

## 분별 프롬프트 학습을 통한 고객 질의응답 데이터의 계층 구조 표현 및 시각화

이 채 별<sup>1</sup> · 최 권 태<sup>2\*</sup>

<sup>1</sup>강남대 학교 ICT공학부 학사과정

<sup>2</sup>강남대 학교 ICT융합공학부 교수

# Hierarchical Representation and Visualization of Customer Query Response Data Through Discriminative Prompt Learning

Chae-Byeol Lee<sup>1</sup> · Kwon-Taeg Choi<sup>2\*</sup>

<sup>1</sup>Bachelor's Course, Department of ICT Engineering, Kangnam University, Yongin 16979, Korea

<sup>2</sup>Professor, Department of ICT Convergence Engineering, Kangnam University, Yongin 16979, Korea

### [요 약]

기업은 고객의 요구 사항을 파악하고 시장에서 경쟁 우위를 확보하기 위해 대규모 데이터를 수집하고, 딥러닝을 통해 분석 및 시각화하는 방법을 활용하고 있다. 그러나 기존의 딥러닝 기반 연구 방법은 새로운 데이터나 문제에 적응하기 어려우며, 복잡하고 다양한 고객 데이터를 충분히 반영하는 데 한계가 있다. 본 논문에서는 대형 언어 모델을 활용하여 고객 질문 데이터를 계층 구조로 표현하고, 분별 프롬프트 학습을 적용하여 보다 정확한 카테고리 분류 알고리즘을 제안한다. AI 허브 데이터를 사용해 제안 방법의 효과성을 검증하고, 고객 질의응답 데이터셋에 대해 97.75%의 카테고리 인식률을 보였다. 제안 방법은 데이터가 가지고 있는 복잡성과 구조를 직관적으로 이해할 수 있도록 하여, 기업이 고객의 요구를 효과적으로 파악하고 의사 결정 과정에 활용될 수 있다.

### [Abstract]

To understand customer needs and gain a competitive advantage in the marketplace, companies are collecting large volumes of data and using deep learning for analysis and visualization. However, existing deep learning methods often struggle to adapt to new data or problems and have limitations in fully capturing the complexity and diversity of customer data. In this paper, we propose a more accurate category classification algorithm that leverages a large language model to represent customer question data in a hierarchical structure, employing Discriminative Prompt Learning. We validate the effectiveness of this approach using AI Hub data, achieving a category recognition rate of 97.75% on a customer question-and-answer dataset. This method offers an intuitive way to understand the complexity and structure of data, enabling companies to better grasp customer needs and effectively integrate these insights into their decision-making processes.

**색인어** : 대규모 언어 모델, 질의 분석, 계층 구조, 시각화, 분별 프롬프트 학습

**Keyword** : LLM(Large Language Models), Query Analysis, Hierarchical Structure, Visualization, Discriminative Prompt Learning

<http://dx.doi.org/10.9728/dcs.2024.25.12.3837>



This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License(<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

**Received** 31 August 2024; **Revised** 16 October 2024

**Accepted** 28 October 2024

**\*Corresponding Author; Kwon-Taeg Choi**

**Tel:** +82-31-280-3660

**E-mail:** kwontaeg.choi@kangnam.ac.kr

## I. 서론

현대 경제에서 기업은 시장 요구를 파악하고 경쟁 우위를 확보하기 위해, 고객의 요구 사항과 관련 제품 및 서비스와 상호작용 방식에 대한 충분한 이해가 필요하다. 이를 위해 우선 기업은 고객 대상 설문 조사를 실시하거나, 웹 크롤링 방식 등을 통해 방대한 데이터를 수집한다. 개별적으로 수집된 데이터는 의사 결정에 필요한 인사이트 도출을 위해 하나의 데이터로 통합하여 고객 데이터의 규모와 복잡성을 관리하고, 보다 쉽게 해석할 여러 가지 방법이 연구되고 있다[1].

이러한 연구들은 주로 데이터를 계층 구조화하거나 시각화하는 두 가지 방향으로 진행되고 있다. 계층 구조화는 정보를 더 넓은 구조로 집계하여 상위 수준에서의 이해를 도모하는 데이터 추상화 기법이다. 고객의 요구 사항을 기존 요구 수준에 맞춰 계층적으로 분류하고, 이를 기준으로 제품의 기능을 정의하며 우선순위를 결정하는 방법이다[2]. 이러한 계층 구조를 활용해 고객 데이터 내에서 유사한 항목 간의 관계를 파악하고, 각 클러스터에서 가장 빈번하게 나타나는 항목을 시각화하여 고객 세분화를 수행하는 연구도 제시되었다[3].

대규모 데이터를 계층 구조화하기 위한 연구에서는 기계학습, 딥러닝 같은 방법을 주로 사용한다. 예를 들어, 데이터를 사전 정의된 기준에 따라 분류하거나[4], 특정 기계학습 알고리즘을 활용하여 클러스터링하는 방식 등이 있다[5]. 그러나 이러한 전통적인 방법은 사전에 정의된 규칙이나 특성을 기반으로 데이터를 분류하기 때문에 새로운 데이터나 문제에 대응하는 데 어려움이 있을 수 있다. 또한, 학습에는 많은 양의 데이터가 필요하며, 데이터의 도메인이 변경되면 재학습이 요구된다.

본 논문에서는 대규모 언어 모델(LLM; Large Language Models)을 활용하여 고객의 다양한 질문 데이터셋을 계층 구조로 표현하고, 시각화하는 방법론을 제안한다. 분류된 카테고리 하나의 통합된 구조로 만들어 전체 내용을 직관적으로 이해할 수 있도록 트리 구조를 활용하고자 한다. 트리 구조를 만들어가는 과정에서 카테고리 분류 성능을 개선하기 위해 분별 프롬프트 학습(DPL; Discriminative Prompt Learning)을 제안한다. 이를 통해 대규모 데이터의 규모와 복잡성을 줄이고 단순화를 통해, 경영진이 고객과 관련된 의사 결정을 내릴 때 활용될 수 있다.

## II. 관련 연구

### 2-1 LLM을 이용한 텍스트 데이터 분석

LLM은 최근 자연어 처리 분야에서 혁신을 이끌어냈으며, 2022년 말 OpenAI의 Chat GPT 출시 이후 생성형 인공지능에 대한 학계와 일반 대중의 관심이 급증했다. 대규모 텍스트

데이터를 기반으로 확률적 표현을 학습하며, 이후 미세조정 프로세스를 통해 번역, 요약, 질문 답변 등 다양한 작업을 수행할 수 있도록 조정될 수 있다.

LLM을 활용한 다양한 텍스트 분류 기법이 연구되었다. 논문[4]에서는 LLM의 고도화된 언어 이해 능력을 통해 방대한 문서 집합을 다양한 주제나 카테고리로 분류하는 방법을 제안했다. 대규모 코퍼스로부터 학습된 LLM은 새로운 작업에 대해서도 높은 정확도로 예측을 수행할 수 있다. 특히, 제로샷 학습을 통해 추가적인 훈련 없이도 새로운 작업을 처리할 수 있으며, 퓨샷 학습을 통해 몇 가지 예시를 제공함으로써 모델의 예측 성능을 더욱 향상시킬 수 있다. BERT 모델은 대규모 코퍼스에서 사전 학습된 후 특정 분류 작업을 위해 미세 조정되어 텍스트를 정확하게 분류할 수 있지만, GPT 모델은 적은 수의 예시만으로도 높은 정확도를 유지할 수 있어 비용과 시간을 절약할 수 있다.

