USER INTENT ANALYSIS AND BEHAVIOR MODELING FOR CONVERSATION AND RECOMMENDATION

LU FAN

PhD

The Hong Kong Polytechnic University

2026

The Hong Kong Polytechnic University

Department of Data Science and Artificial Intelligence

User Intent Analysis and Behavior Modeling for Conversation and Recommendation

Lu Fan

A thesis submitted in partial fulfilment
of the requirements for the degree of
Doctor of Philosophy

Aug 2025

CERTIFICATE OF ORIGINALITY

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgement has been made in the text.

_____ (Signed)

Lu Fan _____ (Name of student)

Abstract

Understanding and modeling user intent and behavior are foundational challenges in the development of intelligent conversational AI and recommender systems. As dialogue systems and personalized recommendation platforms become increasingly prevalent, the ability to accurately identify user intentions and effectively capture user behavior patterns is critical for delivering adaptive, user-centric services. However, these systems face significant hurdles: conversational AI must contend with the rapid emergence of new user intents, meanwhile, recommender systems must model complex, evolving user preferences from behavioral data.

Motivated by these challenges, this thesis explores two complementary research directions: (a) *user intent analysis* in conversational systems, and (b) *user behavior modeling* for recommendation. In the first direction, we advance the state of intent detection, with a particular focus on zero-shot and unknown intent classification. We propose novel methods to reconstruct capsule networks for zero-shot intent classification, introducing dimensional attention mechanisms to address polysemy and leveraging latent information to improve generalization to unseen intents. Additionally, we develop a semantic-enhanced Gaussian mixture model for unknown intent detection, integrating it with zero-shot intent classifiers to further boost performance. Recognizing the limitations of existing approaches in new intent

discovery, we present LANID, a framework that harnesses the semantic power of large language models to guide lightweight encoders, enabling scalable and effective identification of novel intents in task-oriented dialogue systems.

In the second research direction, we address the challenge of modeling user behavior for personalized product search and sequential recommendation. We propose graph-based approaches that capture both local and global user behavior patterns, utilizing user successive behavior graphs and efficient graph convolution techniques to enrich product representations and uncover collaborative signals. Furthermore, we introduce a neighborhood-based hard negative mining strategy for sequential recommendation, exploiting structural information in user-item interaction graphs to enhance negative sampling and improve model training.

Extensive experiments across multiple real-world datasets and benchmarks demonstrate the effectiveness and generalizability of our proposed methods, consistently outperforming strong baselines in both conversational AI and recommender system tasks. By tackling the fundamental problems of intent recognition and user behavior modeling, this thesis contributes novel solutions that advance the adaptability, intelligence, and user-centricity of next-generation conversational and recommendation systems.

Publications arising from the thesis

[i] **Fan, L.**, Yan, G., Li, Q., Liu, H., Zhang, X., Lam, A. Y., & Wu, X. M. (2020, July). Unknown Intent Detection Using Gaussian Mixture Model with an Application to Zero-shot Intent Classification. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (pp. 1050-1060).

[ii] **Fan, L.**, Pu, J., Zhang, R., & Wu, X. M. (2024, May). LANID: LLM-assisted New Intent Discovery. In Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (pp. 10110-10116).

[iii] **Fan, L.**, Li, Q., Liu, B., Wu, X. M., Zhang, X., Lv, F., ... & Yang, K. (2022, April). Modeling User Behavior with Graph Convolution for Personalized Product Search. In Proceedings of the ACM Web Conference 2022 (pp. 203-212).

[iv] **Fan, L.**, Pu, J., Zhang, R., & Wu, X. M. (2023, July). Neighborhood-based Hard Negative Mining for Sequential Recommendation. In Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 2042-2046).

[iv] Liu, H., Zhang, X., **Fan, L.(Equal contribution)**, Fu, X., Li, Q., Wu, X. M., & Lam, A. Y. (2019, November). Reconstructing Capsule Networks for Zero-shot Intent Classification. In Proceedings of the 2019 Conference on Empirical

Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (pp. 4799-4809).

Acknowledgements

Over the past six years, I have encountered setbacks and challenges in my academic journey. I have made mistakes and have often struggled with issues such as procrastination, fear, and avoidance. Without the unwavering support and understanding of Professor Wu, my parents, and my husband, I would not have been able to persevere and reach this point.

I would like to express my sincere gratitude to Professor Wu for your patient guidance, encouragement, care, and support. I am deeply thankful to my parents and my husband for their unconditional love and understanding.

I am profoundly appreciative of all that you have done for me. I aspire to become a stronger and more resilient individual in the future.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Conversational AI and recommender systems have become integral to modern digital platforms, enabling personalized interactions and content delivery at scale. These technologies rely on understanding user needs and preferences to provide relevant and engaging experiences. However, accurately interpreting user intent and effectively modeling user behavior remain core challenges in both fields. This thesis focuses on user intent understanding across two domains: (a) user intent analysis in conversations, and (b) user behavior modeling for recommendations. The first part advances intent detection by addressing the recognition of novel intents in conversational systems with limited training data, allowing adaptation to new scenarios. The second part investigates graph-based approaches to capture and utilize user behavior patterns, thereby improving recommendation performance.

1.1 User Intent Analysis in Conversations

User intent in a dialogue system refers to the underlying goal or purpose behind a user's input or utterance. It represents what the user wants to achieve through their interaction with the system. Understanding user intent lies at the heart of building effective conversational and dialogue systems. As conversational AI becomes increasingly prevalent in applications ranging from virtual assistants to customer service bots, the ability to accurately identify user intent from utterances is critical for driving downstream decision-making and delivering satisfactory user experiences. However, the dynamic nature of user interests and the rapid emergence of new intents present significant challenges for intent analysis. Traditional supervised learning approaches, which rely on large volumes of annotated data [63, 24], struggle to keep pace with the evolving landscape of user needs, making it important to develop methods that can recognize and adapt to novel intents with minimal supervision.

Our research in user intent analysis follows a progressive trajectory, beginning with foundational work on unknown intent detection [33] and zero-shot intent classification [74] using traditional models, and culminating in the integration of large language models (LLMs) [32] to empower lightweight, domain-specific models for new intent discovery. This progression reflects both the evolving technical landscape and the growing demand for scalable, efficient, and adaptive intent recognition systems.

While there have been some pioneering works studying the open-world classification problem in natural language processing [34, 100], only a few methods are designed for unknown intent detection. To our knowledge, the first work is by

Lin et al. [70], in which the authors use large margin cosine loss (LMCL) to learn deep discriminative features and then feed them to a density-based outlier detection algorithm to identify unknown intents. Although this method performs well on some benchmark datasets, it has two limitations. (1) In training, LMCL ignores the prior knowledge of class labels, while it has been shown that label correlations captured in the embedding space can improve prediction performance, especially in the zero-shot learning scenarios [85, 76]. (2) LMCL computes the cosine distance between embeddings in the feature space and trains with a softmax cross-entropy loss, making the embedding distribution of each class long and narrow [110], which may be less suitable for applying density-based anomaly detection algorithms to detect unknown intents. To address these limitations, we propose a novel semantic-enhanced Gaussian mixture model (SEG) for unknown intent detection. In contrast to the softmax function, the Gaussian mixture model enforces embeddings to form ball-like dense clusters in the feature space, which may be more desirable for outlier detection, especially when using density-based outlier detection algorithms. Furthermore, we propose to inject the semantic information of class labels into the Gaussian mixture distribution by assigning the embeddings of class labels or descriptions to be the means of the Gaussians. This enables SEG to learn more class-concentrated embeddings that can benefit downstream outlier detection. We further use a large margin loss to make SEG learn more discriminative features and employ a density-based outlier detection algorithm, Local Outlier Factor (LOF) [13] to detect unknown intents.

Identifying unknown intents is not enough for some application scenarios where it is important to know what exactly the new intents are, e.g., zero-shot intent classification. A few zero-shot learning approaches attempted to address the

aforementioned challenges for intent classification and to make them capable of recognizing novel intents whose instances are not present during training. Early works [35, 36, 130] usually require some external resources like label ontologies or manually defined attributes. However, the external resources are usually unavailable, as they require a lot of extra time and human labour to obtain. To implement zero-shot intent classification more easily and intelligently, recent works rely more on the word embeddings of intent labels, which can be easily pretrained on text corpus. Methods proposed by [21, 60] utilize neural networks to project data instances and intent labels to the same semantic space, and then measure their similarity. However, learning a good projection function for diverse user expressions is difficult, especially in some complex domains such as medical queries [137]. Unlike previous models, IntentCapsNet [119] employs capsule networks to extract high-level semantic features, and then transfers the prediction vectors for seen intents to unknown intents. IntentCapsNet achieved fairly good performance in standard zero-shot intent classification tasks, however, when it comes to generalized zero-shot learning, we observed that it doesn't have the generalization ability to discriminate unknown classes.

In this thesis, we propose to reconstruct capsule networks for zero-shot intent classification (ReCapsNet-ZS), which effectively addresses the limitations of IntentCapsNet and adapts well to the generalized zero-shot intent classification task. ReCapsNet-ZS consists of two parts. First, it introduces a dimensional attention mechanism to tackle polysemy, which helps to extract more representative and semantic features. Second, it computes the similarities between unseen and seen intents by utilizing the rich latent information hidden in the utterances, and then constructs the transformation matrices for unseen intents with the computed

similarities and the trained transformation matrices of seen intents, which greatly improves the generalization ability to identify unseen intents.

In addition, we propose to integrate SEG as an unknown intent identifier into the generalized zero-shot intent classification pipeline. The basic idea is that correctly identifying whether the intent of an utterance is known or unknown will make the subsequent intent classification task much easier. We conduct a case study on a state-of-the-art zero-shot intent classification method, ReCapsNet [74]. The results show that incorporating SEG successfully improves the performance of ReCapsNet by a large margin. It even pushes the performance to a practical level on the SNIPS dataset [24].

Despite these advances, deploying large-scale models for intent discovery in production environments remains resource-intensive and raises concerns around privacy and latency [57]. To bridge the gap between the rich generalization capabilities of LLMs [84, 47] and the efficiency requirements of real-world systems, we introduce LANID [32] (LLM-Assisted New Intent Discovery). LANID distills the semantic knowledge encoded in LLMs into lightweight, in-domain encoders through contrastive learning. By strategically sampling informative utterance pairs and leveraging LLMs to provide supervision signals, LANID enables small models to efficiently discover and represent new intents, even in low-resource settings. This approach not only enhances the adaptability and scalability of intent recognition systems but also demonstrates the practical utility of small sample learning when guided by powerful, general-purpose language models.

In summary, our work advances the field of user intent analysis in conversations by:

- We propose a semantic-enhanced Gaussian mixture model (SEG) for un-

known intent detection by incorporating class semantic information into a Gaussian mixture distribution;

- We explore ways to improve existing generalized zero-shot intent classification systems with an unknown intent identifier. To the best of our knowledge, it is the first attempt to apply unknown intent detection in this task.

- We leverage LLMs to empower lightweight models for new intent discovery through contrastive learning. Together, these contributions lay the groundwork for conversational AI systems that are both adaptive to evolving user needs and practical for deployment in dynamic, real-world environments.

1.2 User Behavior Modeling for Recommendation

In the era of information overload, recommender systems have become indispensable tools for guiding users toward relevant content and products. Central to the effectiveness of these systems is the accurate modeling and analysis of user behavior, which encapsulates the dynamic and multifaceted nature of user preferences. As users interact online through behaviors including browsing, clicking, purchasing, or other forms of engagement, they generate rich behavioral data that, if properly leveraged, can significantly enhance recommendation quality. However, the inherent complexity and diversity of user behaviors present substantial challenges for effective modeling. Noisy signals, data sparsity, and the intricate relationships among different types of user actions often hinder the extraction of meaningful preference patterns.

Recent advances have highlighted the potential of graph-based approaches in

addressing these challenges. Graph structures naturally encode the complex and multi-relational interactions between users, items, and contextual entities, offering a powerful framework for capturing both local and global behavioral patterns. By representing user behaviors and item transitions as nodes and edges in a graph, it becomes possible to model not only direct interactions but also higher-order relationships and collaborative signals that are otherwise difficult to capture with traditional sequential or content-based methods.

Within this context, two complementary lines of research have emerged, both leveraging graph-based modeling to better exploit the relationships among user behaviors for recommendation tasks, yet approaching the problem from distinct perspectives.

The first line of work focuses on personalized product search in e-commerce platforms, where user-item interactions are diverse and product representations are often limited to short, uninformative texts. Traditional methods that rely solely on textual data or truncated behavior sequences struggle to capture the full spectrum of user preferences, especially when long-term histories introduce noise or short-term behaviors lack sufficient signals. To address these limitations, recent studies propose constructing user successive behavior graphs (SBG) that aggregate short-term actions across all users, forming a global structure rich in relational information. By applying graph convolutional networks (GCNs) [141] to these graphs, it becomes possible to learn enriched product representations that encapsulate both local and global user behavior patterns. This approach not only mitigates the sparsity of purchase data but also enables the discovery of implicit preference signals through high-order connectivity, ultimately leading to more accurate and personalized product recommendations. Notably, methods

such as DREM [2] and GraphSRRL [75] have explored unified knowledge graphs and explicit structural relationship modeling, respectively, while recent advances integrate graph convolution with attentive models like ZAM [1] to further enhance user preference modeling.

The second line of research addresses the sequential recommendation (SR) task, where the goal is to predict a user's next likely action based on their historical behavior sequence. A critical yet underexplored aspect of SR is the selection of informative negative samples during model training. Most existing methods adopt uniform negative sampling [22, 41, 118, 56, 124], or employ generative approaches [114, 135, 115, 54], but often overlook the structural information embedded in user behavior sequences. We construct a global weighted item transition graph (WITG) from user interactions. This graph enables us to analyze the neighborhood overlap between items. Using this structural information, we can identify "hard negatives"—items that are structurally similar to the target item but have not been interacted with by the user. A graph-based negative sampling strategy, such as the Graph-based Negative sampling based on Neighborhood Overlap (GNNO), leverages this structural information to select negatives that are more informative for model learning. This not only enhances the discriminative power of the recommendation model but also provides a principled way to balance easy and hard negatives through curriculum learning. Compared to existing region-based negative sampling methods such as RecNS [127], GNNO explicitly utilizes the neighborhood structure in sequential data and samples negatives based on the Jaccard similarity between item neighborhoods.

