing corpora may further reduce this effort.

5.2 Broader Impacts

Our work contributes to the development of inclusive and accessible voice-ordering systems that reduce barriers for users who are unfamiliar with digital interfaces or have limited physical mobility. By democratizing the deployment of conversational AI to smaller food service providers, our framework supports digital equity and encourages the adoption of human-centered technologies in everyday commerce.

6 GenAI Usage Disclosure

In this study, we use the generative AI model Bllossom as one of the baseline tools for data generation. In addition, we utilize ChatGPT, a generative AI system, to assist with grammar checking during the writing process of this manuscript.

Acknowledgments

We thank the anonymous reviewers for their helpful comments. This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. RS-2025-25443681).

References

[1] Jee Ahe Ahn and Soobin Seo. 2018. Consumer responses to interactive restaurant self-service technology (IRSST): The role of gadget-loving propensity. *International Journal of Hospitality Management* 74 (2018), 109–121.

[2] Dhananjay Ashok and Zachary C Lipton. 2023. PromptNER: Prompting For Named Entity Recognition. *arXiv preprint arXiv:2305.15444* (2023).

[3] Namo Bang, Jeehyun Lee, and Myoung-Wan Koo. 2023. Task-Optimized Adapters for an End-to-End Task-Oriented Dialogue System. In *Findings of the Association for Computational Linguistics: ACL 2023*. 7355–7369.

[4] Samyadeep Basu, Amr Sharaf, Karine Ip Kiun Chong, Alex Fischer, Vishal Rohra, Michael Amoake, Hazem El-Hammamy, Ehi Nosakhare, Vijay Ramani, and Benjamin Han. 2022. Strategies to Improve Few-shot Learning for Intent Classification and Slot-Filling. In *Proceedings of the Workshop on Structured and Unstructured Knowledge Integration (SUKI)*. 17–25.

[5] Daniel G Bobrow, Ronald M Kaplan, Martin Kay, Donald A Norman, Henry Thompson, and Terry Winograd. 1977. GUS, A Frame-Driven Dialog System. *Artificial intelligence* 8, 2 (1977), 155–173.

[6] Paweł Budzianowski and Ivan Vulić. 2019. Hello, It's GPT-2 - How Can I Help You? Towards the Use of Pretrained Language Models for Task-Oriented Dialogue Systems. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*. 15–22.

[7] Giuseppe Castellucci, Valentina Bellomaria, Andrea Favalli, and Raniero Romagnoli. 2019. Multi-lingual Intent Detection and Slot Filling in a Joint BERT-based Model. *arXiv preprint arXiv:1907.02884* (2019).

[8] Qian Chen, Zhu Zhuo, and Wen Wang. 2019. BERT for Joint Intent Classification and Slot Filling. *arXiv preprint arXiv:1902.10909* (2019).

[9] ChangSu Choi, Yongbin Jeong, Seoyoon Park, Inho Won, HyeonSeok Lim, Sangmin Kim, Yejee Kang, Chanhyuk Yoon, Jaewan Park, Yiseul Lee, Hyejin Lee, Younggyun Hahm, Hansaem Kim, and Kyungtae Lim. 2024. Optimizing Language Augmentation for Multilingual Large Language Models: A Case Study on Korean. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation*. 12514–12526.

[10] Jennifer Chu-Carroll and Bob Carpenter. 1999. Vector-based natural language call routing. *Computational linguistics* 25, 3 (1999), 361–388.

[11] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised Cross-lingual Representation Learning at Scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 8440–8451.

[12] 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*. 4171–4186.

[13] Guanting Dong, Tingfeng Hui, Zhuoma GongQue, Jinxu Zhao, Daichi Guo, Gang Zhao, Keqing He, and Weiran Xu. 2023. DemoNSF: A Multi-task Demonstration-based Generative Framework for Noisy Slot Filling Task. In *Findings of the Association for Computational Linguistics: EMNLP 2023*. 10506–10518.