LLM을 기반으로 한 클러스터링 알고리즘도 연구되었다. 논문[5]에서는 문맥을 고려한 깊이 있는 단어 관계를 통해 텍스트를 효과적으로 그룹화할 수 있음을 보여주었다. 최근 연구에서는 BERT 등의 LLM을 사용하여 다양한 텍스트 임베딩과 클러스터링 알고리즘을 비교한 결과, LLM 임베딩이 기존의 TF-IDF와 같은 전통적인 임베딩 기법보다 언어의 구조적 측면을 더 정교하게 포착할 수 있음을 실험적으로 보였다.

### 2-2 대규모 데이터 시각화

대규모 고객 데이터를 효과적으로 시각화하고 분석하는 방법은 매우 오랫동안 연구되었다[6]-[9]. 대규모 데이터의 키워드 추출 결과는 키워드의 종류와 빈도를 동시에 담고 있는 분석 결과이기 때문에 직관적인 시각화 방법이 요구된다.

워드 클라우드드는 고객 지원 시스템에서 수집된 키워드의 빈도를 기준으로 가중치를 부여하여, 빈도가 높은 키워드에는 폰트 크기가 크게 표현되도록 시각화하는 방법이다[6]. 이를 통해 고객이 자주 문의하는 키워드를 시각화하면, 고객의 주요 관심사나 문제점을 빠르게 파악할 수 있다. 논문[7]에서는 고객 데이터를 시각화하는 방법으로 기존의 단순한 막대그래프나 선형 차트 대신, 의견 휠(Openion Wheel)이라는 복잡한 시각화 기법을 제안했다. 의견 휠은 다양한 범주의 고객 데이터를 원형으로 배치하여 사용자가 데이터의 전반적인 분포와 특정 범주 내의 세부 사항을 동시에 파악할 수 있는 방법이다. 논문[8]에서는 고객 지원 요청 유형과 이를 처리하는 기업 지원 그룹 간의 관계를 나타내는 코드 다이어그램(Chord Diagram)을 제안했다. 코드 다이어그램은 항목 간의 관계와 흐름을 명확하게 시각화하는 방법이다. 한 해 이상의 고객 데이터를 단일 시각화로 표현하여, 각 고객 지원 요청이 어느 지원 그룹에 의해 처리되었는지를 파악할 수 있다. 논문[9]에서는 구조화되지 않은 텍스트 형식의 개방형 고객 응답 데이터를 시각화하기 위해 패킹 버블 차트(Packed bubble chart)와 선버스트 그래픽(sunburst graphic)을 제안하였다.

패킹 버블 차트는 데이터의 빈도나 중요도를 시각적으로 표현하는 방법이다. 그 외에도, 선버스트 그래픽을 통해 계층 구조를 표현함으로써 개별 키워드와 주제 간의 관계를 파악할 수 있다.

### III. 제안 방법론

#### 3-1 시스템 개요

본 논문에서는 고객의 다양한 질문 데이터를 효과적으로 파악할 수 있는 트리 구조 형태의 계층 구조화 방법을 제안한다.

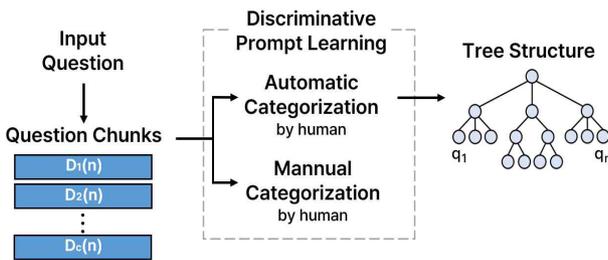
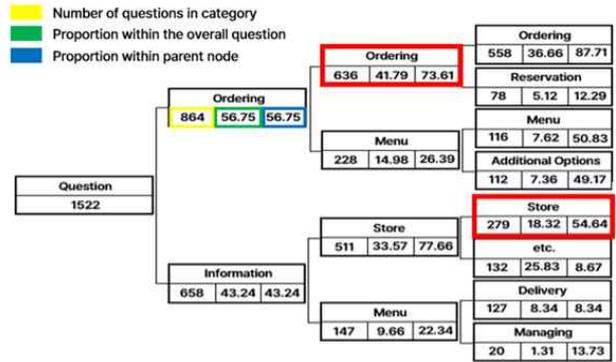


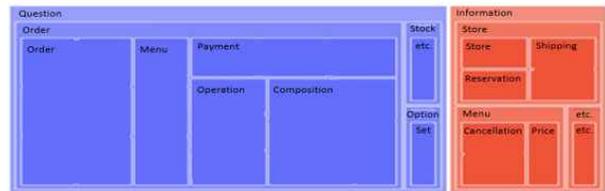
그림 1. 시스템 개요  
Fig. 1. System overview

그림 1은 제안 방법의 개요도이다. 입력으로 주어진 질문 데이터셋은 LLM이 한 번에 분석할 수 있는 적절한 크기의 질문 청크로 분할된 후, 정해진 카테고리로 분류된다. 카테고리는 LLM이 자동으로 생성한 방법과 사람의 임의로 정한 방법을 모두 사용할 수 있다. 질문을 카테고리 분류할 때 각 카테고리에 대한 정의와 각 카테고리에 대한 예시 문장을 프롬프트로 사용하는 분별 프롬프트 학습을 제안한다. 이를 통해 학습 과정이 없는 LLM의 카테고리 분류 성능을 높일 수 있다. 이후 Top-Down 방법과 Bottom-up 방법으로 계층 구조화를 수행해 질문의 상위 개념과 하위 개념을 파악할 수 있도록 트리 구조로 표현한다.

데이터 시각화를 위한 연구에서는 히스토그램, 워드 클라우드, 네트워크 그래프 등의 방법을 사용하고 이를 통해 직관적으로 고객 데이터를 분석할 수 있다[10]. 그러나 이는 데이터의 복잡성을 충분히 반영하지 못하며, 주로 빈도수에 기반한 제한적인 정보를 제공하는 단점이 있다. 또한, 클러스터별로 고객 데이터를 시각화하는 방법은 주로 빈도수가 높은 카테고리만 표시하여 데이터 간의 복잡한 관계나 구조를 명확하게 드러내지 못한다는 한계가 있다[11]. 따라서 본 연구에서는 계층 구조를 표현하고, 각 카테고리 간의 관계와 상대적 중요도를 직관적으로 시각화하기 위해 트리 구조와 트리맵을 활용하고자 한다.