Both approaches underscore the pivotal role of user behavior analysis in modern recommender systems and demonstrate the versatility of graph-based modeling in

capturing the nuanced relationships among user actions. By integrating structural information at different granularities, whether through global behavior graphs for preference modeling or neighborhood analysis for negative sampling. These methods advance the state-of-the-art in recommendation by addressing key challenges such as data sparsity, noise, and the need for more informative training signals. Collectively, they highlight the importance of exploiting the relational structure of user behaviors, paving the way for more robust and effective recommendation systems.

In summary, our work advances the field of user behavior modeling in conversations by:

- We introduce SBG, a method that leverages graph convolution to capture both local and global patterns in sequential user behavior, thereby enhancing user preference modeling. To the best of our knowledge, this is the first attempt to apply graph convolution for improving product search.

- As far as we know, this is the first research to explore the structural characteristics of negative samples in sequential recommendation. Our findings reveal that item pairs with different degrees of neighborhood overlap on WITG display unique properties, providing an important signal for distinguishing hard negatives in sequential recommendation tasks.

- We present GNNO, which adaptively selects negative samples for each item according to their relative Jaccard similarity on WITG. Additionally, we utilize contrastive learning to regulate the maximum difficulty of negative samples during training.

1.3 Thesis Organization

The remainder of this thesis is organized as follows. Chapter 2 reviews related work in two domains: user intent analysis in dialogue systems and user behavior modeling in recommender systems. Chapters 3 to 5 focus on intent analysis in dialogue systems under few-shot scenarios. Specifically, Chapter 3 explores how pre-trained models can be leveraged to endow traditional models with the capability of unknown intent detection. Chapter 4 investigates zero-shot intent classification, extending the conventional zero-shot setting to a more realistic generalized zero-shot model. Chapter 5 addresses the practical new intent discovery (NID) setting and explores the application of large language models (LLMs) within this paradigm. Chapters 6 and 7 shift the focus to recommender systems, leveraging graph-based approaches to model complex user relationships and improve recommendation performance. Chapter 6 integrates user behaviors across multiple temporal scales. Chapter 7 further utilizes graph modeling to capture user relations within the recommendation system, and proposes a method to identify and mine hard negatives by partitioning similar users, thereby enhancing the effectiveness of training. Finally, Chapter 8 concludes the thesis, discusses its limitations, and suggests potential directions for future research.

Chapter 2

Related Work

This thesis primarily investigates the use of user intent and user behavior in dialogue and recommender systems. As outlined in Section , our research focuses on two main scenarios: distinguishing user intent in dialogue systems with limited training data, and leveraging graph-based methods and sequence compression techniques to integrate rich information for improved understanding of user behavior in recommender systems. Accordingly, this chapter provides an overview of related work as follows.

For the first part, user intent analysis in conversations, we review the literature on intent classification under different settings. For the second part—user behavior modeling for recommendation, we summarize advances in product search, information retrieval with graphs, graph-based negative sampling methods, and negative sampling strategies for sequential recommendation, respectively.

2.1 User Intent Analysis in Conversations

2.1.1 Supervised Intent Classification

User intent classification is a fundamental task in dialogue systems. Extensive research has focused on understanding user intent across diverse domains, including search engine queries [53] and medical inquiries [137]. Deep learning models, such as convolutional neural networks (CNNs) [126] and attention-based recurrent neural networks (RNNs) [91, 72], have become prevalent choices for intent classification. CNN-based methods generate sentence embeddings by aggregating the embeddings of neighboring words, while RNN-based approaches encode word embeddings sequentially to form sentence representations. Both paradigms have demonstrated robust performance in real-world applications [132].

Traditional intent classification approaches generally require a large amount of labeled data for each intent class to train effective classifiers. In contrast, zero-shot intent classification [107, 140] addresses the challenge that not all intent categories are present during training. This capability is particularly valuable in natural language understanding, as dialogue systems must continually adapt to emerging intents [72, 80, 126]. The objective of zero-shot intent classification is to leverage the knowledge and concepts learned from seen intents to accurately identify unknown intents. Early methods [35, 36, 130] explored the relationships between seen and unknown intents by utilizing external resources, such as manually defined attributes or label ontologies, which can be expensive to obtain. To mitigate this limitation, subsequent approaches [21, 60] have projected both utterances and intent labels into a shared embedding space, modeling their relationships within this unified representation. More recently, IntentCapsNet-ZS [119] has

extended capsule networks [95] to the zero-shot intent classification scenario by transferring prediction vectors from seen to unknown classes. However, ReCapsNet [74] has shown that IntentCapsNet-ZS faces difficulties in recognizing utterances from unknown intents under the generalized zero-shot classification setting. To address this, ReCapsNet proposes transferring transformation matrices from seen to unknown intents. In this thesis, we present ReCapsNet as a case study to demonstrate that integrating an unknown intent identifier into the generalized zero-shot classification framework can significantly improve both the recognition of unknown intents and the overall system performance.

2.1.2 Few-shot Intent Classification

Few-shot learning seeks to recognize novel categories using only a limited number of labeled examples, leveraging prior knowledge acquired from known categories [65]. In recent years, meta-learning has emerged as an effective approach for few-shot learning, where a meta-learner is trained on a large collection of tasks to develop the ability to rapidly adapt to new tasks involving unseen categories with minimal labeled data. Meta-learning methods are generally categorized into optimization-based approaches [96, 37, 79] and metric-based approaches, such as matching networks [109], prototypical networks [101], and relation networks [104]. While these methods have primarily focused on image classification, they are not specifically designed for text or user intent classification.

Several studies have attempted to adapt metric-based meta-learning algorithms for text classification tasks [93]. For instance, RobustTC [136] first clusters training tasks into groups and then learns an adaptive metric by combining the metrics

from different groups. The Induction Network [40] learns class prototypes through dynamic routing and computes the similarity between queries and prototypes using a trainable relation module. However, both methods represent each class with a single feature vector, which may be insufficient for capturing the diverse semantic features inherent in user intents.

2.1.3 Zero-shot Intent Classification

Zero-shot learning [62, 85] aims to recognize unseen classes, where no labeled samples are available. A common solution is to transfer knowledge learned from seen classes with abundant labeled data. This paradigm has been extensively studied in computer vision [6, 121] and natural language processing [107, 140].

Zero-shot intent classification is a critical task in natural language processing [53, 72, 80, 126], as new intents frequently emerge and cannot always be recognized accurately. Various methods have been proposed to address zero-shot intent classification. For example, [35, 36, 130] leverage external resources such as label ontologies or manually defined attributes to establish relationships between seen and unseen intent labels. However, acquiring such resources is often labor-intensive and time-consuming. Other approaches [21, 60] project utterances and intent labels into a shared semantic space and compute their similarities [21, 60]. Nevertheless, the diversity of user expressions can complicate the learning of effective projection functions, potentially impacting classification performance. More recently, [119] extended capsule networks for zero-shot intent classification by transferring prediction vectors from seen to unseen classes. Despite these advances, challenges remain, such as polysemy in word embeddings and limited generalization to unseen

intents in the generalized zero-shot intent classification setting.

2.1.4 New Intent Discovery (NID)

To better address real-world production requirements, a more practical framework known as New Intent Discovery (NID) has been proposed [139, 138]. NID leverages known labels while simultaneously uncovering novel intent categories, thereby expanding the set of supported intents. Building on this foundation, further research has sought to enhance NID methodologies. For example, Zhang et al. [146] introduced a contrastive loss to exploit self-supervisory signals in unlabeled data for improved clustering. Zhou et al. [147] developed a probabilistic framework for NID, treating intent assignments as latent variables. An et al. [4] proposed a robust pseudo-label training and source domain joint-training network to refine noisy pseudo labels and fully utilize prior knowledge.

Critical Analysis of NID Methods While the aforementioned NID approaches have demonstrated promising results, each methodology exhibits distinct advantages and limitations that warrant careful examination.

Clustering-based methods [99, 87, 17] offer computational efficiency and interpretability, as they directly group similar utterances without requiring extensive labeled data. However, these approaches often struggle with high-dimensional sparse representations and are sensitive to initialization and hyperparameter selection. The quality of discovered intents heavily depends on the choice of distance metrics and clustering algorithms, which may not capture the nuanced semantic relationships inherent in natural language.

Semi-supervised pre-training methods address some of these limitations by

leveraging both labeled and unlabeled data to learn more robust representations. These approaches can capture richer semantic information through pre-training on large corpora. However, they typically require substantial computational resources and may suffer from domain shift when the pre-training data distribution differs significantly from the target domain. Additionally, the learned representations may not be optimally discriminative for the specific task of intent discovery.

Methods that incorporate external knowledge [139, 146] can enhance semantic understanding by leveraging structured information such as knowledge graphs or ontologies. This external knowledge can provide valuable context for disambiguating similar intents and improving clustering quality. Nevertheless, the dependency on high-quality external resources limits their scalability and adaptability across different domains. The cost of acquiring and maintaining such knowledge bases can be prohibitive, particularly for specialized or rapidly evolving domains.

Recent LLM-based approaches offer a promising direction by combining the zero-shot reasoning capabilities of large models with the efficiency of lightweight encoders. These methods can leverage the rich semantic knowledge encoded in LLMs while maintaining practical deployment feasibility. However, challenges remain in ensuring the quality and consistency of LLM-generated pseudo-labels, as well as in designing effective knowledge distillation strategies that preserve the nuanced understanding of the teacher model.

2.1.5 Open-world Intent Classification

Most existing classification methods operate under the closed-world assumption, meaning no new classes are expected to appear during testing. However, the real

world is open and dynamic, and in many applications, AI agents must contend with previously unseen classes, making open-world learning and classification essential challenges.

There are two primary strategies for open-world classification. The first involves equipping classifiers with an additional confidence score to estimate whether a test sample belongs to a seen or unseen class. For example, cbsSVM [34] introduces a center-based similarity (CBS) learning strategy and employs SVM to construct 1-vs-rest CBS classifiers. MSP [49] utilizes the maximum softmax probability as a confidence score. Alternatively, DOC [100] replaces the softmax output layer with a 1-vs-rest layer containing a sigmoid function for each seen class, thereby reducing open space risk.

The second approach treats open-world classification as an outlier detection problem, leveraging anomaly detection techniques such as robust covariance estimators [94], one-class SVM [97], isolation forest [73], and local outlier factor (LOF) [13]. Robust covariance estimators assume the data follows a Gaussian mixture distribution and fit an elliptic envelope, identifying outliers as points distant from the fitted shape. One-class SVM defines a hyperplane that encloses positive samples as the decision boundary. Isolation forest constructs binary search trees to isolate samples, with outliers typically isolated earlier and closer to the root node. LOF is a density-based algorithm that compares the density of a point to its neighbors to determine abnormality; lower density indicates a higher likelihood of being an outlier. Additionally, to facilitate anomaly detection, some methods [70, 110] employ large margin loss functions to learn more discriminative feature representations.

2.2 User Behavior Modeling for Recommendation

2.2.1 Product Search

Latent space-based product search. In recent years, neural network-based models have become the dominant approach in product search research [44, 11, 122, 43, 81, 129]. LSE, proposed by Van Gysel et al. [45], is the first latent vector space-based search framework, mapping queries and products into a shared space. Building on this, Ai et al. [3] incorporated user preferences by jointly learning embeddings for users, products, and words through two tasks: language modeling and information retrieval. Subsequently, Ai et al. [1] considered user historical behaviors conditioned on the current query and introduced a zero attention vector to control the degree of personalization. To simultaneously capture both long- and short-term user preferences, Guo et al. [43] designed a dual attention-based network that models users' current search intentions alongside their long-term preferences. More recently, Transformer-based architectures have also been explored for product search [9, 10]. In this work, we adopt the widely used latent space-based product search framework as in ZAM [1] and propose to enhance product representations using graph convolution.

Graph-based product search. Early research leveraged relational information, such as social signals or user behavior traces, to improve search and ranking [18, 8, 7]. More recently, graph-based modeling of such information has gained traction [145, 14, 2, 75]. Zhang et al. [145] introduced GEPS (Graph Embedding-based ranking model for Product Search), which utilizes pre-training techniques to learn product and query embeddings, marking the first use of graphs in product search, though it does not account for user preferences. Bu et al. [14]

modeled product textual semantic relationships with hypergraphs to capture structural information. Ai et al. [2] proposed DREM (Dynamic Relation Embedding Model), which constructs a directed unified knowledge graph and jointly learns all embeddings via graph regularization; however, DREM is limited by its inability to select informative information and is susceptible to noise. In this work, we leverage information from a successive behavior graph to mitigate this issue. Liu et al. [75] presented GraphSRRL (Graph embedding based Structural Relationship Representation Learning model), which explicitly models user-query-product relationships and focuses on local relations. In our approach, we employ graph convolution with jump connections to aggregate high-order graph information.

Comparison: Latent-Space vs Graph-Based Approaches The literature on user behavior modeling for product search and recommendation reveals two primary technical paradigms: latent-space methods and graph-based methods. Each approach offers distinct advantages and faces unique challenges in capturing user preferences and item relationships.

Latent-Space Methods learn dense vector representations of users and items in a shared embedding space, where similarity is typically measured using inner products or cosine similarity. These methods, including matrix factorization and neural collaborative filtering, excel at capturing global patterns and can efficiently handle large-scale datasets through optimized linear algebra operations. They are particularly effective when user-item interactions are relatively dense and when the goal is to model general preference patterns. However, latent-space methods may struggle to capture complex, non-linear relationships and local structural patterns in user behavior. The learned representations can be difficult to interpret, and these

methods may not effectively leverage the rich relational information inherent in user interaction sequences.

Graph-Based Methods, in contrast, explicitly model the structural relationships between users and items as a graph, where nodes represent entities and edges encode interactions or similarities. Graph neural networks and related techniques can propagate information through the graph structure, enabling the capture of higher-order relationships and local neighborhood patterns. These methods are particularly powerful for modeling sequential dependencies and can naturally incorporate various types of side information through heterogeneous graph structures. Graph-based approaches often provide better interpretability, as the learned representations are grounded in explicit structural relationships.

Performance and Efficiency Trade-offs: In terms of computational efficiency, latent-space methods generally offer faster inference times due to simple vector operations, making them suitable for real-time recommendation scenarios. Graph-based methods, while potentially more expressive, often require more complex computations involving graph traversal and message passing, which can impact scalability. However, recent advances in graph sampling and mini-batch training have significantly improved the efficiency of graph-based approaches.