[14] Lee D Erman, Frederick Hayes-Roth, Victor R Lesser, and D Raj Reddy. 1980. The Hearsay-II speech-understanding system: Integrating knowledge to resolve uncertainty. *ACM Computing Surveys (CSUR)* 12, 2 (1980), 213–253.

[15] Nahyun Eun, Soobin Ou, Mijin Kim, Chaewon Yoo, and Jongwoo Lee. 2022. Speech-Recognizing KIOSK Mobile Application for the Visually Impaired. In *Proceedings of the 14th International Conference on Education Technology and Computers*. 575–580.

[16] Allen L Gorin, Giuseppe Riccardi, and Jeremy H Wright. 1997. How may I help you? *Speech communication* 23, 1-2 (1997), 113–127.

[17] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783* (2024).

[18] Lydia Hanks, Nathan D Line, and Anna S Mattila. 2016. The impact of self-service technology and the presence of others on cause-related marketing programs in restaurants. *Journal of Hospitality Marketing & Management* 25, 5 (2016), 547–562.

[19] Wanwei He, Yinpei Dai, Min Yang, Jian Sun, Fei Huang, Luo Si, and Yongbin Li. 2022. Unified dialog model pre-training for task-oriented dialog understanding and generation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 187–200.

[20] Wanwei He, Yinpei Dai, Yinhe Zheng, Yuchuan Wu, Zheng Cao, Dermot Liu, Peng Jiang, Min Yang, Fei Huang, Luo Si, et al. 2022. Galaxy: A generative pre-trained model for task-oriented dialog with semi-supervised learning and explicit policy injection. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 36. 10749–10757.

[21] Ehsan Hosseini-Asl, Bryan McCann, Chien-Sheng Wu, Semih Yavuz, and Richard Socher. 2020. A simple language model for task-oriented dialogue. *Advances in Neural Information Processing Systems* 33 (2020), 20179–20191.

[22] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. In *International conference on machine learning (ICML)*. 2790–2799.

[23] Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Yejin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of Hallucination in Natural Language Generation. *Comput. Surveys* 55, 12 (2023), 248:1–248:38.

[24] George Karypis. 2001. Evaluation of Item-Based Top-N Recommendation Algorithms. In *Proceedings of the Tenth International Conference on Information and Knowledge Management*. 247–254.

[25] Jinkyung Jenny Kim, Insin Kim, and Jinsoo Hwang. 2021. A change of perceived innovativeness for contactless food delivery services using drones after the outbreak of COVID-19. *International Journal of Hospitality Management* 93 (2021), 102758.

[26] Seongseop (Sam) Kim, Jungkeun Kim, Frank Badu-Baiden, Marilyn Giroux, and Youngjoon Choi. 2021. Preference for robot service or human service in hotels? Impacts of the COVID-19 pandemic. *International Journal of Hospitality Management* 93 (2021), 102795.

[27] Atharva Kulkarni, Bo-Hsiang Tseng, Joel Moniz, Dhivya Piraviperumal, Hong Yu, and Shruti Bhargava. 2024. SynthDST: Synthetic Data is All You Need for Few-Shot Dialog State Tracking. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*. 1988–2001.

[28] Pradnya Kulkarni, Ameya Mahabaleshwarkar, Mrunalini Kulkarni, Nachiket Sirsikar, and Kunal Gadgil. 2019. Conversational AI: An Overview of Methodologies, Applications & Future Scope. In *2019 5th International Conference On Computing, Communication, Control And Automation (ICCUBEA)*. 1–7.

[29] Dongyub Lee, Taesun Whang, Chanhee Lee, and Heuiseok Lim. 2023. Towards reliable and fluent large language models: Incorporating feedback learning loops in qa systems. *arXiv preprint arXiv:2309.06384* (2023).

[30] Xiujun Li, Yun-Nung Chen, Lihong Li, Jianfeng Gao, and Asli Celikyilmaz. 2017. End-to-End Task-Completion Neural Dialogue Systems. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing*. 733–743.

[31] Rensis Likert. 1932. A technique for the measurement of attitudes. *Archives of Psychology* (1932).

[32] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv preprint arXiv:1907.11692* (2019).