(a) Tree Structure



(b) Treemap

그림 2. 계층 구조화 이후 두 가지 시각화  
Fig. 2. Two visualization methods after hierarchical structuring

그림 2는 전체 구조를 파악하기 위해 적용할 수 있는 두 가지 시각화 방법을 보여준다. 그림 2(a)는 계층 구조의 각 요소를 완전 이진 트리로 표현한 방법이다. 트리를 구성하는 각 노드는 카테고리 명과 해당 카테고리에 속하는 질문의 수를 명시한다. 또한, 트리 내에서 해당 카테고리가 차지하는 비율을 나타내기 위해 전체 질문에서의 비율과 상위 노드 내에서의 비율도 함께 제시한다. 이는 각 카테고리의 상대적 비율과 데이터의 구조적 분포를 명확하게 이해할 수 있도록 한다. 그림 2(b)는 계층 구조의 각 레벨에서 재귀적으로 세분화되는 카테고리를 시각적으로 나타내어, 카테고리 간의 상관관계와 계층구조를 효과적으로 표현한다. 한정된 공간에서 전체 데이터 분포를 볼 수 있으며, 크기와 색상을 통해 각 특징 간의 상관관계를 쉽게 파악할 수 있다.

이렇게 표현된 데이터는 특정 그룹에 속하는 질문을 효과적으로 검색하는 데 활용될 수 있다. 예를 들어, 그림 2(a)에서 '매장'을 선택하면 '매장' 카테고리에 속하는 질문들을 사용자에게 제공하고, '주문'을 선택하면 '주문'의 하위 카테고리인 '주문'과 '예약' 카테고리에 속하는 질문들을 그룹화하여 제공할 수 있다. '주문'과 '매장' 모두 선택하면 주문, 예약, 매장 카테고리에 해당하는 질문을 제공할 수 있다.

제안하는 방법을 통해 사용자는 많은 양의 질문 데이터 중에서 필요한 정보를 빠르게 찾을 수 있으며, 한정된 공간에서 카테고리의 상관관계 및 전체 분포를 한눈에 파악할 수 있다.

#### 3-2 분류 카테고리 생성

본 논문에서는 LLM을 사용해 주어진 문서를 대표할 수 있

는  $k$ 개의 카테고리를 생성한다. 이 과정은 식 (1)로 표현할 수 있다.

$$\begin{aligned}
 f(D) &= f([n_1, n_2, \dots, n_N]) \\
 &= [f(D_1), f(D_2), \dots, f(D_c)] \\
 &= [C_1, C_2, \dots, C_K]
 \end{aligned}
 \tag{1}$$

식 (1)에서  $D$ 는 전체 문서이고,  $N$ 개의 질문으로 이루어져 있다.  $f$ 는 LLM을 사용해  $N$ 개의 질문에 대해서  $K$ 개의 카테고리를 출력하는 함수다. 전체 문서  $D$ 는  $c$ 개의 청크 문서로 분할되어 각각에 대해 LLM 함수가 적용된다.

이렇게  $c$ 개의 청크로 분할하는 이유는 LLM이 분석할 수 있는 프롬프트 크기에 제한이 있기 때문이다. 예를 들어, 구글이 발표한 제미나이 1.0의 경우 문맥(context) 크기는 대략 32,000 토큰이다. 영어의 경우 토큰 1개는 약 4글자, 한글은 토큰 1개가 대략 1.5 자이다. 문맥 크기를 A4 용지 기준으로 변환하면, 3만 2,000개의 토큰은 약 50페이지에 해당한다.

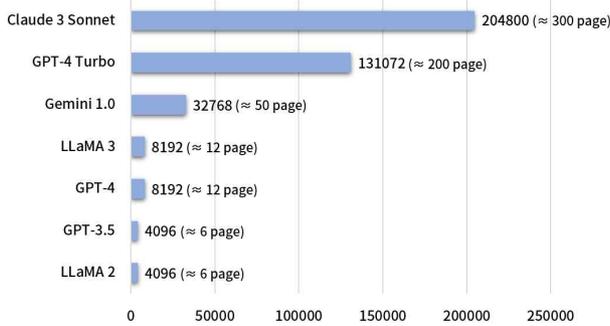


그림 3. LLM의 컨텍스트 크기 비교(괄호 안의 숫자는 대략적인 A4 용지 기준 페이지 수)

Fig. 3. Comparison of LLM Context Sizes (the numbers in parentheses are approximately the number of pages based on A4 paper)

그림 3은 상용 LLM 문맥 크기를 비교한 그래프이다. 엔트 로픽의 클로드 3 소네트는 20만 개 토큰, Open AI의 GPT-4 Turbo는 12만 8,000개 토큰의 문맥을 지원한다. 현재 상용화된 여러 모델은 문맥 크기가 비교적 큰 편이지만, 대형 LLM 이외에도 최근 많은 연구가 이루어지고 있는 소형 언어 모델(sLLM; small Large Language Models)은 여전히 문맥 크기가 작은 경우가 많다[12]. 예를 들어, 메타의 LLaMa3는 약 8,000개의 토큰으로 문맥 크기가 제한되어 있다. 모델이 클수록 비용이 많이 들기 때문에 모델 자체가 작은 환경에서는 문맥 크기를 확대하기 어렵다.

문맥 크기의 제한으로 인해, 식 (1)에서 설명한 바와 같이 전체 질문을 적절한 크기의 청크로 나누어 분석해야 한다. 이후 식 (2)를 사용하여 각 청크에 포함될 수 있는 최대 질문 수를 계산할 필요성이 있다.

$$N_c = \text{floor}\left(\frac{C \times \text{weight}}{w \times t}\right)
 \tag{2}$$

식 (2)에서  $C$ 는 LLM의 최대 문맥 크기를,  $w$ 는 질문 하나 당 평균 단어 수를,  $t$ 는 평균 단어당 토큰 수를 의미한다. 실제로 사용할 수 있는 문맥 크기는 LLM의 최대 문맥 크기에 명령이나 요청 등의 보조 프롬프트를 고려한 가중치를 곱하여 계산한다. 그런 다음, 계산된 문맥 크기를 각 질문의 평균 토큰 수로 나누고 내림하여 정수로 표현한다. 식 (2)를 통해 LLM의 문맥 크기를 초과하지 않도록 조절할 수 있다.

그러나 하나의  $D$ 를  $c$ 개의 청크로 분할하면  $K$ 개의 카테고리를 생성할 수 없다. 예를 들어  $c = 3, K = 3$ 일 때,  $D_1$ 은 ‘Order’, ‘Menu’, ‘Cafeteria’ 카테고리로 분류되고,  $D_2$ 는 ‘Menu’, ‘Service’, ‘Payment’ 카테고리로 분류되고,  $D_3$ 은 ‘Product’, ‘Cafeteria’, ‘Service’ 카테고리로 분류되었다고 가정하면, 이를 다시 3개의 카테고리로 통합할 방법이 필요하다.

본 논문에서는  $D_1$ 부터  $D_3$ 까지 분류된 ‘Order’, ‘Menu’, ‘Cafeteria’, ‘Payment’, ‘Product’, ‘Service’ 카테고리를 모아서, LLM을 사용해 6개의 카테고리를 포괄할 수 있는 3개의 카테고리를 다시 생성한다. 이러한 방법을 통해 수식 (1)처럼 전체 문서  $D$ 를 청크로 분할하더라도 정확히  $K$ 개의 카테고리를 생성할 수 있다.