Data Requirements and Robustness: Latent-space methods typically require substantial amounts of interaction data to learn meaningful representations, and their performance can degrade significantly in cold-start scenarios. Graph-based methods can be more robust to data sparsity by leveraging structural information and can more effectively handle new users or items through graph-based inference. However, the quality of graph-based methods heavily depends on the graph construction strategy and the choice of graph neural network architecture.

In practice, the choice between these paradigms depends on the specific application requirements, including the nature of available data, computational constraints, interpretability needs, and the importance of capturing local versus global patterns in user behavior.

2.2.2 Information Retrieval with Graphs

Graph-based methods have been extensively studied in the field of sequential recommendation. These methods can be broadly divided into two categories: embedding-based methods and graph neural network (GNN)-based methods. Embedding-based approaches [26, 39, 69, 29] utilize network embedding techniques such as DeepWalk [88] and Node2Vec [42], learning structural information from graphs by leveraging the skip-gram model [78]. GNN-based methods [69, 83, 90, 117] aggregate information over graphs using GNNs [27, 59]. Our work is closely related to GCE-GNN (Global Context Enhanced Graph Neural Networks) [117], as both approaches utilize global information. However, while GCE-GNN emphasizes transitions between items and incorporates item order, our method does not consider product order and instead focuses on their co-occurrence relationships.

2.2.3 Graph-based Negative Sampling Methods

Most existing graph-based negative sampling methods are designed for collaborative filtering (CF)-based recommendation [54, 127] and graph contrastive learning [128, 148, 148, 120]. MixGCF [54] introduces a mix-up mechanism to efficiently inject information from positive samples into negative samples. HORACE [148] and STENCIL [148] investigate heterogeneous graph contrastive

learning and leverage global topology features, such as PageRank [61] and Laplacian vectors. However, these features are typically static, which can introduce risks in static hard negative sampling. ProGCL [120] addresses the challenge of false-negative discrimination in graph contrastive learning by fitting a two-component (true-false) beta mixture model to distinguish true negatives. RecNS [127] proposes the three-region principle to guide negative sampling for graph-based recommendation, suggesting that more negatives should be sampled from the intermediate region, with fewer from adjacent and distant regions. In our work, we construct a global weighted item transition graph to model sequential structural information for sequential recommendation (SR). To mitigate the risks associated with static hard sampling, we employ curriculum learning to dynamically adjust the maximum hardness of negative samples. Furthermore, to address the false negative issue [120], which is particularly pronounced in graph-based methods, we avoid sampling negatives from the "adjacent region" relative to the target item.

Efforts to develop informative negative sampling strategies for sequential recommendation (SR) can be broadly categorized into sampling-based methods [22, 41, 118, 56, 124, 113] and generation-based methods [114, 135, 115, 54]. ANCE [124] proposes sampling hard negatives globally using an asynchronously updated approximate nearest neighbor (ANN) index. CBNS [113] introduces cross-batch negative sampling, moving beyond simple in-batch sampling. SRNS [28] leverages the variance-based characteristics of false negatives and hard negatives to enhance the sampling strategy.

Chapter 3

Unknown Intent Detection

3.1 Motivation

Understanding user intent is crucial for developing conversational and dialogue systems. It is essential to accurately identify the intent behind a user utterance to better guide downstream decisions and policies. With the advent of conversational AI, dialogue systems are becoming central tools in many applications such as mobile apps, companion bots, virtual assistants and so on. Since user interests may change frequently over time, the AI agents may continuously see unknown (new) user intents. Manual annotation can hardly catch up with such rapid development, which motivates the problem of unknown intent detection that has recently attracted increasing interest from both academia and industry.

While there have been some pioneering works studying the open-world classification problem in natural language processing [34, 100], very few methods are designed for unknown intent detection. To our knowledge, the first work is by Lin et al. [70], in which the authors use large margin cosine loss (LMCL) to learn deep

discriminative features and then feed them to a density-based outlier detection algorithm to identify unknown intents. Although this method performs well on some benchmark datasets, it has two limitations. (1) In training, LMCL ignores the prior knowledge of class labels, while it has been shown that label correlations captured in the embedding space can improve prediction performance, especially in the zero-shot learning scenarios [85, 76]. (2) LMCL computes the cosine distance between embeddings in the feature space and trains with a softmax cross-entropy loss, making the embedding distribution of each class long and narrow [110], which may be less suitable for applying density-based anomaly detection algorithms to detect unknown intents.

In this chapter, we aim to address these limitations and propose a novel semantic-enhanced Gaussian mixture model (SEG) for unknown intent detection. In contrast to the softmax function, the Gaussian mixture model enforces embeddings to form ball-like dense clusters in the feature space, which may be more desirable for outlier detection, especially when using density-based outlier detection algorithms. Furthermore, we propose to inject the semantic information of class labels into the Gaussian mixture distribution by assigning the embeddings of class labels or descriptions to be the means of the Gaussians. This enables SEG to learn more class-concentrated embeddings that can benefit downstream outlier detection. We further use a large margin loss to make SEG learn more discriminative features and employ a density-based outlier detection algorithm Local Outlier Factor (LOF) [13] to detect unknown intents.

Identifying unknown intents is not enough for some application scenarios where it is important to know what exactly the new intents are, e.g., zero-shot intent classification. Current generalized zero-shot intent classification methods

[21, 60, 119, 74] attempt to classify test instances directly by making predictions in the pool of all the seen and unseen intents. However, their prediction performances are quite low, and they are still far from practical use. In this work, we propose to integrate SEG as an unknown intent identifier into the generalized zero-shot intent classification pipeline. The basic idea is that correctly identifying if the intent of an utterance is known or unknown will make the subsequent intent classification task much easier. We conduct a case study on a state-of-the-art zero-shot intent classification method ReCapsNet [74]. The results show that incorporating SEG successfully improves the performance of ReCapsNet by a large margin. It even pushes the performance to a practical level on the SNIPS dataset [24]. In sum, in this chapter, we explore the problem of unknown intent detection, and present the proposed method, a Smantic-Enhanced Gaussian mixture (SEG) model for unknown intent detection. As illustrated in Figure 4.1, it first learns a discriminative feature extractor with seen classes, then, in the test procedure, SEG classifies the seen classes with the trained classifier and detects new intents with a density-based outlier detection algorithm, namely LOF.

3.2 Problem Formulation

Unknown intent detection is expected to detect the existence of new intents. Specifically, given the training data $D^{tr} = (X^{tr}, Y^{tr})$ where $Y^{tr} \in C_{\text{seen}}$, a zero-shot classification system is required to predict labels $\hat{y}^{te}$ of test data based on the knowledge learned from the labeled data of seen classes.

3.3 Method

Feature Extraction Given an utterance $\boldsymbol{x} = \{\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_T\}$ with T words, where $\boldsymbol{w}_t \in \mathbb{R}^{d_w}$ is the embedding of the t-th word. Each word can be further encoded sequentially using a bidirectional LSTM (BiLSTM), i.e.,

$$\begin{aligned} \overrightarrow{\boldsymbol{h}}_t &= \text{LSTM}_{fw}(\boldsymbol{w}_t, \overrightarrow{\boldsymbol{h}}_{t-1}), \\ \overleftarrow{\boldsymbol{h}}_t &= \text{LSTM}_{bw}(\boldsymbol{w}_t, \overleftarrow{\boldsymbol{h}}_{t+1}), \end{aligned} \tag{3.1}$$

where $\overrightarrow{\boldsymbol{h}}_t, \overleftarrow{\boldsymbol{h}}_t \in \mathbb{R}^{d_h}$ are the hidden states of the word $\boldsymbol{w}_t$ by forward LSTM_{fw} and backward LSTM_{bw} respectively. The word $\boldsymbol{w}_t$ is encoded as the entire hidden state, which is represented by concatenating $\overrightarrow{\boldsymbol{h}}_t$ and $\overleftarrow{\boldsymbol{h}}_t$, i.e. $\boldsymbol{h}_t = [\overrightarrow{\boldsymbol{h}}_t; \overleftarrow{\boldsymbol{h}}_t]$, and the hidden state matrix of the utterance can be represented as $\boldsymbol{H} = [\boldsymbol{h}_1, \boldsymbol{h}_2, \ldots, \boldsymbol{h}_T] \in \mathbb{R}^{2d_h \times T}$. Furthermore, we use the self-attention mechanism to obtain the sentence embedding. Specifically,

$$\begin{aligned} \boldsymbol{a} &= \text{softmax}\left(\boldsymbol{W}_{s2}\text{tanh}(\boldsymbol{W}_{s1}\boldsymbol{H})\right), \\ \boldsymbol{z} &= \boldsymbol{W}\boldsymbol{H}\boldsymbol{a}, \end{aligned} \tag{3.2}$$

where $\boldsymbol{a} \in \mathbb{R}^T$ is the self-attention weight vector, $\boldsymbol{W}_{s1} \in \mathbb{R}^{d_a \times 2d_h}$ and $\boldsymbol{W}_{s2} \in \mathbb{R}^{1 \times D_a}$ are trainable parameters, $\boldsymbol{W} \in \mathbb{R}^{d_z \times 2d_h}$ is also trainable feedforward weight parameter, and $\boldsymbol{z} \in \mathbb{R}^{d_z}$ is the final representation of the utterance $\boldsymbol{x}$.

Semantic-Enhanced Large Margin Gaussian Mixture Loss The softmax cross-entropy loss is widely used in many machine learning problems. However, the embedding distribution of each class learned by the softmax cross-entropy loss tends to be long, narrow, and radiating from the center, with different classes

distributed next to each other closely [110]. Such embedding distribution may not be ideal for detecting new intent classes, as there might not be much space for new classes. Nevertheless, the Gaussian mixture loss can enforce each class to gather into a dense and small cluster, which may be more desirable for detecting new intents. Here, we design a semantic-enhanced large margin Gaussian mixture loss for embedding learning.

Large-Margin Cross-Entropy Loss. Given a K-way classification task, we assume the extracted feature vector (embedding) $\boldsymbol{z}$ of the training samples follows a Gaussian mixture distribution, where $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ are the mean and covariance of class k in the embedding space respectively and $p(k)$ is the prior probability of class k. The probability density function of $\boldsymbol{z}$ is given by

$$p(\boldsymbol{z}) = \sum_k \mathcal{N}(\boldsymbol{z}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) p(k), \tag{3.3}$$

where $\mathcal{N}(\boldsymbol{z}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ is the Gaussian distribution.

For the embedding $\boldsymbol{z}_i$ of any training sample $\boldsymbol{x}_i$, the posterior probability that $\boldsymbol{z}_i$ belongs to its class y_i can be expressed as

$$p(y_i|\boldsymbol{z}_i) = \frac{\mathcal{N}(\boldsymbol{z}_i; \boldsymbol{\mu}_{y_i}, \boldsymbol{\Sigma}_{y_i}) p(y_i)}{\sum_k \mathcal{N}(\boldsymbol{z}_i; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) p(k)}. \tag{3.4}$$

The cross-entropy loss of $\boldsymbol{z}_i$ between the true class label y_i and the inference $p(y_i|\boldsymbol{z}_i)$ can then be computed as:

$$\mathcal{L}_{ce,i} = -\log p(y_i|\boldsymbol{z}_i), \tag{3.5}$$

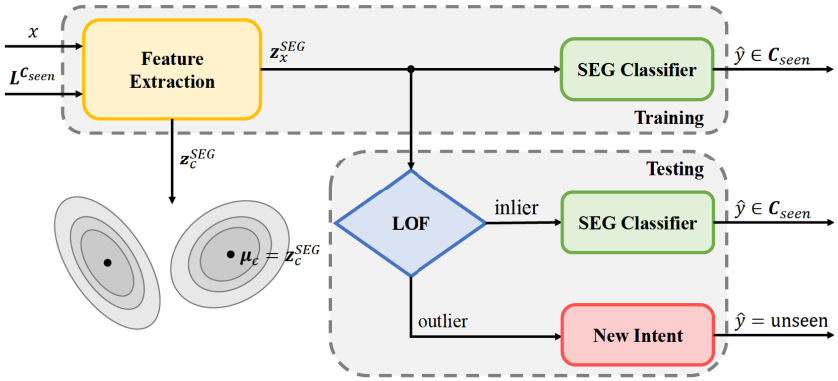


Figure 3.1: Illustration of the proposed framework for unknown intent classification. The backbone network is a self-attention Bi-LSTM encoder, which is trained by the proposed semantic-enhanced large margin Gaussian mixture loss (SEG classifier). In the testing phase, LOF is employed to detect outliers. The predicted outliers will be considered as unseen intent class instances, while the inliers will be classified by the SEG classifier.

and the total loss of N training samples is

$$\mathcal{L}_{ce} = \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}_{ce,i}. \tag{3.6}$$

Let d_k be the Mahalanobis distance between $\boldsymbol{z}_i$ and $\boldsymbol{\mu}_k$, i.e.,

$$d_k = (\boldsymbol{z}_i - \boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_k^{-1} (\boldsymbol{z}_i - \boldsymbol{\mu}_k)/2. \tag{3.7}$$

Then $\mathcal{L}_{ce,i}$ can be expressed as

$$\mathcal{L}_{ce,i} = -\log \frac{p(y_i)|\boldsymbol{\Sigma}_{y_i}|^{-\frac{1}{2}} e^{-d_{y_i}}}{\sum_k p(k)|\boldsymbol{\Sigma}_k|^{-\frac{1}{2}} e^{-d_k}}. \tag{3.8}$$

Consider a simplified case where $p(k)$ and $\boldsymbol{\Sigma}_k$ are identical for all classes. In this case, the model will give a correct prediction of $\boldsymbol{z}_i$ if the distance of $\boldsymbol{z}_i$ to its class mean $\boldsymbol{\mu}_{y_i}$ is less than or equal to its distance to any other class mean.

In general, large margin loss helps to improve classification performance. Here, we also introduce a classification margin $m \in [1, +\infty)$ into the cross-entropy loss, which then becomes:

$$\begin{aligned} \mathcal{L}_{ce}^{m} &= \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}_{ce,i}^{m}, \\ \mathcal{L}_{ce,i}^{m} &= -\log \frac{p(y_i)|\mathbf{\Sigma}_{y_i}|^{-\frac{1}{2}} e^{-m d_{y_i}}}{\sum_k p(k)|\mathbf{\Sigma}_k|^{-\frac{1}{2}} e^{-d_k}}. \end{aligned} \tag{3.9}$$

With the large margin loss, $\boldsymbol{z}_i$ is correctly classified only when its distance to class mean $\boldsymbol{\mu}_{y_i}$ is significantly less than (no more than $\frac{1}{m}$ of) its distance to any other class mean.