[33] Ximing Lu, Sean Welleck, Peter West, Liwei Jiang, Jungo Kasai, Daniel Khashabi, Ronan Le Bras, Lianhui Qin, Youngjae Yu, Rowan Zellers, Noah A. Smith, and Yejin Choi. 2022. NeuroLogic A*esque Decoding: Constrained Text Generation with Lookahead Heuristics. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 780–799.

[34] Bingfeng Luo, Zuo Bai, Kunfeng Lai, and Jianping Shen. 2020. Make Templates Smarter: A Template Based Data2Text System Powered by Text Stitch Model. In *Findings of the Association for Computational Linguistics*. 1057–1062.

[35] Am-Suk Oh. 2022. Interface design for service improvement of unmanned ordering device to the digital underprivileged. *Journal of the Korea Institute of Information and Communication Engineering* 26, 11 (2022), 1592–1598.

[36] Soham Parikh, Mitul Tiwari, Prashil Tumbade, and Quaizar Vohra. 2023. Exploring Zero and Few-shot Techniques for Intent Classification. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 5: Industry Track)*. 744–751.

[37] Sunyoung Park, Hyun K Kim, Yuryeon Lee, and Jaehyun Park. 2023. Kiosk accessibility challenges faced by people with disabilities: an analysis of domestic and international accessibility laws/guidelines and user focus group interviews. *Universal Access in the Information Society* (2023), 1–17.

[38] Soona Park, David J. Kwun, Jeong-Yeol Park, and Diego Bufquin. 2022. Service Quality Dimensions in Hotel Service Delivery Options: Comparison between Human Interaction Service and Self-Service Technology. *International Journal of Hospitality & Tourism Administration* 23, 5 (2022), 931–958.

[39] Soona Park, Xinran Lehto, and Mark Lehto. 2021. Self-service technology kiosk design for restaurants: An QFD application. *International Journal of Hospitality Management* 92 (2021), 102757.

[40] Sunghyun Park, Han Li, Ameen Patel, Sidharth Mudgal, Sungjin Lee, Young-Bum Kim, Spyros Matsoukas, and Ruhi Sarikaya. 2021. A Scalable Framework for Learning From Implicit User Feedback to Improve Natural Language Understanding in Large-Scale Conversational AI Systems. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. 6054–6063.

[41] Sungjoon Park, Jihyung Moon, Sungdong Kim, Won-Ik Cho, Jiyoon Han, Jangwon Park, Chisung Song, Junseong Kim, Youngsook Song, Tae Hwan Oh, Joohong Lee, Juhyun Oh, Sungwon Lyu, Younghoon Jeong, Inkwon Lee, Sangwoo Seo, Dongjun Lee, Hyunwoo Kim, Myeonghwa Lee, Seongbo Jang, Seungwon Do, Sunkyoung Kim, Kyungtae Lim, Jongwon Lee, Kyumin Park, Jamin Shin, Seonghyun Kim, Eunjeong Lucy Park, Alice Oh, Jung-Woo Ha, and Kyunghyun Cho. 2021. KLUE: Korean Language Understanding Evaluation. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1*.

[42] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Z. Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019*. 8024–8035.

[43] Jennifer Pearson, Gavin Bailey, Simon Robinson, Matt Jones, Tom Owen, Chi Zhang, Thomas Reitmaier, Cameron Steer, Anna Carter, Deepak Ranjan Sahoo, and Dani Kalarikalayil Raju. 2022. Can't Touch This: Rethinking Public Technology in a COVID-19 Era. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. 401:1–401:14.

[44] Baolin Peng, Chenguang Zhu, Chunyuan Li, Xiujun Li, Jinchao Li, Michael Zeng, and Jianfeng Gao. 2020. Few-shot Natural Language Generation for Task-Oriented Dialog. In *Findings of the Association for Computational Linguistics*. 172–182.

[45] Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. 2021. AdapterFusion: Non-Destructive Task Composition for Transfer Learning. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics*. 487–503.

[46] Jonas Pfeiffer, Andreas Rücklé, Clifton Poth, Aishwarya Kamath, Ivan Vulic, Sebastian Ruder, Kyunghyun Cho, and Iryna Gurevych. 2020. AdapterHub: A Framework for Adapting Transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. 46–54.