### 3-3 계층 구조화

본 연구에서는 질문 데이터를  $k$ 개의 카테고리로 분류하는 것에서 나아가, 각 카테고리 간의 관계를 고려하여 계층 구조로 표현한다. 이를 통해 질문 간의 구조적인 특성을 파악할 수 있어 상위 수준에서의 이해를 도모할 수 있다[2].

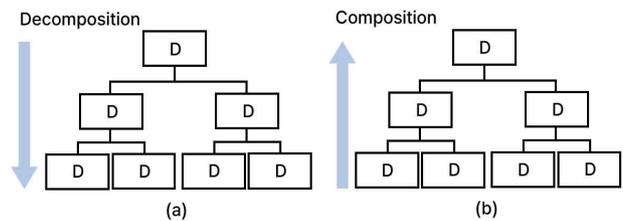


그림 4. 계층 구조화를 위한 두 가지 접근 방식  
Fig. 4. Two approaches to hierarchical structure

본 논문에서는 그림 4에 제시된 두 가지 접근 방식을 사용해 계층 구조화를 검토하였다. 그림 4(a)는 분석할 질문의 수가 많지 않아 전체 문서가 LLM 문맥에 한 번에 들어갈 수 있을 때 적합한 트리 구조로, 전체 질문을  $k$ 개의 카테고리로 분할하고 이를 다시  $k$ 개로 분할하는 작업을 반복하는 Top-down 방식이다. 질문을 카테고리로 분류할 때마다 전체 문서를 다시 분석해야 하기 때문에 트리 구조를 생성하는 계산 복잡도가 높다. 또한 구조를  $k = 2$ 에서  $k = 3$ 으로 변경할 경우, 하위 단계에서 카테고리를 재분류해야 하므로 다양한 조건에

서 활용하기 어렵다는 단점도 있다.

그림 4(b)는 전체 문서 D를 많은 수의 카테고리로 분류하고, 올라가면서 상위 개념의 카테고리 수로 축소해 루트 노드에서 k개의 카테고리가 생성될 때까지 과정을 반복하는 Bottom-up 방식이다. 예를 들어, k = 2인 Top-Down 방식의 트리 구조를 생성하기 위해 처음에 문서 전체를 8개의 카테고리로 분할하고, 그다음 4개로 합쳐지고 마지막으로 2개로 합치는 방법으로 2개씩 분할되는 트리 구조를 생성할 수 있다. Bottom-up 방식은 처음에 많은 카테고리로 세분화하고, 전체 문서가 아닌 카테고리를 줄이는 과정에서 LLM을 사용하기 때문에 전체 문서를 매번 카테고리로 분할하는 Top-Down 방식에 비해 계산 복잡도가 매우 작은 장점이 있다. 다만 Top-Down 방식은 원하는 높이만큼 트리를 생성할 수 있으나, Bottom-up 방식은 데이터가 적을 때 성할 수 있는 트리의 깊이가 제한된다는 단점이 있다.

### 3-4 분별 프롬프트 학습

계층 구조화를 통해 전체 문서 D를 어떤 카테고리로 분할할지 결정한 후 전체 질문 각각이 어느 카테고리에 해당하는지 분류해야 한다. 이를 통해 특정 카테고리에 해당하는 질문들을 검색할 수 있다. 질문의 의도를 파악해 어떤 카테고리에 해당하는지 분류하기 위해 딥러닝 기반의 BERT 알고리즘이 널리 사용되어 좋은 연구 성과를 나타내었다. 그러나 딥러닝 알고리즘은 복잡한 사전 훈련과 미세 조정이 필요하다는 단점이 있다[4].

최근 Chat GPT와 같은 대형 언어 모델(LLM; Large Language Models)의 발전으로 대규모 데이터의 효과적인 분석과 관리가 용이해졌다[13]-[15]. 특히 LLM과의 상호작용을 위한 프롬프트 엔지니어링에 대한 연구가 활발히 진행되고 있다[16]-[19]. 주어진 예시 없이 문장을 분류하는 제로샷(zero-shot) 프롬프트 학습은 성능이 저조하여 이를 개선하기 위한 다양한 학습 방법이 연구되고 있다. 원샷(One-shot) 프롬프트 학습은 하나의 예시만 제공해 모델이 주어진 문맥을 이해하도록 하는 방법이며, 퓨샷(few-shot) 프롬프트 학습은 소수의 예시를 제공해 학습 성능을 향상시키는 방법이다[16].

본 논문에서는 기존의 퓨샷 프롬프트 학습을 확장해 분별 가능한 정보를 포함하는 새로운 방법을 제안한다.

그림 5는 제로샷 프롬프트와 제안된 분별 프롬프트 학습 방법을 통해 질문을 카테고리로 분류하는 과정을 비교하여 나타낸다. 제로샷 프롬프트는 질문과 k개의 카테고리 명, JSON 출력 형식으로 구성되어 있다. LLM은 질문과 목적, 출력 양식을 이해하고 가장 적합한 카테고리를 결정한다. 이러한 제로샷 프롬프트는 각 카테고리의 의미를 다른 카테고리 와 비교할 수 있는 비교 정보가 충분하지 않기 때문에 카테고리 분류 성능이 떨어질 수 있다.

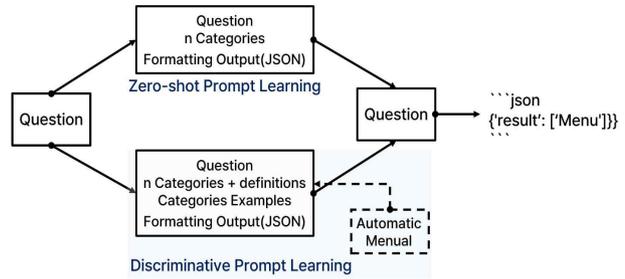


그림 5. 제안하는 분별 프롬프트 학습

Fig. 5. The proposed discriminative prompt learning

제안하는 분별 프롬프트는 분류할 k개의 카테고리 명뿐만 아니라 분류할 카테고리가 분별 가능하도록 카테고리에 대한 구체적인 정의와 예시 문장으로 구성되어 있다. 이러한 프롬프트는 LLM을 이용해 자동으로 생성하거나 사람에 의한 수동으로 생성할 수 있다. 프롬프트를 자동으로 생성할 때 카테고리가 서로 명확하게 구분되도록 요청해야 한다. 예를 들어, 프롬프트에 “k개의 카테고리가 명확하게 구분되도록 정의를 내려주세요.”라는 지시 사항을 포함시키면 LLM은 각각의 카테고리 의미가 명확히 구분되도록 카테고리 정의와 관련 예시를 생성시킨다.