Semantic Enhancement via Class Description. This is one of the key features of our proposed method. We inject the semantic information of each class into the Gaussian mixture model by assigning the embedding learned from the text description d_k of class k to be the class centroid $\boldsymbol{\mu}_k$. The text description d_k can either be a single-word class name or a sentence or paragraph that describes the class. That is,

$$\boldsymbol{\mu}_k = \text{feature\_extract}(d_k), \tag{3.10}$$

where feature_extract$(\cdot)$ indicates the feature extraction module in Section 3.1.

Generation Loss. In addition to the cross-entropy loss, we want to maximize the observed likelihood of the embeddings with the Gaussian mixture distribution.

Specifically, we minimize the following negative logarithm likelihood,

$$\begin{aligned}\mathcal{L}_g &= -\sum_{i=1}^{N} \log \mathcal{N}(\boldsymbol{z}_i; \boldsymbol{\mu}_{y_i}, \boldsymbol{\Sigma}_{y_i}) \\ &= \frac{1}{2}\sum_{i=1}^{N} (\boldsymbol{z}_i - \boldsymbol{\mu}_{y_i})^\top \boldsymbol{\Sigma}_{y_i}^{-1} (\boldsymbol{z}_i - \boldsymbol{\mu}_{y_i}) + \text{const},\end{aligned} \tag{3.11}$$

where const means a constant number. As shown in Eq. (3.11), the generation loss $\mathcal{L}_g$ encourages the embedding $\boldsymbol{z}_i$ to be close to its class centroid $\boldsymbol{\mu}_{y_i}$, which facilitaes learning a more class-concentrated embedding distribution that may benefit the downstream outlier detection task. By combining the cross-entropy loss and the generation loss, the total objective function is:

$$\mathcal{L} = \mathcal{L}_{ce}^{m} + \lambda \mathcal{L}_g, \tag{3.12}$$

where λ is a trade-off parameter.

Outlier Detection By the above feature learning procedure, each utterance $\boldsymbol{x}$ can be encoded as an embedding $\boldsymbol{z}$. Then, the embedding $\boldsymbol{z}$ is fed to a well-known outlier detection algorithm LOF [13] to detect new or unknown intents (outliers). LOF is an unsupervised density-based anomaly detection method based on the following intuition. By comparing the local density of an object to those of its neighbors, it can identify regions of similar density. The objects with substantially lower density than their neighbors' are considered to be outliers.

LOF defines the local outlier factor of an object $\boldsymbol{z}$ as

$$\text{LOF}_k(\boldsymbol{z}) = \frac{1}{|N_k(\boldsymbol{z})|} \sum\nolimits_{\boldsymbol{o} \in N_k(\boldsymbol{z})} \frac{\text{lrd}_k(\boldsymbol{o})}{\text{lrd}_k(\boldsymbol{z})}, \tag{3.13}$$

where $N_k(\boldsymbol{z})$ denotes the set of k-nearest neighbors of $\boldsymbol{z}$, and "lrd" denotes the *local reachability density* which measures the local density around an object. The local reachability density is defined as the inverse of the average reachability distance between $\boldsymbol{z}$ and its neighbors, i.e.,

$$\text{lrd}_k(\boldsymbol{z}) = \frac{|N_k(\boldsymbol{z})|}{\sum_{\boldsymbol{o}\in N_k(\boldsymbol{z})} \text{reach-dist}_k(\boldsymbol{z}, \boldsymbol{o})}. \tag{3.14}$$

Here, the reachability distance reach-dist$_k(\boldsymbol{z}, \boldsymbol{o})$ is defined as

$$\text{reach-dist}_k(\boldsymbol{z}, \boldsymbol{o}) = \max\left\{\text{k-dist}(\boldsymbol{o}), d(\boldsymbol{z}, \boldsymbol{o})\right\}, \tag{3.15}$$

where k-dist$(\boldsymbol{o})$ denotes the distance of the object $\boldsymbol{o}$ to its k-th nearest neighbor, and $d(\boldsymbol{z}, \boldsymbol{o})$ is the distance between $\boldsymbol{z}$ and $\boldsymbol{o}$.

If the LOF factor of an utterance is much larger than 1, it has substantially lower local density than its neighbors', which means the utterance embedding is relatively distant from its neighbors. Hence, it can be inferred the utterance is likely to belong to an unknown intent class.

Overall Procedures Figure 3.1 illustrates the overall training and testing procedures of the proposed framework for unknown intent detection. The backbone network is a self-attention Bi-LSTM encoder. In the training phase, the encoder is trained by minimizing the semantic-enhanced large margin Gaussian mixture loss (SEG classifier) as in Eq. (3.12) on the training samples (seen intent class instances). In the testing phase, user utterances may come from both seen and unseen intent classes. Given an utterance, we first obtain its feature representation $\boldsymbol{z}$ with the trained encoder, then we use LOF to decide whether $\boldsymbol{z}$ is an outlier or

Dataset	SNIPS	ATIS	SMP-2018
Vocab Size	11642	938	3189
Avg. Length	9.05	11.37	4.87
# of Samples	13802	6371	2460
# of Classes	7	18	30

Table 3.1: Dataset statistics.

not. If $\boldsymbol{z}$ is an outlier, we take it as an instance of some new intent class. Otherwise, we classify $\boldsymbol{z}$ to one of the seen intent classes using the SEG classifier.

3.4 Experiments

In this section, we present experimental results on unknown intent detection. Formally, we train an unknown intent detection system with training data $D^{tr} = (X^{tr}, Y^{tr})$, where $Y^{tr} \in \{l_1, \cdots, l_K\} = C_{\text{seen}}$ (the set of seen intent classes). For test utterances belonging to seen intents, the unknown intent detection system should assign the correct intent labels. For utterances from unseen intents, the system is expected to recognize them as outliers.

3.4.1 Experimental Setup

Datasets and Baselines We evaluate our method SEG for unknown intent detection on 3 real task-oriented dialogue datasets: SNIPS [24], ATIS [48] and SMP-2018 [143]. SNIPS is an open-source single-turn English corpus, which contains 7 types of user intents across different domains. ATIS is also an English dataset, which contains 18 types of user intent in the airline travel domain. SMP-2018 is a Chinese dialogue corpus for user intent recognition, which contains 30

different types of user intents. The statistics of the datasets are summarized in Table 3.1.

We compare SEG with the following unknown intent detection methods.

- **Maximum Softmax Probability (MSP)** [49] considers the maximum softmax probability of a sample as the confidence score to measure the probability that it belongs to a seen intent. The smaller the confidence score is, the more likely it belongs to an unknown intent.

- **DOC** [100] builds m 1-vs-rest sigmoid classifiers for m seen classes respectively. The maximum probability is considered as the confidence of whether the sample belongs to the seen intent.

- **Softmax**. It can be considered as an ablation study of our method SEG, which uses softmax instead of Gaussian mixture distribution to learn discriminative features.

- **LMCL** [70] uses large margin cosine loss instead of Gaussian mixture distribution to learn discriminative embeddings.

- **SEG/o**. A variant of our method SEG. It does not inject the class semantic information into the Gaussian mixture model.

Detailed Setup We follow the setting in LCML [70] for unknown intent detection. Considering that some datasets may be unbalanced, we randomly select seen intents by a weighted random sampling over the entire intent set. The rest of the intents are regarded as unknown. We randomly select 30% samples of each intent to form the test set. The rest of each seen intent is added to the training set. We also follow

Dataset	SNIPS			ATIS			SMP-2018		
% of known intents	25%	50%	75%	25%	50%	75%	25%	50%	75%
MSP	**.5543**	.8060	**.8585**	**.6848**	.5158	.3853	.6132	.7089	.7716
DOC	.5462	.7962	.8564	**.7007**	.5073	.3659	.6095	.7197	.7642
Softmax	.5508	.8036	.8393	.6597	.6310	.5732	.5818	.6860	.7351
LMCL	.5489	.8041	.8458	.6763	**.6778**	**.6110**	.6059	.7094	.7580
SEG/o	.5440	**.8067**	.8474	.6768	.6699	.5918	**.6734**	**.7676**	**.8128**
SEG	**.5599**	**.8193**	**.8612**	.6410	**.6700**	**.6466**	**.6966**	**.7895**	**.8205**

Table 3.2: Macro F1-score of unknown intent detection with different proportion of seen classes. The top 2 results for each metric are marked in bold.

LMCL to use macro f1-score as the evaluation metric, which makes sense because the ATIS dataset is extremely unbalanced.

For SNIPS, ATIS and SMP-2018, we use 300-dim embeddings pre-trained on Fasttext, Glove, and Chinese-Word-Vectors respectively. For BiLSTM, we set the number of layers as 2 and the output dimension as 128. In the self-attention layer, we set the attention dimension d_a=10. After the self-attention layer, we project the feature vector to a d_z-dimension vector via a linear layer. We set d_z=12 for SNIPS and SMP-2018, and d_z=4 for ATIS. We report the average results over 10 runs. For the loss function, we set the margin $m = 1$ and the trade-off parameter $\lambda = 0.5$.

For MSP, we set the threshold as 0.5 following [70]. For DOC, we set the threshold as 0.5 as used in the original paper. During training of MSP and DOC, we clip the gradient norm to avoid gradient exploding. For LMCL, we follow the original paper to set the scaling factor $s = 30$ and the cosine margin $m = 0.35$. Softmax, LMCL, SEG/o and SEG all use LOF as the outlier detector, and we use the same set of parameters for LOF.

3.4.2 Result Analysis

From Table 3.2, it can be seen that our method SEG outperforms the baselines in most cases. Especially, on the most challenging dataset SMP-2018, SEG and SEG/o outperfom others by a large margin, demonstrating its high effectiveness. Moreover, we can make the following observations:

(1) SEG consistently achieves better results than SEG/o in most scenarios, demonstrating the effectiveness of the proposed semantic enhancement mechanism.

(2) SEG/o generally attains higher scores compared to Softmax and LMCL, with the difference being particularly notable on the more challenging SMP-2018 dataset. These findings highlight the advantage of using a Gaussian mixture model over Softmax and its variant LMCL for learning class-concentrated embeddings, which are better suited for integration with the LOF outlier detector.

(3) All methods perform well on SNIPS, likely due to its simplicity. On ATIS with only 25% seen classes, MSP and DOC show superior performance compared to other approaches. However, as the proportion of seen classes increases, their performance drops sharply. This decline can be attributed to the severe class imbalance in ATIS, where a single intent makes up 96% of the data. Under such conditions, DOC and MSP struggle to learn an effective supervised classifier, as the overwhelming dominance of one class hinders their ability to generalize.

3.5 Chapter Summary

This chapter addressed the critical problem of unknown intent detection in dialogue systems, a challenge that arises due to the dynamic and evolving nature of user intents in real-world conversational AI applications. Recognizing the limitations of

existing approaches, particularly those relying on softmax-based classifiers or large margin cosine loss (LMCL). This chapter introduced a novel Semantic-Enhanced Gaussian Mixture Model (SEG) as a robust solution for identifying previously unseen user intents.

The proposed methodology centers on modeling utterance embeddings with a Gaussian mixture distribution, wherein class semantic information is explicitly injected into the mean of its corresponding Gaussian component. This semantic enhancement encourages the formation of class-concentrated, ball-like clusters in the embedding space, which are more amenable to density-based outlier detection. To further improve discriminative power, a large margin loss is incorporated into the training objective. The overall framework employs a self-attention Bi-LSTM encoder for feature extraction, followed by the application of the Local Outlier Factor (LOF) algorithm to detect outliers, which are interpreted as utterances with unknown intents.

Extensive experiments were conducted on three real-world task-oriented multilingual dialogue datasets, including SNIPS, ATIS, and SMP-2018. The empirical results demonstrated that SEG consistently outperforms several strong baselines, including MSP, DOC, Softmax, and LMCL, particularly in scenarios with a high proportion of unseen classes or imbalanced data distributions. The ablation studies further validated the effectiveness of the semantic enhancement mechanism and the superiority of Gaussian mixture modeling over traditional approaches for this task.

Chapter 4

Zero-shot Intent Classification

4.1 Motivation

With the advent of conversational AI, more and more devices are equipped with goal-oriented spoken dialogue systems, e.g., the popular virtual personal assistants including Amazon Alexa, Apple Siri, Google Now, Microsoft Cortana and so on [20]. To improve business effectiveness and user satisfaction, accurately identifying the intents behind user utterances is indispensable. However, it is extremely challenging not only because user queries are sometimes short and expressed diversely, but also because it may continuously encounter unacquainted intents from various domains. Conventional intent classification methods [53, 72, 80, 126] typically train a supervised learning model on a large amount of data, which is not effective in detecting novel intents, since it is difficult to collect or label enough samples for each possible intent.

A few zero-shot learning approaches attempted to address the aforementioned challenges for intent classification and to make them capable of recognizing novel

intents whose instances are not present during training. Early works [35, 36, 130] usually require some external resources like label ontologies or manually defined attributes. However, the external resources are usually unavailable, as they require a lot of extra time and human labour to obtain. To implement zero-shot intent classification more easily and intelligently, recent works rely more on the word embeddings of intent labels, which can be easily pretrained on text corpus. Methods proposed by [21, 60] utilize neural networks to project data instances and intent labels to the same semantic space, and then measure their similarity. However, learning a good projection function for diverse user expressions is difficult, especially in some complex domains such as medical queries [137].

Unlike previous models, IntentCapsNet [119] employs capsule networks to extract high-level semantic features, and then transfers the prediction vectors for seen intents to unknown intents. Although IntentCapsNet has achieved fairly good performance in some zero-shot intent classification tasks, it still has two unaddressed limitations. (1) The self-attention mechanism of IntentCapsNet cannot handle polysemy, which weakens the representation capacity of semantic capsules. (2) The prediction vector construction method of IntentCapsNet can easily cause the model to have no generalization ability for detecting unknown intents in generalized zero-shot intent classification. This is certainly undesirable and inadequate, since in real dialogue systems, newly arrived utterances usually come from both seen and unknown intents.