[47] Dinesh Puranam, Vrinda Kadiyali, and Vishal Narayan. 2021. The impact of increase in minimum wages on consumer perceptions of service: A transformer model of online restaurant reviews. *Marketing Science* 40, 5 (2021), 985–1004.

[48] Yevgeniy Puzikov and Iryna Gurevych. 2018. E2e nlg challenge: Neural models vs. templates. In *Proceedings of the 11th International Conference on Natural Language Generation*. 463–471.

[49] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2023. Robust speech recognition via large-scale weak supervision. In *International Conference on Machine Learning*. 28492–28518.

[50] Antoine Raux, Yi Ma, Paul Yang, and Felicia Wong. 2018. PizzaPal: Conversational Pizza Ordering using a High-Density Conversational AI Platform. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. 151–156.

[51] Gerard Salton and Christopher Buckley. 1988. Term-weighting approaches in automatic text retrieval. *Information processing & management* 24, 5 (1988), 513–523.

[52] Berthania Stanley, Yudha Pratama, and Agung Gita Subakti. 2023. The impact of self-order kiosk and service quality on customer experience in McDonald's Citra Garden 6 Jakarta. In *E3S Web of Conferences*, Vol. 426. EDP Sciences, 02073.

[53] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. *Advances in Neural Information Processing Systems* 30 (2017).

[54] Jian Wang, Chak Tou Leong, Jiashuo Wang, Dongding Lin, Wenjie Li, and Xiaoyong Wei. 2024. Instruct Once, Chat Consistently in Multiple Rounds: An Efficient Tuning Framework for Dialogue. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 3993–4010.

[55] Jiayin Wang, Weizhi Ma, Peijie Sun, Min Zhang, and Jian-Yun Nie. 2024. Understanding User Experience in Large Language Model Interactions. *arXiv preprint arXiv:2401.08329* (2024).

[56] Jixuan Wang, Kai Wei, Martin Radfar, Weiwei Zhang, and Clement Chung. 2021. Encoding Syntactic Knowledge in Transformer Encoder for Intent Detection and Slot Filling. In *Proceedings of the AAAI conference on artificial intelligence*. 13943–13951.

[57] Shuhe Wang, Xiaofei Sun, Xiaoya Li, Rongbin Ouyang, Fei Wu, Tianwei Zhang, Jiwei Li, and Guoyin Wang. 2023. GPT-NER: Named entity recognition via large language models. *arXiv preprint arXiv:2304.10428* (2023).

[58] Xueqin Wang, Yiik Diew Wong, Feng Liu, and Kum Fai Yuen. 2021. A push–pull–mooring view on technology-dependent shopping under social distancing: When technology needs meet health concerns. *Technological Forecasting and Social Change* 173 (2021), 121109.

[59] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. 38–45.

[60] Chien-Sheng Wu, Steven C. H. Hoi, Richard Socher, and Caiming Xiong. 2020. TOD-BERT: Pre-trained Natural Language Understanding for Task-Oriented Dialogue. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*. 917–929.

[61] Heng-Da Xu, Xian-Ling Mao, Puhai Yang, Fanshu Sun, and He-Yan Huang. 2024. Rethinking task-oriented dialogue systems: From complex modularity to zero-shot autonomous agent. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2748–2763.

[62] Zhao Yan, Nan Duan, Peng Chen, Ming Zhou, Jianshe Zhou, and Zhoujun Li. 2017. Building Task-Oriented Dialogue Systems for Online Shopping. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*. 4618–4626.

[63] Haode Zhang, Haowen Liang, Li-Ming Zhan, Xiao-Ming Wu, and Albert YS Lam. 2023. Revisit Few-shot Intent Classification with PLMs: Direct Fine-tuning vs. Continual Pre-training. In *Findings of the Association for Computational Linguistics: ACL 2023*. 11105–11121.

[64] Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. 2020. DIALOGPT : Large-Scale Generative Pre-training for Conversational Response Generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. 270–278.