분별 프롬프트 학습의 핵심은 분류할 클래스에 대한 명확한 정의를 통해 각각의 클래스에 대한 범위를 명시적으로 제시하는 것이다. 이를 통해 내부적으로 긍정 정보와 부정 정보, 즉 클래스에 해당하는 정보와 해당하지 않는 정보를 사용해 카테고리 인식 능력을 향상시킬 수 있다. 본 논문에서는 공개용 데이터셋을 이용한 실험을 통해 제안하는 분별 프롬프트 학습으로 분류 성능 어느 정도 향상되는지 실험적으로 제시하고자 한다.

## IV. 실험 및 분석

### 4-1 실험 데이터셋

본 실험에서는 AI 허브에서 제공하는 ‘소상공인 고객 주문 질의-응답 텍스트’ 데이터셋을 활용해 제안하는 알고리즘의 성능을 평가하였다. 이 데이터셋은 무인 상점 서비스 운영 시 발생할 수 있는 다양한 상황에서의 고객 질문과 그에 따른 답변 14만 건으로 구성되어 있으며, 음식점, 슈퍼, 카페 등 14개의 상황별 데이터셋으로 분류되어 있다. 데이터셋은 질의-응답을 이루는 문장을 포함하고 있으며, 대화 순번, 화자 구분, QA 구분, 감성 구분, 상점 카테고리, 질문 의도, 개체명으로 구성되어 있다. 본 실험에서는 사용자의 질문에 대한 의도를 파악하는 데 중점을 두고 있으므로, 질문(Q)과 의도만을 추출하여 사용한다.

표 1. 실험 내용

Table 1. Experimental contents

Field	Experimental Contents	Sessions
Size of Chunk	Relationship between number of questions and number of tokens	4-2
Hierarchical Classification	k=2, 3 Hierarchical Classification using the Top-down method	4-3
	k=2 Hierarchical Classification using the Bottom-up method	4-3
Discriminative Prompt Learning	Automatic and Designated Category Classification	4-4
	Discriminative Prompt Learning	4-4

본 논문에서는 표 1에 명시된 다섯 가지 기초 실험을 수행하였으며, 모든 실험은 Gemini-1.0-pro API를 사용해 진행되었다. 실험 1에서는 문맥 크기에 따른 최적의 질문 수를 도출하기 위해 질문 수와 토큰 수의 관계를 분석하였다. 실험 2와 실험 3은 Top-down 방식과 Bottom-up 방식에 따른 계층 구조화 특성에 관한 실험을 진행하고, k가 2일 때, 3일 때의 계층 구조도 비교하였다. 실험 4에서는 자동 카테고리 분류가 올바르게 수행되었는지 검증하기 위해 데이터셋의 원문 의도와 제안하는 방법으로 생성된 카테고리를 비교 분석하였다. 마지막으로, 실험 5에서는 분별 프롬프트 학습의 효과성을 검증하기 위해 인식률을 계산하고, 성능을 개선하는 실험을 수행하였다.

4-2 질문 수와 토큰 수의 관계

표 2. 질문 수와 토큰 수의 관계

Table 2. Relationship between number of questions and number of tokens

Number of questions	Number of tokens	Number of tokens per question
500	7683	15.37
600	9219	15.37
700	10986	15.69
800	12344	15.43
900	14025	15.58
1000	15570	15.57
1100	17165	15.60
1200	18570	15.47
1300	20168	15.51
1400	21719	15.51
1500	23268	15.51

첫 번째 실험에서는 최적의 질문 수를 도출하기 위해 질문 수와 토큰 수의 관계를 분석하였다. 이 실험을 통해 LLM을 사용하여 문서를 분석할 때, 한 번에 분석할 수 있는 질문 수

를 파악할 수 있다. 표 2의 결과를 보면, 보통 질문 하나당 16개의 토큰으로 이루어져 있음을 알 수 있다. 실험에서 LLM의 문맥 크기는 8,000 이기 때문에 전체 질문을 500개씩 분할하였다. 나머지 토큰 313개(8000-7687)는 프롬프트 학습에 사용할 수 있다.

4-3 Top-down, Bottom-up

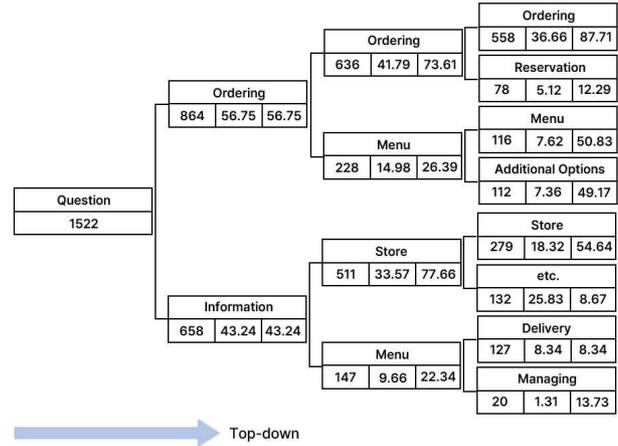


그림 6. Top-down 방식을 이용한 k=2 계층 구조 분류  
Fig. 6. k=2 hierarchical classification using the Top-down method

그림 6은 질문 1522개를 Top-Down 방식으로 계층구조를 생성한 결과이다. 각 노드는 카테고리 명, 분류된 질문 개수와 두 개의 비율 값으로 구성되어 있다. 첫 번째 비율은 전체 질문에서 해당 카테고리 질문 수의 비율이고, 두 번째 비율은 부모 노드의 카테고리 질문에서 해당하는 질문 수의 비율을 의미한다. 전체 질문은 주문과(864개, 56.75%) 정보로(858, 43.24%) 분할되었다. 그리고 주문은 다시 주문과(636개, 73.61%) 메뉴로(228개, 26.39%) 분할되었다. 즉 주문 관련 질문의 많음을 의미한다. 주문 관련 질문은 전체 질문에서는 41.79%(636/1522) 비율이고, 메뉴가 차지하는 비율은 14.98%(228/1522)이다.

그림 6과 그림 7은 Top-down 방식을 사용하여 k=2와 k=3의 차이점을 비교 분석한 결과를 보여준다. 카테고리 수가 적으면 카테고리 간 구분이 명확하지 않을 수 있다. 예를 들어 주문과 메뉴 두 개의 카테고리로 나눌 경우, “~~메뉴 주문할 수 있나요?”는 주문 카테고리에 속하기도 하지만 메뉴 카테고리에도 속할 수 있다. k 값이 작은 경우 카테고리 경계가 명확하지 않을 수 있다. 반면 k 값이 크면 카테고리가 세부적으로 나누어지고 깊이도 증가하여 계층구조가 커진다. 따라서 k 값을 조정하여 더 세부적으로 카테고리를 나누는 것이 효과적임을 알 수 있다.