In this chapter, we propose to reconstruct capsule networks for zero-shot intent classification (ReCapsNet-ZS), which effectively addresses the limitations of IntentCapsNet and adapts well to the generalized zero-shot intent classification task. ReCapsNet-ZS consists of two parts, as illustrated in Figure 4.1. First, it introduces

a dimensional attention mechanism to tackle polysemy, which helps to extract more representative and semantic features. Second, it computes the similarities between unknown and seen intents by utilizing the rich latent information hidden in the utterances, and then constructs the transformation matrices for unknown intents with the computed similarities and the trained transformation matrices of seen intents, which greatly improves the generalization ability to identify unknown intents. Empirical studies on two real task-oriented dialogue datasets in English and Chinese show that our proposed ReCapsNet-ZS performs much better than existing zero-shot intent classification methods.

4.2 Preliminary

4.2.1 Problem Formulation

Given the set of all intent labels $Y = Y^s \bigcup Y^u$, where $Y^s = \{y_1^s, y_2^s, \ldots, y_K^s\}$ and $Y^u = \{y_1^u, y_2^u, \ldots, y_L^u\}$ are the sets of seen and unknown intent labels respectively. There is no overlap between Y^s and Y^u, i.e., $Y^s \bigcap Y^u = \emptyset$, and K and L are respectively the numbers of seen and unknown intent labels. The embeddings of the seen labels and unknown labels are denoted by $E^s = \{\boldsymbol{e}_1^s, \boldsymbol{e}_2^s, \ldots, \boldsymbol{e}_K^s\}$ and $E^u = \{\boldsymbol{e}_1^u, \boldsymbol{e}_2^u, \ldots, \boldsymbol{e}_L^u\}$ respectively. Each embedding is a d-dimensional vector. For all the seen and unknown intent labels, their associated embeddings are available. The instance (utterance) sets with seen labels and unknown labels are denoted by $X^s = \{x_1^s, x_2^s, \ldots, x_{n_s}^s\}$ and $X^u = \{x_1^u, x_2^u, \ldots, x_{n_u}^u\}$, where n_s and n_u are the numbers of instances with seen and unknown labels respectively.

- **Standard Zero-shot Intent Classification.** The training set is $\{X^s, Y^s\}$,

X^u is not available in the training procedure. Its goal is to assign an unknown intent label $y \in Y^u$ to a given utterance x.

- **Generalized Zero-shot Intent Classification.** The training procedure is the same with zero-shot intent classification. Its goal is to assign an intent label $y \in Y^s \bigcup Y^u$ to a given utterance x.

4.2.2 Limitations of IntentCapsNet

IntentCapsNet [119] is the first method to extend capsule networks for zero-shot intent classification. It exploits self-attention mechanism to extract the semantic capsules, and then utilizes the vote vector and Euclidean distance based similarity to build the prediction vectors for unknown intents. Although IntentCapsNet has achieved fairly potential performance, there are some unaddressed key issues.
(1) Self-attention mechanism in IntentCapsNet cannot deal with the polysemy, thus weakening the representation capacity of semantic capsules.

In particular, a word is usually represented by a multi-dimensional embedding. As a word can have different meanings in different contexts, different dimensions of a word embedding will tend to represent different meanings. For example, the word "book" has two different meanings in the following two utterances: "Book a restaurant in Michigan for 4 people" and "Give 4 out of 6 points to this book". Some dimensions of the word embedding of "book" may tend to express the first meaning, while the remaining dimensions tend to represent the second one. Self-attention mechanism assigns a same attention coefficient for all dimensions of a word embedding. It cannot select the appropriate dimensions which can best indicate the specific meaning of a word in the given context, thus greatly limiting

the representation capacity of semantic capsules and further affecting the intent classification performance.

(2) The prediction vector construction method will very likely cause that IntentCapsNet loses the generalization ability for unknown intents in generalized zero-shot intent classification. The reasons are as follows.

Given an utterance x, its probability of belonging to a seen intent label k is P_k, which can be formulated as:

$$P_k = \| \sum_{r=1}^{R} c_{kr} \boldsymbol{p}_{k|r} \| = \| \sum_{r=1}^{R} \boldsymbol{g}_{k,r} \|, \tag{4.1}$$

where $\| \cdot \|$ is the 2-norm of a vector, R is the number of semantic capsules, $\boldsymbol{p}_{k|r}$ is the prediction vector of the r-th semantic capsule to the seen intent k, and c_{kr} is the weight that the r-th semantic capsule to the seen intent k, which is determined by dynamic routing algorithm. $\boldsymbol{g}_{k,r} = c_{kr}\boldsymbol{p}_{k|r}$ is the r-th vote vector for the seen intent k. From Eq.(4.1), we have:

$$P_k \leqslant \sum_{r=1}^{R} \|\boldsymbol{g}_{k,r}\|. \tag{4.2}$$

For the utterance x, its probability of belonging to an unknown intent label l is P_l, which can be formulated as:

$$P_l = \| \sum_{r=1}^{R} c_{lr} \boldsymbol{u}_{l|r} \| = \| \sum_{r=1}^{R} c_{lr} \sum_{k=1}^{K} q_{lk} \boldsymbol{g}_{k,r} \|, \tag{4.3}$$

where $\boldsymbol{u}_{l|r}$ is the prediction vector of the r-th semantic capsule to the unknown intent l, and c_{lr} is the weight that the r-th semantic capsule to the unknown intent l, which is determined by dynamic routing algorithm. $\boldsymbol{u}_{l|r} = \sum_{k=1}^{K} q_{lk}\boldsymbol{g}_{k,r}$, where

K is the number of seen intents, $\boldsymbol{g}_{k,r}$ is the r-th vote vector for the seen intent k, and q_{lk} is the similarity between an unknown intent $y_l^u \in Y^u$ and a seen intent $y_k^s \in Y^s$. $q_{lk} = \frac{exp(-d(\boldsymbol{e}_k^s, \boldsymbol{e}_l^u))}{\sum_{k=1}^{K} exp(-d(\boldsymbol{e}_k^s, \boldsymbol{e}_l^u))}$, where $\boldsymbol{e}_k^s$ and $\boldsymbol{e}_l^u$ are the embeddings of the seen and unknown intents, and $d(\boldsymbol{e}_k^s, \boldsymbol{e}_l^u)$ is the scalable squared Euclidean distance between $\boldsymbol{e}_k^s$ and $\boldsymbol{e}_l^u$. From the computation method of q_{lk} and c_{lr}, it is easy to get that $q_{lk} \in (0, 1)$, $c_{lr} \in (0, 1)$ and $\sum_{k=1}^{K} q_{lk} = 1$, $\sum_{r=1}^{R} c_{lr} = 1$. So we have:

$$P_l \leqslant \sum_{r=1}^{R} c_{lr} \sum_{k=1}^{K} q_{lk} \|\boldsymbol{g}_{k,r}\| \leqslant \|\boldsymbol{g}_{k,r}\|_{max}, \tag{4.4}$$

where $\|\boldsymbol{g}_{k,r}\|_{max}$ is the max value among all the $\|\boldsymbol{g}_{k,r}\|$, $r \in \{1, 2, \ldots, R\}$.

From Eq. (4.2) and Eq. (4.4), it can be seen that the upper bound of P_k is much larger than P_l, which indicates that for an utterance x, the probability that x is assigned to a seen intent will be larger than the probability that x is assigned to an unknown intent, i.e., the generalization ability of IntentCapsNet for unknown intents is limited.

4.3 Method

To tackle the limitations of previous work, we reconstruct capsule networks for zero-shot intent classification. The proposed approach consists of two parts. (1) Introducing the dimensional attention mechanism to capsule networks, which can extract more representative semantic capsules. (2) Constructing the transformation matrices for unseen intents, which can improve the generalization ability for detecting unseen intents.

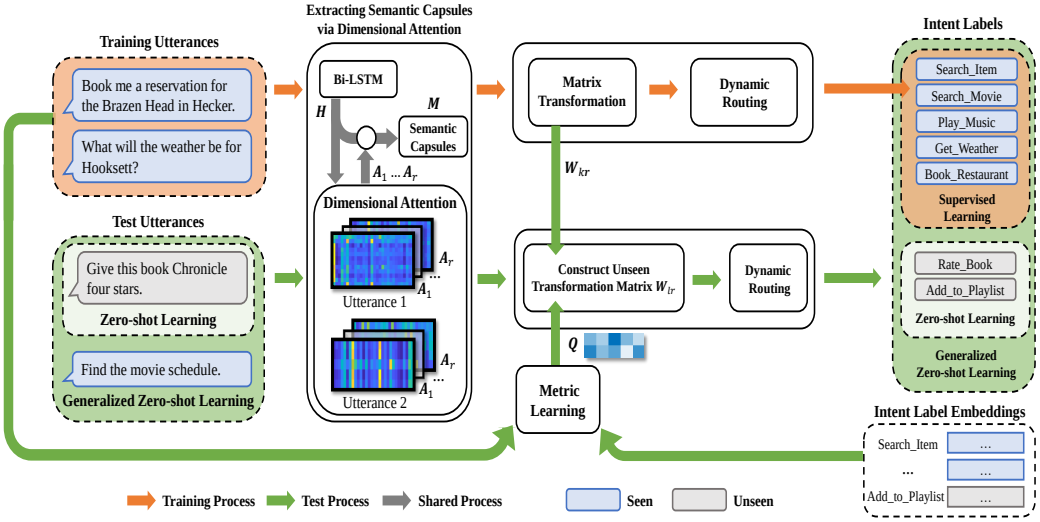


Figure 4.1: Illustration of ReCapsNet-ZS. **In training process**, the labeled training utterances are first encoded by Bi-LSTM, then a set of semantic capsules are extracted via dimensional attention mechanism, finally these semantic capsules are fed into capsule networks to train the intent classification model for seen intents. **In test process**, to predict unseen intents, a metric learning method is performed on training utterances and intent label embeddings to learn the similarities between unseen and seen intents, then these similarities and the trained transformation matrices of seen intents in capsule networks are used to construct the transformation matrices for unseen intents. When test utterances come, they are first encoded into semantic capsules via the trained Bi-LSTM and dimensional attention mechanism. Then there are two cases to predict their intents. (1) Zero-shot learning: only utterances from unseen intents participate in the test, i.e., the test utterances can only be classified into unseen intents. In this case, only transformation matrices of unseen intents are used for prediction. (2) Generalized zero-shot learning: test utterances come from both seen and unseen intents, so the test utterances can be classified into any one of the seen and unseen intents. In this case, transformation matrices of seen and unseen intents are all used for prediction.

4.3.1 Dimensional Attention Capsule Networks

Pre-processing Phase. Given an utterance $x = \{\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_T\}$ with T words, $\boldsymbol{w}_t \in \mathbb{R}^{d_w}$ is the word embedding of the t-th word, and can be pretrained by the skip-gram model [77]. Then a recurrent neural network such as a bidirectional LSTM [51] can be used to further encode each word embedding in the utterance sequentially, i.e.,

$$\begin{aligned} \overrightarrow{\boldsymbol{h}}_t &= \text{LSTM}_{fw}(\boldsymbol{w}_t, \overrightarrow{\boldsymbol{h}}_{t-1}), \\ \overleftarrow{\boldsymbol{h}}_t &= \text{LSTM}_{bw}(\boldsymbol{w}_t, \overleftarrow{\boldsymbol{h}}_{t+1}), \end{aligned} \tag{4.5}$$

where LSTM_{fw} and LSTM_{bw} denote the forward and backward LSTM respectively. $\overrightarrow{\boldsymbol{h}}_t \in \mathbb{R}^{d_h}$ and $\overleftarrow{\boldsymbol{h}}_t \in \mathbb{R}^{d_h}$ are the hidden states of the word $\boldsymbol{w}_t$ learned from LSTM_{fw} and LSTM_{bw} respectively. For each word $\boldsymbol{w}_t$, its hidden state $\boldsymbol{h}_t$ is obtained by concatenating $\overrightarrow{\boldsymbol{h}}_t$ and $\overleftarrow{\boldsymbol{h}}_t$, i.e., $\boldsymbol{h}_t = [\overrightarrow{\boldsymbol{h}}_t, \overleftarrow{\boldsymbol{h}}_t]$. So the whole hidden state matrix of each utterance is $\boldsymbol{H} = [\boldsymbol{h}_1, \boldsymbol{h}_2, \ldots, \boldsymbol{h}_T] \in \mathbb{R}^{2d_h \times T}$.

Extracting Semantic Capsules via Dimensional Attention. To better represent each utterance, we hope to extract the high-level semantic information from its words. Intuitively, some words have the same semantic information, so a semantic feature can be aggregated by a set of words. The key problem is how to learn the importance weight of each word for a semantic feature. IntentCapsNet [119] utilizes the self-attention mechanism to generate the semantic features (capsules) for each utterance. However, self-attention mechanism cannot solve the polysemy issue. Inspired by the success of dimensional attention mechanism to alleviate the polysemy for sentence embedding [98], we introduce the dimensional attention mechanism to extract the semantic capsules.

Assume there are R semantic features for each utterance, then a dimensional attention importance weight matrix $\boldsymbol{A}_r \in \mathbb{R}^{2d_h \times T}$ between T words and the r-th semantic feature is formulated as:

$$\boldsymbol{A}_r = \text{softmax}\left(\boldsymbol{F}_2 \, \text{ReLU}\left(\boldsymbol{F}_1 \boldsymbol{H}\right)\right), \tag{4.6}$$

where $\boldsymbol{F}_1 \in \mathbb{R}^{d_a \times 2d_h}$ and $\boldsymbol{F}_2 \in \mathbb{R}^{2d_h \times d_a}$ are the parameters of dimensional attention mechanism, and $\boldsymbol{A}_r(i, j)$ (the i-th row and the j-th column element of $\boldsymbol{A}_r$) means the importance of the j-th word to the r-th semantic feature on the i-th dimension. Compared with self-attention, dimensional attention mechanism can select the appropriate dimensions that can best express the specific meaning of a word in the given context.

After obtaining $\boldsymbol{A}_r$, the r-th semantic feature $\boldsymbol{m}_r \in \mathbb{R}^{2d_h}$ can be computed by:

$$\boldsymbol{m}_r = \sum\nolimits_{row} \left(\boldsymbol{A}_r \odot \boldsymbol{H}\right), \tag{4.7}$$

where $\odot$ is element multiplication, and $\sum_{row}$ is an operator to sum the elements in each row. The whole semantic features for each utterance is $\boldsymbol{M} = [\boldsymbol{m}_1, \boldsymbol{m}_2, \ldots, \boldsymbol{m}_R] \in \mathbb{R}^{2d_h \times R}$.