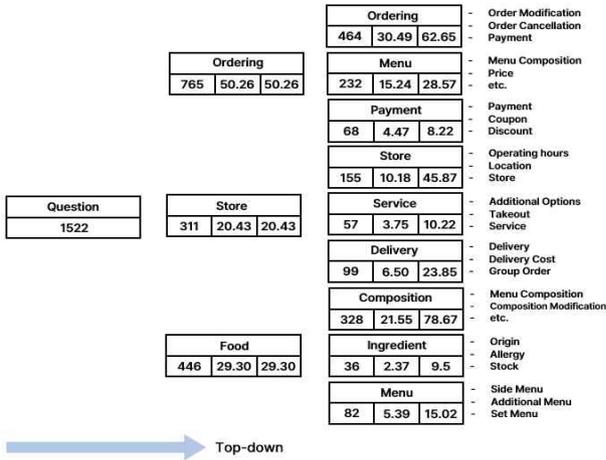


그림 7. Top-down 방식을 이용한 k=3 계층 구조 분류  
 Fig. 7. k=3 hierarchical classification using the Top-down method

Bottom-up 방식에서는 데이터를 서브 셋으로 나누어 개별적으로 카테고리를 분류하고 이를 서브 셋의 결과를 모아서 상위 개념으로 재분류하는 과정이 필요함을 이전 장에서 설명했다. 이를 위해 1,522개의 질문을 약 500개씩 나누어 3개의 서브 셋으로 나누었다. k = 2일 때 각 서브 셋에 속한 질문을 두 개의 카테고리로 분류함을 의미한다. 이를 위해 Bottom-up 방식으로 카테고리를 8개, 4개, 2개로 축소하여 전체 질문에 대한 계층 구조를 생성했다. 각 서브 셋에서 분류된 카테고리가 모두 다르기 때문에, 최종적으로 두 개의 클러스터를 통해 서브 셋의 계층 구조를 통합하여 모든 질문을 분류하였다.

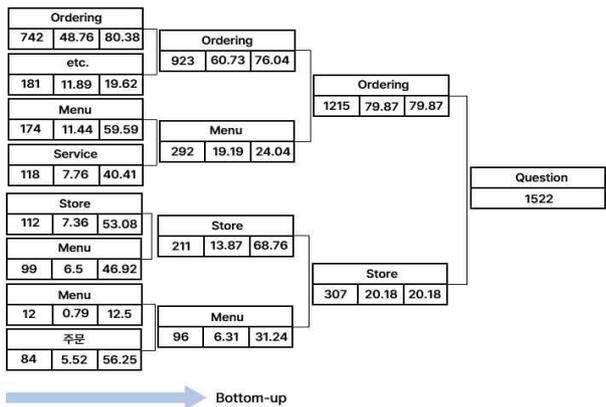


그림 8. Bottom-up 방식을 이용한 k=2 계층 구조 분류  
 Fig. 8. k=2 hierarchical classification using the Bottom-up method

그림 8은 k=2 실험 결과에 따른 계층 구조를 보여준다. 전체 질문은 크게 ‘주문’과 ‘매장’ 두 가지 카테고리로 나누어졌다. 그리고 1,522개의 질문 중 ‘주문’ 카테고리에는 1,215개, ‘매장’ 카테고리에는 307개이다. 트리의 깊이가 깊어지면서

카테고리가 점점 더 세분화되어 각 질문을 보다 구체적으로 분류하고 있다.

표 3은 그림 8의 계층 구조에서 최상의 카테고리 융합 결과를 나타낸다. ‘주문’과 ‘매장’ 카테고리가 각 서브 셋의 어느 카테고리를 포함하고 있는지 표 3을 통해 확인할 수 있다. ‘주문’ 카테고리는 ‘주문’, ‘주문-배달’, ‘주문-메뉴’와 같이 주문과 관련된 정보로 구성되어 있다. ‘매장’ 카테고리는 ‘매장’, ‘메뉴-구성’, ‘매장-배달’과 같이 매장에 관한 질문이 주를 이룬다.

표 3. 카테고리 융합 결과  
 Table 3. Category convergence

	Category1(questions)	Category2(questions)
Set1	Order(403)	Store(97)
Set2	Order-Delivery(338)	Menu-Composition(162)
Set3	Order-Menu(388)	Store-Delivery(134)

제안하는 방법을 사용해 자동으로 카테고리 계층 구조를 생성할 때 부정확한 카테고리가 생성될 수 있다. 가장 대표적인 경우는 불필요한 ‘기타’ 카테고리 생성이다. 현재 분류할 질문들이 너무 다양한 분야의 질문으로 구성되어 있을 때 발생한다. 예를 들어, “현재 진행 중인 프로모션이 무엇인가요?”와 같은 질문은 초기에 ‘기타’로 분류되었지만, 세부 분석을 통해 ‘프로모션’ 카테고리로 재분류된다.

모든 질문은 하나의 카테고리로 명확히 분류되지만, 어떤 질문은 의미상으로 여러 카테고리에 속할 수 있다. 예를 들어, “특정 메뉴를 주문할 수 있나요?”라는 질문은 ‘메뉴’와 ‘주문’ 카테고리 모두에 해당할 수 있다. 이 질문은 고객이 특정 메뉴의 존재 여부를 묻는 동시에, 그 메뉴를 실제로 주문할 수 있는지에 대한 정보를 원하기 때문에 유사한 카테고리가 계층을 달리해 생성될 수 있다. 또 다른 예로, “특정 메뉴의 가격은 얼마인가요?”와 “특정 메뉴를 주문할 때 추가 요금이 발생하나요?”는 모두 메뉴와 관련된 질문이지만, 하나는 가격에, 다른 하나는 주문과 관련된 추가 비용에 관한 질문이 될 수 있다. 향후 이러한 문제는 전체 대화상에서 문맥을 파악해 정확히 분류할 수 있는 방법에 대한 연구가 필요하다.

#### 4-4 분별 프롬프트 학습에 대한 효과성 검증

제안 방법의 효과성을 검증하기 위해 두 가지 실험을 진행하였다. 첫 번째 실험에서는 계층 구조를 구축하는 과정에서 카테고리 분류 성능을 검증하였다. 표 4에는 동일한 데이터셋에 대해 카테고리의 자동 분류 결과와 지정 분류 결과를 비교 분석한 내용을 제시하였다.

표 4. 자동 분류와 지정 분류의 비교

Table 4. Comparison of automatic and designated category classification

Category Classification		Category
Automatic (depth=2)	k=2 (8 EA)	Order, Reservation, Menu, Additional Option, Store, etc., Delivery, Operation
	k=3 (27 EA)	Order Change, Order Cancellation, Calculation, Menu Configuration, Price, Other, Calculation, Coupon, Discount, Business hours, Location, Store, Additional Options, Packaging, Service, Delivery, Delivery Cost, Group Order, Configuration Change, Origin, Allergy, Stock, Side Menu, Extra Menu, Set Menu
Designated (18EA)		Payment Receipt, Store Information, Membership Usage, Shipping Method, Shipping Cost, Shipping Error, Reservation Method, Reservation Cancellation, Product Price, Product Composition, Material, Origin, Inventory, Product Recommendation, Product Quality, Order, Change Order, Withdraw Order

자동 분류는 제안 방법을 의미하며, 지정 분류는 원본 데이터셋에 매핑된 카테고리를 나타낸다. 자동 분류는 깊이가 2 일 때 k가 2인 경우와 k가 3인 경우로 나누어 정리하였으며, 각각 8개와 27개의 카테고리로 분류되었다. 이를 통해 k 값 설정에 따라 계층 구조 분류의 정확성과 세분화 수준이 달라짐을 알 수 있다.