Capsule Networks with Improved Max-margin Loss. The semantic features for each utterance can be fed into capsule networks for learning its intent. Specifically, we first encode each semantic feature $\boldsymbol{m}_r$ in an utterance with respect to each intent:

$$\boldsymbol{p}_{k|r} = \boldsymbol{W}_{kr} \boldsymbol{m}_r, \tag{4.8}$$

where $\boldsymbol{p}_{k|r} \in \mathbb{R}^{d_p}$ is the prediction vector of the r-th semantic feature to the k-th

intent, and $\boldsymbol{W}_{kr} \in \mathbb{R}^{d_p \times 2d_h}$ is the associated transformation matrix.

The number of seen intents is K, so in the training process there are K output capsules. The k-th output capsule $\boldsymbol{o}_k$ is the weighted sum of all the prediction vectors $\boldsymbol{p}_{k|r}$ ($r \in \{1, \ldots, R\}$),

$$\boldsymbol{o}_k = \sum_{r=1}^{R} c_{kr} \boldsymbol{p}_{k|r}, \tag{4.9}$$

where c_{kr} is the coupling coefficient which can reflect the contribution degree of the r-th semantic feature to the k-th intent, and it can be computed by the dynamic routing algorithm.

To make the length of the output capsule represent the probability that each intent label exists, a squashing function squash$(\cdot)$ is applied to $\boldsymbol{o}_k$, then the final output capsule of the k-th intent is:

$$\boldsymbol{v}_k = \text{squash}(\boldsymbol{o}_k) = \frac{\|\boldsymbol{o}_k\|^2}{1 + \|\boldsymbol{o}_k\|^2} \frac{\boldsymbol{o}_k}{\|\boldsymbol{o}_k\|}. \tag{4.10}$$

The computation procedure of $\boldsymbol{v}_k$ is shown in Algorithm 1, and $\boldsymbol{p}_{k|r} \cdot \boldsymbol{v}_k$ denotes the inner product between $\boldsymbol{p}_{k|r}$ and $\boldsymbol{v}_k$.

For training the dimensional attention capsule networks, we propose an improved max-margin loss function which consists of two parts.

The first part is the max-margin loss on each labeled utterance, which is the original loss function of capsule networks and formulated as:

$$\begin{aligned} L_1 = \sum_{k=1}^{K} \{ & y_k \max(0, m^+ - \|\boldsymbol{v}_k\|)^2 \\ & + \lambda(1 - y_k) \max(0, \|\boldsymbol{v}_k\| - m^-)^2 \}, \end{aligned} \tag{4.11}$$

Algorithm 1: Dynamic Routing Algorithm

Input: Prediction vectors $\boldsymbol{p}_{k|r}$, number of routing iterations n_{route}
Output: Output capsules $\boldsymbol{v}_k$

for *all semantic capsules* r *and intent capsules* k **do**
  $b_{kr} \leftarrow 0$;
end
for $i = 1$ **to** n_{route} **do**
  for *all semantic capsules* r **do**
    $\boldsymbol{c}_r \leftarrow \text{softmax}(\boldsymbol{b}_r)$;
  end
  for *all intent capsules* k **do**
    $\boldsymbol{o}_k \leftarrow \sum_{r=1}^{R} c_{kr} \boldsymbol{p}_{k|r}$;
    $\boldsymbol{v}_k \leftarrow \text{squash}(\boldsymbol{o}_k)$;
  end
  for *all semantic capsules* r *and intent capsules* k **do**
    $b_{kr} \leftarrow b_{kr} + \boldsymbol{p}_{k|r} \cdot \boldsymbol{v}_k$;
  end
end
return $\boldsymbol{v}_k$

where $\boldsymbol{y} = [y_1, y_2, \ldots, y_K]^\top$ is a one-hot vector which indicates the ground truth of each labeled utterance. λ is a down-weighting parameter, m^+ and m^- are the margins.

The second part is to guarantee the diversity of the semantic capsules, i.e., different semantic capsules tend to be generated by different words in an utterance. As the importance of each word to the r-th semantic capsule can be expressed by the average value of each column of the dimensional attention importance weight matrix $\boldsymbol{A}_r$, i.e.,

$$\boldsymbol{s}_r = \frac{1}{2d_h} \sum\nolimits_{column} \boldsymbol{A}_r, \tag{4.12}$$

where $\boldsymbol{s}_r \in \mathbb{R}^{1 \times T}$, $\sum_{column}$ is an operator to sum the elements in each column. Then the importance of each word for all the R semantic capsules can be denoted

as $\boldsymbol{S} = [\boldsymbol{s}_1^\top, \boldsymbol{s}_2^\top, \ldots, \boldsymbol{s}_R^\top] \in \mathbb{R}^{T \times R}$. To ensure the diversity of the semantic capsules, a natural idea is to constrain $\boldsymbol{S}$ to be orthogonal. Formally, the objective function is:

$$L_2 = \|\boldsymbol{S}^\top \boldsymbol{S} - \boldsymbol{I}\|_F^2, \tag{4.13}$$

where $\|\cdot\|_F$ is the Frobenius norm of a matrix.

By combining Eq.(4.11) and Eq.(4.13), the improved max-margin loss function of the dimensional attention capsule networks is:

$$L_{total} = L_1 + \beta L_2, \tag{4.14}$$

where β is a trade-off parameter. By minimizing L_{total} with the gradient descent method, all model parameters like $\boldsymbol{F}_1$, $\boldsymbol{F}_2$ and $\boldsymbol{W}_{kr}$ can be obtained.

4.3.2 Working with Zero-shot Intent Classification

To better achieve zero-shot intent classification task with capsule networks, two key problems need to be solved. (1) How to find the relationship between unseen intents and seen intents. (2) How to make predictions for unseen intents.

To tackle the first problem, we propose to use Mahalanobis distance based metric learning method to measure the relationship between unseen and seen intents. Specifically, given an unseen intent l and a seen intent k, we have:

$$d_M(\boldsymbol{e}_l^u, \boldsymbol{e}_k^s) = (\boldsymbol{e}_l^u - \boldsymbol{e}_k^s)^\top \boldsymbol{\Omega}^{-1} (\boldsymbol{e}_l^u - \boldsymbol{e}_k^s), \tag{4.15}$$

where d_M denotes the squared Mahalanobis distance between two intent embeddings, and $\boldsymbol{\Omega}$ is a learned covariance matrix which models the dimension correlation

of intent embeddings. Note that IntentCapsNet [119] also tries to use Eq.(4.15) to get the relationship between unseen and seen intents, but it ignores the dimension correlation and just simply sets $\mathbf{\Omega} = \sigma^2 \boldsymbol{I}$, where σ is a hyper-parameter for scaling. Obviously, it is actually a scalable squared Euclidean distance.

As the number of intents is limited, just using the intent embeddings to compute the covariance matrix $\mathbf{\Omega}$ is difficult to obtain a reasonable result. Fortunately, the word embeddings of the utterances and the intent embeddings are both pre-trained by the same skip-gram model, i.e., they come from the same semantic space, so we can compute the covariance matrix $\mathbf{\Omega}$ by utilizing the labeled utterances in the training set. Inspired by the work in [133], we use the following objective function to achieve the Mahalanobis distance based metric learning:

$$\begin{aligned} &\max_{\mathbf{\Omega}} \min_{(i,j)\in\mathcal{D}} d_M(\boldsymbol{z}_i^s, \boldsymbol{z}_j^s), \\ &s.t. \sum_{(i,j)\in\mathcal{S}} d_M(\boldsymbol{z}_i^s, \boldsymbol{z}_j^s) \leqslant 1, \end{aligned} \tag{4.16}$$

where $\mathcal{D}$ and $\mathcal{S}$ respectively denote the pair sets in which utterances belong to different classes and the same class. For an utterance i, $\boldsymbol{z}_i^s$ denotes the average sum of all its word embeddings. As [133] stated, optimizing Eq.(4.16) w.r.t $\mathbf{\Omega}$ is equivalent to solve an efficient eigenvalue optimization problem. Then substituting $\mathbf{\Omega}$ into Eq.(4.15), we can get the relationship between unseen and seen intents. Furthermore, we use the following function to compute the similarity between unseen and seen intents: $q_{lk} = \exp\left(-d_M(\boldsymbol{e}_l^u, \boldsymbol{e}_k^s)\right)$.

Intuitively, if the similarity between an unseen intent and a seen intent is closer, their corresponding transformation matrices will be more analogous. Based on this observation, to solve the second problem, we propose to complement the

transformation matrices of unseen intents and then make predictions for unseen intents, by using the transformation matrices of seen intents in dimensional capsule networks and the similarities between unseen and seen intents. Specifically, given the similarity matrix between unseen and seen intents $\boldsymbol{Q} \in \mathbb{R}^{L \times K}$, for an unseen intent l, the transformation matrix of the r-th semantic capsule to the l-th intent $\boldsymbol{W}_{lr}$ can be constructed by:

$$\boldsymbol{W}_{lr} = \sum_{k=1}^{K} q_{lk} \boldsymbol{W}_{kr}, \tag{4.17}$$

where q_{lk} is the l-th row and the k-th column element of $\boldsymbol{Q}$, $\boldsymbol{W}_{kr}$ is the transformation matrix for the r-th semantic capsule to the k-th seen intent. By using Eq.(4.17), the transformation matrices for unseen intents can be complemented. When a test utterance arrives, it can be directly fed into the already trained dimensional attention capsule networks to predict its intent.

Compared with IntentCapsNet, our strategy for zero-shot classification can deal with both zero-shot intent classification and generalized zero-shot intent classification task well.

4.4 Experiments

4.4.1 Experimental Setup

Datasets We evaluate our model on two real task-oriented dialogue datasets in different languages. The detailed statistics are shown in Table 4.1.

SNIPS-NLU. Following [119], we use the SNIPS-NLU (SNIPS Natural Language Understanding) benchmark dataset [24]. SNIPS-NLU is an open-source single-turn

Dataset	SNIPS-NLU	SMP-2018
Vocab Size	11641	2682
Number of Samples	13802	2460
Average Sentence Length	9.05	4.86
Number of Seen Intents	5	24
Number of Unseen Intents	2	6

Table 4.1: Dataset statistics.

Dataset	d_w	d_h	d_a	d_p	R	n_{route}
SNIPS-NLU	300	16	10	10	3	3
SMP-2018	300	32	30	10	8	3

Table 4.2: The network structure hyper-parameters.

English corpus which contains crowdsourced queries evenly distributed in 7 intents.

SMP-2018. It is a real Chinese dialogue corpus released in SMP 2018 (The China National Conference on Social Media Processing) for the Chinese user intent classification task [143]. The data is provided by the iFLYTEK Corporation and can be divided into two parts: chit-chat dialogues and task-oriented dialogues. Here we only use the task-oriented part of the dataset.

Dataset Splitting. In zero-shot intent classification task, we take all the samples of seen intents as the training set, and all the samples of unseen intents as the test set. In generalized zero-shot intent classification task, we randomly take 70% samples of each seen intent as the training set, and the remaining 30% samples of each seen intent and all the samples of unseen intents as the test set.

Baselines We compare ReCapsNet-ZS with the state-of-the-art zero-shot learning methods: DeViSE [38], CMT [102], CDSSM [21], Zero-shot DNN [60] and IntentCapsNet [119]. For DeViSE and CMT, to make them more applicable

to our task, we use a multi-head self-attention Bi-LSTM model to encode the utterances and then feed the final hidden states to their zero-shot learning strategies. In addition, we also perform the ablation test to study the contribution of each proposed module. "ReCapsNet-ZS-Dim" means that the model only uses the dimensional attention mechanism, and "ReCapsNet-ZS-TM" means that the model only uses the proposed transformation matrix construction method.

Implement Details **Settings.** For SNIPS-NLU, we use 300 dim embeddings pre-trained on English Wikipedia [12]. For SMP-2018, we use 300 dim Chinese word embeddings pre-trained by [68]. The main network structure hyperparameters are shown in Table 4.2. To avoid overfitting, we apply dropout with 0.5 dropout rate on the input of the attention layer. In addition, we consistently set $\lambda = 0.5$, $m^+ = 0.9$, $m^- = 0.1$, $\beta = 0.001$ in the loss function, and use an Adam optimizer [58] with 0.01 initial learning rate to minimize it.

Evaluation Metrics. We adopt two widely used metrics: accuracy (Acc) and micro-average F1-measure (F1) to evaluate the performance. Both the metrics use the average value weighted by their support on per class, where the support means the sample ratio of the corresponding class.

4.4.2 Result Analysis

Zero-shot Intent Classification. Table 4.3 summarizes the average results over 10 runs, where the top 2 results are highlighted in bold. The results of the baselines for SNIPS-NLU are taken from [119]. It can be seen that (1) ReCapsNet-ZS outperforms the compared baselines, which shows the superiority of our proposed model for tackling zero-shot intent classification. (2) ReCapsNet-ZS performs

Method	SNIPS-NLU		SMP-2018	
	Acc	F1	Acc	F1
DeViSE	.7447	.7446	**.5456**	.3875
CMT	.7396	.7206	.4452	.4245
CDSSM	.7588	.7580	.4308	.3765
Zero-shot DNN	.7165	.7116	.4615	.3897
IntentCapsNet	.7752	.7750	.4864	.4227
ReCapsNet-ZS-Dim	**.7868**	**.7859**	.5005	.4501
ReCapsNet-ZS-TM	.7860	.7837	.5315	**.4630**
ReCapsNet-ZS	**.7996**	**.7980**	**.5418**	**.4769**

Table 4.3: Results of zero-shot intent classification.

better than either ReCapsNet-ZS-Dim or ReCapsNet-ZS-TM, which validates the effectiveness of the proposed transformation matrix construction method for predicting unseen intents and the dimensional attention mechanism for capturing more representative semantic capsules.

Generalized Zero-shot Intent Classification. Table 4.4 shows the average results over 10 runs, where the top 2 results are highlighted in bold. It can be seen that (1) The results for unseen intents are much lower than those in zero-shot intent classification, this is not surprising because the seen intent labels are also included into the search space for predicting the utterances from unseen intents. (2) ReCapsNet-ZS sometimes performs a little worse than the baselines for detecting seen intents, because some baselines seem more inclined to classify all the test utterances into seen intents, which also explains why they perform so worse for detecting unseen intents. (3) ReCapsNet-ZS and ReCapsNet-ZS-TM perform much better than other methods for detecting unseen intents, whereas IntentCapsNet and ReCapsNet-ZS only have 0% Acc and F1, which verifies that the proposed transformation matrix construction method has more powerful generalization ability

Method	SNIP-NLU						SMP-2018					
	Seen		Unseen		Overall		Seen		Unseen		Overall	
	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1
DeViSE	.9481	.6536	.0211	.0398	.4215	.3049	.8040	.6740	.0270	.0310	.5030	.4250
CMT	**.9755**	.6648	.0397	.0704	.4438	.3271	.8314	.7221	.0798	.1069	.5398	.4834
CDSSM	.9549	**.7033**	.0111	.0218	.4234	.3194	.6653	.5540	.1436	.1200	.4864	.4052
Zero-shot DNN	.9432	.6679	.0682	.1041	.4488	.3493	.7323	.6116	.0590	.0869	.5013	.4316
IntentCapsNet	.9741	.6517	.0000	.0000	.4200	.2810	**.8850**	.7281	.0000	.0000	.5375	.4423
ReCapsNet-ZS-Dim	**.9743**	.6580	.0000	.0000	.4201	.2837	**.8952**	.7360	.0000	.0000	.5437	.4470
ReCapsNet-ZS-TM	.9663	**.6845**	**.0981**	**.1534**	**.4724**	**.3824**	.7863	**.7390**	**.1896**	**.1537**	**.5521**	**.5092**
ReCapsNet-ZS	.9664	.6743	**.1121**	**.1764**	**.4805**	**.3911**	.8230	**.7450**	**.1720**	**.1526**	**.5674**	**.5124**

Table 4.4: Results of generalized zero-shot intent classification. "Seen", "Unseen" and "Overall" respectively denote the performance for the utterances from seen intents, unseen intents, and both seen and unseen intents.

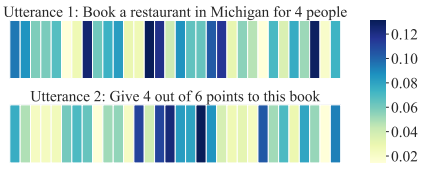


Figure 4.2: Dimensional attention comparison of the same word "book" in different contexts.

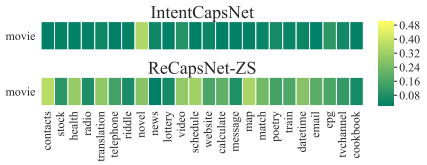


Figure 4.3: Similarity comparison of IntentCapsNet and ReCapsNet-ZS.

for detecting unseen intents. (4) For the overall results, ReCapsNet-ZS always performs the best, which further demonstrates the superiority of ReCapsNet-ZS on dealing with the generalized zero-shot intent classification task.

Visualization **Dimensional Attention Mechanism.** Figure 4.2 visualizes the attention values on each dimension of the same word in different utterances (contexts) by the heatmaps. Here we take the word "book" from SNIPS-NLU as example, the corresponding original utterances are "Book a restaurant in Michigan for 4 people" and "Give 4 out of 6 points to this book". It can be seen that these two dimensional attentions for the same word are totally different due to their different

meanings in different contexts. In addition, for a word in an utterance, the attention scores on different dimensions are also different. This indicates that dimensional attention mechanism can encode more information than traditional attention, which can be used to capture the difference of the same word in different contexts and then alleviate the polysemy.

Metric Learning based Similarity. Figure 4.3 visualizes the similarities between unseen and seen intents for IntentCapsNet and ReCapsNet-ZS. Due to space limit, we only show the similarities of one unseen intent "movie" with seen intents. It can be seen that IntentCapsNet can discover only few connections between unseen and seen intents. In contrast, ReCapsNet-ZS can find more connections between unseen and seen intents. Though the similarity computed by ReCapsNet-ZS may be mixed with few noises (e.g., "contacts" and "map"), it captures more positive latent relations for unseen intents (e.g., "novel", "video", "schedule", and "datetime"), which can bring greater benefits for detecting unseen intents.

4.5 A Case Study: Unknown Intent Identifier Integration for Generalized Zero-shot Intent Classification

Previous attempts try to tackle the challenge of zero-shot intent classification from three directions. (1) What prior knowledge is more supportive, such as morphology (character-level embedding), class descriptions, and knowledge-based entity attributes [35, 36, 21, 60]. (2) How to better utilize these prior knowledge to extract more informative semantic representations, such as data augmentation

and hierarchical representations learned by capsule networks [119]. (3) With the extracted semantic features, how to design a better zero-shot learning strategy, such as reconstructing weight matrix for unknown intents through relation learning [74].

4.5.1 Integrate SEG into the ReCapsNet

As shown in Figure 4.4, a typical generalized zero-shot classification framework can be abstracted into two layers, the encoder layer and the zero-shot classifier layer. In the encoder layer, a user utterance x in the text format needs to be first mapped to the semantic representation $\boldsymbol{z}_x^{ZS}$. In addition, it is common to encode class information as $\boldsymbol{S}$ for better semantic learning or knowledge transfer. In order to learn better semantic representation, prior knowledge is usually incorporated at this stage. Then, the learned representation will be fed to the zero-shot classifier layer. Various zero-shot classification strategies have been proposed to transfer knowledge to new categories. Finally, the system outputs the prediction $\hat{y}^{te} \in \{C_{\text{seen}}, C_{\text{unseen}}\}$ for the utterance x.

We integrate SEG into the pipeline between the encoder layer and the classifier layer as shown in Figure 4.5. With the semantic feature $\boldsymbol{z}_x$, we predict if the utterance x is an outlier via:

$$p(g|\boldsymbol{z}_x), g \in \{\text{“seen”}, \text{“unseen”}\}. \tag{4.18}$$

For the case $g =$ “seen”, the intent of the utterance is considered to be a seen one. We then predict the intent by $p(y|\boldsymbol{z}_x, y \in C_{\text{seen}}, X^{tr}, \boldsymbol{\theta})$ where $\boldsymbol{\theta}$ denotes the parameters of the original framework. Otherwise, the intent of the utterance is considered to be unseen, and we predict it via $p(y|\boldsymbol{z}_x, y \in C_{\text{unseen}}, X^{tr}, \boldsymbol{\theta})$.

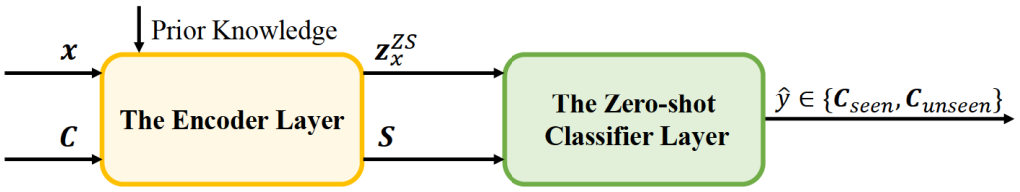


Figure 4.4: A typical generalized zero-shot intent classification pipeline.

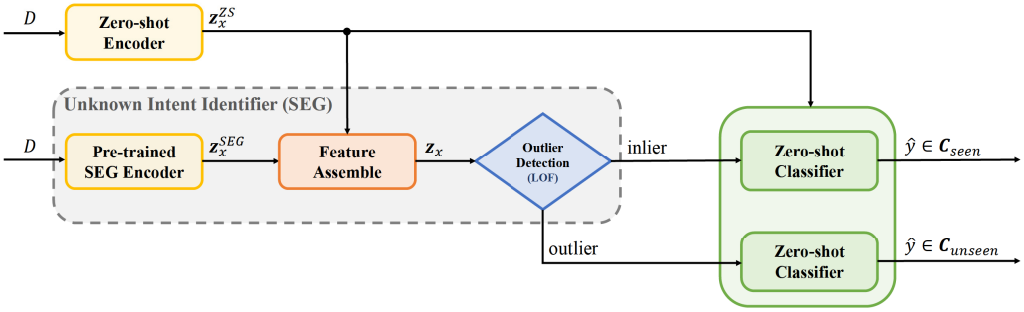


Figure 4.5: Integration of the new intent identifier (SEG) into the generalized zero-shot intent classification pipeline.

Feature Assemble We adopt two ways "Separate" and "Combine" to assemble features for the following outlier detection task.

- **Separate (Sep).** We directly feed the output of the pre-trained SEG encoder z_x^{SEG} to LOF for outlier detection, i.e.,

$$z_x = z_x^{SEG}. \tag{4.19}$$

- **Combine.** To take advantage of the original model, we first obtain the original semantic feature representation z_x^{ZS} and define a transform function f. Then, $f(z_x^{ZS})$ is concatenated with the pre-trained features by SEG, z_x^{SEG}, to make a combined feature representation:

$$z_x = [z_x^{SEG} || f(z_x^{ZS})]. \tag{4.20}$$

ReCapsNet As illustrated in Section 3.5, ReCapsNet demonstrates state-of-the-art performance in generalized zero-shot intent classification. In this section, we conduct a case study on integrating the new intent identifier into ReCapsNet.

Here we illustrate the framework of ReCapsNet as the proposed generalized zero-shot intent classification pipeline, as shown in Figure 4.6. In the encoder layer, each utterance x is encoded with R semantic capsules $[\boldsymbol{m}_1, \boldsymbol{m}_2, ..., \boldsymbol{m}_R]$ as the representations in R different semantic spaces. In addition, the training set $\boldsymbol{D}^{tr}$ and class labels $\boldsymbol{L}$ are encoded as $\boldsymbol{S}^{tr}$ and $\boldsymbol{S}^{C}$, respectively. In the zero-shot classifier layer, $\boldsymbol{z}_{\boldsymbol{x}}^{ZS}$ is fed to a capsule network to make prediction. Each seen class k has R transformation matrices $\{\boldsymbol{W}_{kr}\}_{r=1}^{R}$. In the testing phase, ReCapsNet reconstructs the r-th transformation matrix for each unseen class l as $\boldsymbol{W}_{lr} = \sum_k q_{lk}\boldsymbol{W}_{kr}$, where q_{lk} is the relation between unseen class l and seen class k learned from $(\boldsymbol{S}^{tr}, \boldsymbol{Y}^{tr})$ and $\boldsymbol{S}^{C}$ by metric learning.

For the variant "Combine", to exploit the property that each utterance is variously represented in different semantic spaces as discussed in [74], we define the semantic feature representation of ReCapsNet as

$$f(\boldsymbol{z}_x^{ZS}) = [\|\boldsymbol{m}_1\|_2, \|\boldsymbol{m}_2\|_2, \cdots, \|\boldsymbol{m}_R\|_2]. \tag{4.21}$$

4.5.2 Results

Experimental Setup We integrate SEG into the ReCapsNet pipeline with both "Sep" and "Combine" variants and test the performance of generalized zero-shot classification.

Following the settings of generalized zero-shot classification in [74], we test

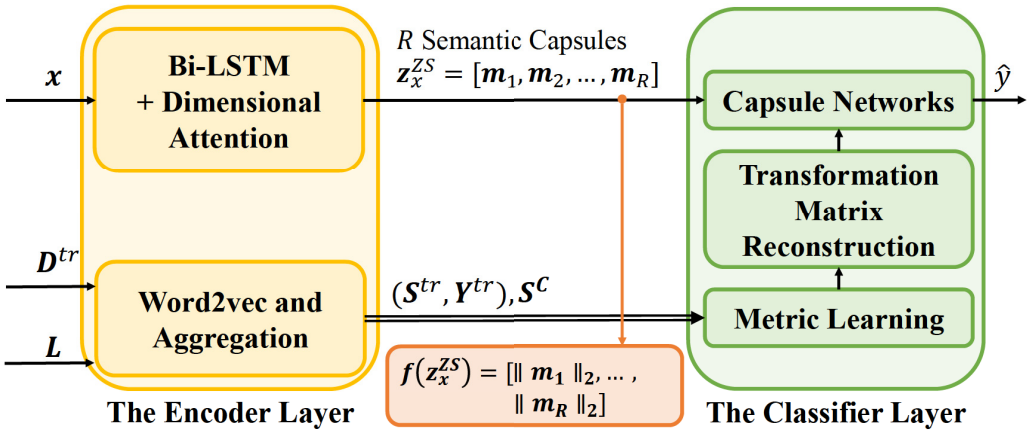


Figure 4.6: How the feature is extracted from ReCapsNet.

our methods on two datasets SNIPS [24] and SMP-2018 [143] and report the micro-averaged recall (accuracy) and F1 scores. The baselines include DeVISE [38], CMT [102], CDSSM [21], Zero-shot DNN [60], IntentCapsNet [119], and ReCapsNet [74]. The average results over 10 runs of our methods and ReCapsNet are reported in Table 4.5, where the results of other baselines are taken from [74].

We use the same setting and hyper-parameters as in ReCapsNet [74]. We set d_z=4 for SNIPS and d_z=12 for SMP-2018. The rest of the parameters of SEG are the same as those used in Section 3.4.1. In addition, we also conduct an ablation study to demonstrate the effectiveness of the proposed semantic enhancement mechanism by testing two variants of our integration ("Sep / o" and "Combine / o") without using it.