자동 분류와 지정 분류를 비교 분석한 결과, 일부 형태가 약간 변형되거나 의미는 유지한 채 카테고리 명이 변경된 경우도 있었다. 예를 들어, 지정 분류에서 ‘제품 구성’이 자동 분류에서는 ‘메뉴 구성’으로 변경된 경우가 있었으나, 전체적으로 볼 때 지정 분류와 거의 유사하다는 점을 알 수 있다.

두 번째 실험에서는 LLM의 질문 의도 분석 능력을 검증하기 위해 LLM이 분류한 카테고리 와 실제 데이터셋의 카테고리를 비교하여 인식률을 계산하였다.

표 5. SG, DPL, 및 DPL+SG 적용에 따른 인식률 비교

Table 5. Comparison of recognition rates by application of SG, DPL, and DPE+SM

SG Before/after		DPL	DPL + SG
78.10 %	81.52 %	91.59 %	97.75 %

표 5는 SG(Similarity Grouping) 전후, DPL, 그리고 DPL과 SG를 함께 적용한 방식의 인식률(%)을 비교한 결과를 보여준다. SG는 유사한 카테고리 개념을 그룹화한 것이며, DPL은 분별 프롬프트 학습을 의미한다.

특정 카테고리는 ‘매장 구조’와 ‘매장 시설’처럼 사람이 보기에도 두 카테고리의 의미가 모호한 경우가 존재한다. 효율

적인 카테고리 분류를 위해 유사한 개념을 가진 카테고리를 하나로 통합하였으며, 그 결과는 표 6에 나타났다.

표 6. SG 적용 예시

Table 6. SG application example

Group	Category
Change order	Change Purchase, change order
Product Composition	Product Composition, Product General
Inventory	Product Stock Request, Inventory, Inventory Check
Store Information	Store Structure, Store Facilities, Store Use, Store Confirmation, Store Information
etc.	Payment Receipt Request, Membership Usage Request, Shipping Method, Shipping Cost, Shipping Error, Reservation Method, Reservation Cancellation, Product Price, Material, Origin, Product Recommendation, Product Quality, Order, Product Cancellation

예를 들어, ‘매장 구조’와 ‘매장 시설’은 의미가 중복되며, ‘매장 이용’, ‘매장 확인’, ‘매장 정보’에 포함된 질문의 내용이 유사하므로, 이를 ‘매장 정보’라는 하나의 포괄적인 카테고리로 그룹화할 수 있다. 또한, ‘제품 구성’과 ‘제품 일반’ 카테고리는 모두 제품의 일반적인 특성과 관련이 있어, 질문 수가 더 많은 ‘제품 구성’ 카테고리로 통합하였다. 실험 결과, 기존의 25개 카테고리가 18개로 줄었으며, 표 5에서 볼 수 있듯이, 분류 성능은 SG를 적용하기 전보다 3.42% 향상된 81.52%로 나타났다.

성능 향상을 위한 또 다른 방법으로 DPL을 적용하였다. 카테고리 분류를 요청할 때, 각 카테고리를 명확하게 구분하기 위한 지시 사항과 함께 카테고리 정의 및 5개의 예시 문을 제공하였다. 표 5의 결과에 따르면, DPL을 적용한 방식은 SG를 적용한 방식보다 10.07% 향상된 91.59%의 인식률을 보였다. 이는 DPL이 카테고리 분류에 효과적이며, 성능 개선에 충분히 기여한다는 점을 보여준다.

성능 개선을 위해 DPL과 SG를 함께 적용하였다. 표 5의 결과에 따르면, 인식률은 97.75%로, SG를 적용했을 때보다 16.23%, DPL을 적용했을 때보다 19.65% 향상되었다. 각각의 방식을 개별적으로 적용하는 것보다 두 방식을 함께 적용했을 때 LLM의 질문 의도 분류 성능이 크게 향상된다는 것을 알 수 있다. 그림 9의 혼동 행렬을 보면, 대부분의 카테고리는 올바르게 분류되었지만, 일부 카테고리는 예측이 어렵거나 유사한 다른 카테고리로 잘못 분류된 경우가 있었다. 예를 들어, ‘제품 구성’ 카테고리에 속하는 9,240개의 질문은 정확하게 예측되었지만, 일부 질문은 ‘제품 소재’나 ‘제품 주문’ 등의 카테고리로 잘못 분류되었다. 이는 문맥적으로 유사한 카테고리 간의 혼동에서 비롯된 결과로 볼 수 있다.

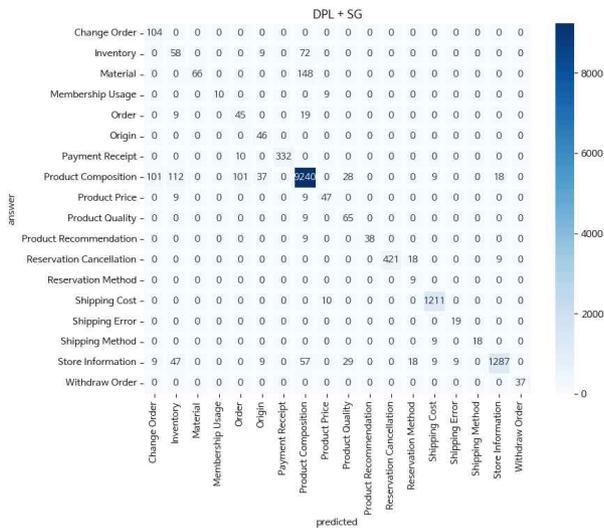


그림 9. 제안 방법의 효율성 분석을 위한 혼동 행렬  
 Fig. 9. Confusion matrix for efficiency analysis of the proposed method

## V. 결 론

본 연구에서는 LLM을 활용하여 고객의 다양한 질문 데이터를 카테고리 분류하고, 이를 계층 구조로 표현하여 직관적으로 이해할 수 있는 시각화 방법을 제안하였다. 기존 딥러닝 기반 연구의 한계를 극복하고, 향상된 프롬프트 학습을 통해 보다 정확한 카테고리 분류가 가능함을 검증하였다. 이러한 접근 방식은 기업이 고객의 요구 사항을 정확하게 파악하고 신속하게 대응하는 전략을 수립하는 데 기여할 수 있을 것이다.

특히 제안 방법은 LLM을 활용하여 질문 데이터를 자동으로 분류하고, 계층 구조를 통해 질문 간의 관계를 명확히 파악할 수 있도록 한다. 이를 통해 대규모 데이터의 복잡성을 줄여, 사용자는 데이터를 보다 구조적으로 이해할 수 있다.

제안 방법의 효과성을 입증하기 위한 실험에서는 카테고리 분류 성능을 평가하고자 카테고리 인식률을 계산하였다. 각 카테고리의 명확한 정의와 예시 문장을 프롬프트로 생성하는 분별 프롬프트 학습을 적용한 방법은 기존 방법의 78.10%에 비해 카테고리 분류 성능이 91.59%로 향상되었다. 더 나아가, 분별 프롬프트 학습과 유사 개념 그룹화 방법을 함께 적용한 결과, 성능은 97.75%로 개선되었다.

제안 방법은 기업이 고객의 요구사항을 효과적으로 파악하고, 이를 기반으로 전략적 의사 결정을 지원하는 데 기여할 수 있다. 그러나, 제안 방법은 특정 데이터셋을 중심으로 입증되었기 때문에 여러 도메인에서의 일반화 가능성을 추가적으로 검증하고, RAG(Retrieval-Augmented Generation) 방식을 활용하여 다양한 텍스트 데이터를 효과적으로 분류할 수 있는 연구를 진행할 예정이다.

## 참고문헌

- [1] A. Macquisten, A. M. Smith, and S. J. Fernstad, "Evaluation of Hierarchical Visualization for Large and Small Hierarchies," in *Proceedings of the 24th International Conference Information Visualisation (IV)*, Melbourne, Australia, pp. 166-173, September 2020. <https://doi.org/10.1109/IV51561.2020.00036>
- [2] P. Dysert and S. Prabhala, "Customer Hierarchy of Needs: Customer Centric Approach to Agile Product Development," in *Proceedings of the 11th International Conference on Design, User Experience, and Usability (DUXU 2022)*, Online, pp. 3-11, June-July 2022. [https://doi.org/10.1007/978-3-031-05897-4\\_1](https://doi.org/10.1007/978-3-031-05897-4_1)
- [3] F.-M. Hsu, L.-P. Lu, and C.-M. Lin, "Segmenting Customers by Transaction Data with Concept Hierarchy," *Expert Systems with Applications*, Vol. 39, No. 6, pp. 6221-6228, May 2012. <https://doi.org/10.1016/j.eswa.2011.12.005>
- [4] Y. Chae and T. Davidson, "Large Language Models for Text Classification: From Zero-Shot Learning to Instruction-Tuning," *SocArXiv*, August 2023. <https://doi.org/10.31235/osf.io/sthwk>
- [5] A. Petukhova, J. P. Matos-Carvalho, and N. Fachada, "Text Clustering with LLM Embeddings," arXiv:2403.15112v1, March 2024. <https://doi.org/10.48550/arXiv.2403.15112>
- [6] K. R. Yoo and Y.-S. Jeong, "Answering User Queries on Online Learning Platforms through Natural Language Processing and Keyword Visualization Using Word Cloud," in *Proceedings of the Korean Society of Computer Information Conference*, Asan, pp. 351-354, January 2024.
- [7] Y. Wu, F. Wei, S. Liu, N. Au, W. Cui, H. Zhou, and H. Qu, "OpinionSeer: Interactive Visualization of Hotel Customer Feedback," *IEEE Transactions on Visualization and Computer Graphics*, Vol. 16, No. 6, pp. 1109-1118, November-December 2010. <https://doi.org/10.1109/TVCG.2010.183>
- [8] T. A. Keahey, *Using Visualization to Understand Big Data*, IBM Corporation, Somers: NY, YTW03349-USEN-00, September 2013.
- [9] J. Roudier, O. Saucier, R. Kinder, and M. Jans, "What to Do with All Those Open-Ended Responses? Data Visualization Techniques for Survey Researchers," *Survey Practice*, August 2021. <https://doi.org/10.29115/SP-2021-0008>
- [10] T. V. Rajini Kanth, D. V. Nagarjana Devi, K. Neha, T. V. S. Aneesh, and T. Gupta, "Analysis of Customer Sales Data Using Numerous Data Visualization Techniques," *SSRN*, April 2024. <https://doi.org/10.2139/ssrn.4781944>
- [11] A. Strehl and J. Ghosh, "Relationship-Based Clustering and

Visualization for High-Dimensional Data Mining,” *INFORMS Journal on Computing*, Vol. 15, No. 2, pp. 208-230, May 2003. <https://doi.org/10.1287/ijoc.15.2.208.14448>

- [12] Meta. Introducing Meta Llama 3: The Most Capable Openly Available LLM to Date [Internet]. Available: <https://ai.meta.com/blog/meta-llama-3/>.
- [13] J. Li, B. Hui, G. Qu, J. Yang, B. Li, B. Li, ... and Y. Li, “Can LLM Already Serve as a Database Interface? A Big Bench for Large-Scale Database Grounded Text-to-SQLs,” arXiv:2305.03111v3, November 2023. <https://doi.org/10.48550/arXiv.2305.03111>
- [14] X. Lin, W. Wang, Y. Li, S. Yang, F. Feng, Y. Wei, and T.-S. Chua, “Data-Efficient Fine-Tuning for LLM-Based Recommendation,” in *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '24)*, Washington, DC, pp. 365-374, July 2024. <https://doi.org/10.1145/3626772.3657807>
- [15] X. Zhou, X. Zhao, and G. Li, “LLM-Enhanced Data Management,” arXiv:2402.02643, February 2024. <https://doi.org/10.48550/arXiv.2402.02643>
- [16] S. Rahman, S. Khan, and F. Porikli, “A Unified Approach for Conventional Zero-Shot, Generalized Zero-Shot, and Few-Shot Learning,” *IEEE Transactions on Image Processing*, Vol. 27, No. 11, pp. 5652-5667, November 2018. <https://doi.org/10.1109/TIP.2018.2861573>
- [17] J. White, Q. Fu, S. Hays, M. Sandborn, C. Olea, H. Gilbert, ... and D. C. Schmidt, “A Prompt Pattern Catalog to Enhance Prompt Engineering with ChatGPT,” arXiv:2302.11382, February 2023. <https://doi.org/10.48550/arXiv.2302.11382>
- [18] J. D. Velásquez-Henao, C. J. Franco-Cardona, and L. Cadavid-Higueta, “Prompt Engineering: A Methodology for Optimizing Interactions with AI-Language Models in the Field of Engineering,” *DYNA*, Vol. 90, No. 230, pp. 9-17, November 2023. <https://doi.org/10.15446/dyna.v90n230.111700>
- [19] P. Sahoo, A. K. Singh, S. Saha, V. Jain, S. Mondal, and A. Chadha, “A Systematic Survey of Prompt Engineering in Large Language Models: Techniques and Applications,” arXiv:2402.07927, February 2024. <https://doi.org/10.48550/arXiv.2402.07927>



**이채별(Chae-Byeol Lee)**

2021년 : 강남대학교 (공학사-ICT공학)

2021년~현 재: 강남대학교 ICT공학부 학사과정  
※ 관심분야 : 기계학습, 컴퓨터비전, 생성모델 등



**최권택(Kwon-Taeg Choi)**

2006년 2월 : 연세대학교, 컴퓨터공학과(공학석사)

2011년 2월 : 연세대학교, 컴퓨터공학과(공학박사)

2016년 3월~현 재: 강남대학교 ICT융합공학부 교수  
※ 관심분야 : 가상현실, 기계학습, 컴퓨터비전, 생성모델, Web 등