Result Analysis From the results in Table 4.5, we can make the following observations: (1) All variants of our integration achieve a significant boost in the overall accuracy and F1 scores on the two datasets, especially on SNIPS, where the performance increase is huge. Each variant leads to a qualitative leap in the

Method	SNIPS						SMP-2018					
	Seen		Unseen		Overall		Seen		Unseen		Overall	
	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1
DeViSE	.9481	.6536	.0211	.0398	.4215	.3049	.8040	.6740	.0270	.0310	.5030	.4250
CMT	**.9755**	.6648	.0397	.0704	.4438	.3271	**.8314**	.7221	.0798	.1069	.5398	.4834
CDSSM	.9549	.7033	.0111	.0218	.4234	.3194	.6653	.5540	.1436	.1200	.4864	.4052
Zero-shot DNN	.9432	.6679	.0682	.1041	.4488	.3493	.7323	.6116	.0590	.0869	.5013	.4316
IntentCapsNet	**.9741**	.6517	.0000	.0000	.4200	.2810	**.8850**	.7281	.0000	.0000	.5375	.4423
ReCapsNet	.9511	.6777	.0994	.1594	.4705	.3826	.8107	**.7417**	.1959	.1727	.5692	.5182
SEG (Sep / o)	.9308	.7501	.3523	.4514	.6014	.5800	.7066	**.7391**	.3848	.3038	.5802	.5681
SEG (Combine / o)	.9217	.7924	.4642	.5321	.6612	.6441	.7054	.7326	.3888	.3116	.5811	.5672
SEG (Sep / w)	.7898	**.8335**	**.6728**	**.6420**	**.7232**	**.7245**	.6624	.7243	**.4779**	**.3627**	**.5899**	**.5823**
SEG (Combine / w)	.8644	**.8658**	**.6961**	**.6931**	**.7685**	**.7674**	.6821	.7359	**.4848**	**.3806**	**.6046**	**.5963**

Table 4.5: Results of generalized zero-shot intent classification equipped with our new intent identifier SEG. “Seen”, “Unseen” and “Overall” respectively denote the prediction performance on the utterances from seen intents, unseen intents, and both seen and unseen intents. The suffixes ”/w” and ”/o” stand for with and without semantic enhancement, respectively. The top 2 results for each metric are marked in bold.

performance on unseen intents. The prediction accuracy (micro-averaged recall) on seen intents may be reduced compared to ReCapsNet and other baselines, since some utterances of seen intents are classified to unseen intents. However, the F1 score on seen intents increases significantly, indicating that it has much higher precision than that of the baselines. (2) The variants of our integration with semantic enhancement significantly outperform those without using it on predicting unseen intents by very large margins. Although their accuracy scores on seen intents are lower, their overall accuracy and F1 scores are consistently better, which confirms the effectiveness of semantic enhancement. (3) It can be seen that the “Combine” variants generally perform much better than the “Sep” variants, especially the one with semantic enhancement (“Combine / w”), which performs outstandingly. It surpasses the performance of “Sep / w” in every metric, demonstrating the usefulness of the simple feature assemble strategy of concatenating the feature representations of ReCapsNet and SEG.

4.6 Chapter Summary

This chapter addressed the challenge of zero-shot and generalized zero-shot intent classification in task-oriented dialogue systems, a critical problem given the rapid emergence of new user intents in real-world applications. The chapter began by identifying the limitations of existing approaches, particularly IntentCapsNet, which struggles with polysemy in semantic capsule extraction and exhibits poor generalization to unknown intents in the generalized zero-shot setting.

To overcome these challenges, we introduced a novel framework, ReCapsNet-ZS, which reconstructs capsule networks for zero-shot intent classification. The proposed methodology incorporates two principal innovations: (1) a dimensional attention mechanism that enhances the extraction of semantic features by addressing polysemy at the embedding level, and (2) a transformation matrix construction scheme that leverages metric learning to better relate seen and unknown intents, thereby improving generalization.

Furthermore, we successfully applied SEG to improve generalized zero-shot intent classification and achieved remarkable performance gain over the most recent competitive method, ReCapsNet. Experiments on real task-oriented dialogue datasets in different languages show that our proposed method outperforms IntentCapsNet and other strong baselines.

Finally, the effectiveness of ReCapsNet-ZS and the injection of SEG were empirically validated through extensive experiments on multilingual, task-oriented dialogue datasets. The results demonstrated that the proposed approach consistently outperformed IntentCapsNet and other strong baselines, particularly in the more challenging generalized zero-shot setting. Ablation studies further confirmed the

complementary benefits of the dimensional attention module and the transformation matrix construction method.

Chapter 5

New Intent Discovery with LLMs' Assistance

5.1 Motivation

Task-Oriented Dialogue Systems (TODS) are designed to assist users in accomplishing specific goals, such as booking flights or resolving technical issues. However, as these systems evolve, they inevitably encounter user intents that were unseen during initial training. New Intent Discovery (NID) aims to enable systems to autonomously identify novel intents while maintaining robust performance on known ones. Such capability is critical for deploying TODS in dynamic real-world environments. Previous approaches have utilized clustering algorithms [99, 87, 17] to group utterances with similar intents, or have implemented semi-supervised pre-training methods for representation learning. These methods face significant limitations: many struggle to learn discriminative semantic representations for unknown intents [71, 89, 138], while others depend on the quality and quantity of

external knowledge [139, 146], hindering scalability and adaptability.

Recent advances in large language models (LLMs) [84] have demonstrated impressive zero-shot reasoning capabilities [47], providing promising solutions for intent recognition. However, implementing these large-scale models in production environments is both complex and resource-intensive. This complexity raises potential privacy issues [57], and the significant computational demands can lead to increased latency, impacting the real-time performance essential for many applications. Consequently, there is a growing interest in exploring the use of LLMs to guide and optimize offline, lightweight NID models.

To leverage the capabilities of LLMs while ensuring lightweight performance and privacy for the NID task, we propose LANID (**L**LM-**A**ssisted **N**ew **I**ntent **D**iscovery), a novel framework that integrates the rich knowledge of LLMs into lightweight encoders using contrastive learning to enhance in-domain NID performance. Specifically, we propose to sampling pairs of utterances and using their relationships discriminated by LLMs as contrastive signals. As establishing relationships for all possible utterance pairs would incur excessively high costs, we propose two data sampling strategies to sample informative and representative utterance pairs. Both strategies leverage unsupervised algorithms to enhance data sampling. The initial strategy employs the concept of core points from DBSCAN [30] to effectively sample representative data. Conversely, the second strategy leverages the K-nearest neighbor algorithm [25] to enhance the probability that the sampled data will accurately represent class boundaries. The proposed selection strategies ensure that the chosen pairs are more representative and thereby enhancing the system's efficiency.

Next, a powerful LLM is employed to determine the relationships between

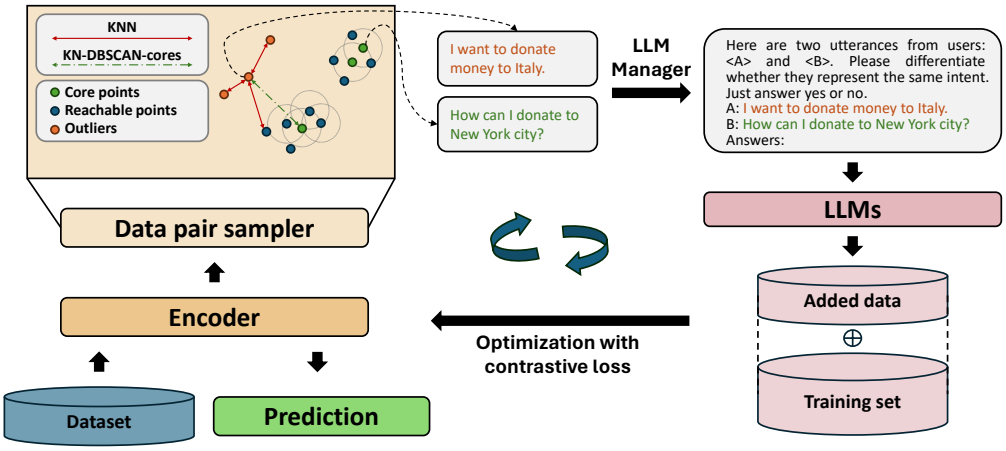


Figure 5.1: Illustration of the proposed LANID framework. First, utterances are encoded by our target light encoder, and informative utterance pairs are sampled from D_{train} using the proposed data pair sampler. Next, the LLM manager inputs the sampled data pairs into the LLM and uses it to annotate the relationship for each utterance pair. Contrastive tasks are then constructed to optimize the target encoder. These three steps are repeated iteratively until convergence is reached. Once this is achieved, we apply the trained encoder to the test set, D_{test}, and utilize the k-means algorithm to cluster the encoded utterances. Utterances within the same cluster are then considered to share the same intent.

the selected utterance pairs. The obtained supervision signals are then used to construct a contrastive learning task. By optimizing a lightweight encoder with a triplet loss objective, LANID enhances the model's ability to discern both known and new intents without compromising efficiency. The data sampling and teaching steps are conducted iteratively, and the training set can be incrementally updated with either LLM-annotated or manually annotated new data, ensuring the system's flexibility and scalability.

5.2 Problem Formulation.

To better align with the practical need for continuous training to recognize new intents, we follow the approach of prior research [138] and evaluate our method in both unsupervised and semi-supervised evaluation settings. We represent an utterance and its intent label as x and y, respectively. y can either belong to the set of unknown intents $\mathcal{C}_u$ or the set of known intents $\mathcal{C}_k$. The training set, validation set, and test set are denoted as D_{train}, D_{val} and D_{test}. These three sets share the same distribution of intent labels. In the unsupervised setting, D_{train} does not contain any annotated utterances, meaning that no intents are known or labeled. In the semi-supervised setting, D_{train} comprises both a labeled dataset, $D_{labeled} = \{(x_i, y_i)|y_i \in \mathcal{C}_k\}$ and an unlabeled dataset $D_{unlabeled}$. The intent labels of utterances in $D_{unlabeled}$ may belong to either $\mathcal{C}_k$ or $\mathcal{C}_u$. In both settings, the goal is to group utterances from D_{test} into clusters, with each cluster representing a distinct intent.

5.3 Method

The proposed LANID framework is illustrated in Figure 5.1. LANID utilizes contrastive learning to distill knowledge from LLMs, thereby facilitating the training of a lightweight encoder for novel intent detection. The contrastive learning process consists of three main steps: 1) sampling selective utterance pairs from D_{train} to form the contrastive tasks based on clustering algorithms, 2) determining the relationships between these utterance pairs using a powerful LLM that provides pseudo-labeling, and 3) fine-tuning the encoder with the LLM's output to distill

the knowledge from the powerful LLM. This three-step training procedure is iteratively repeated until either a predefined condition set by the user is satisfied or convergence is attained. Finally, the utterances in D_{test} are encoded using the optimized encoder and clustered with the k-means algorithm, consistent with previous attempts in NID [138, 146].

Selecting Utterance Pairs We propose injecting knowledge encoded in LLMs into the small model through contrastive learning. In the learning process, the LLM acts like a teacher, while the small model acts like a student. One of the most essential things in the teaching stage is designing contrastive tasks. In this chapter, we propose sampling utterance pairs and asking the LLM to determine the relation between each one. The contrastive tasks are formed with utterance pairs and their pseudo-relation label annotated by LLMs. It is impractical to establish relationships for all possible utterance pairs. Therefore, we propose two selective sampling strategies for utterance pair sampling, which will selectively sample representative and informative data. Next, we will illustrate the two proposed sampling strategies.

Selection based on K-Nearest Neighbors. Starting locally, we first find those utterances that are close to each other (based on the original representation) and determine whether the distribution among them is reasonable. In each iteration, we randomly sample $p\%$ utterances from D_{train}. Then, for each sampled utterance x_i, we search for its top-K nearest neighbors $\mathcal{N}_i^{Near}$ using the Euclidean distance, and we uniformly sample $n_k (n_k < |\mathcal{N}_i^{Near}|)$ utterances from $\mathcal{N}_i^{Near}$. We denote the nearest-neighbor set for x_i as $\mathcal{M}_i^{Near} = \{(x_i, x_j) | x_j \in \mathcal{N}_i^{Near}\}$, where $|\mathcal{M}_i^{Near}| = n_k$.

Selection based on Global Density. It is difficult to divide a data set into exactly a few categories, and there will always be some outliers. Also, the distribution of semantics is usually not uniform, and there are high and low densities of different semantic clusters. We propose a DBSCAN-based [30] sampling approach to reflect the relationship between globally high and low-density regions of semantics. Concretely, we conduct DBSCAN clustering on D_{train} and obtain a set of core points $\mathbf{x}_c$ and a set of non-core points $\mathbf{x}_{nc}$. Then, we randomly sample a subset $\mathbf{x}_{nc}^{'}$ from $\mathbf{x}_{nc}$. For each utterance x_i in $\mathbf{x}_{nc}^{'}$, we search for its m nearest neighbors in $\mathbf{x}_c$, forming a global density set as $\mathcal{M}_i^{Den} = \{(x_i, x_j)|x_j \in \mathcal{N}_i^{Core}\}$, where $\mathcal{N}_i^{Core}$ is the set consisting of the nearest points to x_i in $\mathbf{x}_c$.

LLM Manager The LLM manager is another major module in LANID. It constructs prompts using sampled data and parses the responses from LLMs. We design prompts with three components [86]: schema, regulations, and sentence input. The schema component is intended to guide LLMs to produce responses that meet our desired criteria. To identify the optimal schema for each dataset, several schemas were manually crafted and evaluated based on their performance on $D_{labeled}$. The regulations component constrains the format of the LLM's responses. For simplicity, we uniformly use the phrase "just answer yes or no." The sentence input component consists of utterance pairs sampled as detailed in Section 5.3. Finally, we predict $r(i, j) = 1$ for a data pair (i, j) if 'yes' appears in the LLM's corresponding response; otherwise, $r(i, j) = 0$.

Analysis of LLM Labeling Accuracy A critical aspect of LANID's framework is the quality of pseudo-labels generated by the LLM manager. The reliability

of LLM-generated relationship judgments directly impacts the effectiveness of the knowledge distillation process. While we do not conduct separate accuracy measurements in this work, understanding the potential sources of labeling errors and the framework's robustness to such errors is essential for practical deployment.

LLM mislabeling can occur in several scenarios: (1) utterances with ambiguous or context-dependent meanings, where the intent cannot be reliably determined from the text alone; (2) pairs from semantically related but distinct intents, such as "transfer_money" and "check_balance" in banking domains; and (3) utterances with similar surface forms but different underlying intents. These potential error patterns suggest that while LLMs provide valuable supervision signals, the pseudo-labels may not be perfect and could introduce some noise into the training process.

Importantly, LANID's design incorporates mechanisms that provide robustness to imperfect pseudo-labels. The iterative training procedure and the use of multiple utterance pairs help to average out individual labeling errors. Furthermore, the contrastive learning objective with margin-based loss provides some tolerance to noisy labels, as the model learns to separate intent clusters rather than fitting exact pairwise relationships. This robustness to imperfect supervision is a key advantage of the LANID framework, enabling practical deployment without requiring perfect LLM accuracy.

Robustness to LLM Errors and Mitigation Strategies Given that LLM-generated pseudo-labels are not perfectly accurate, it is crucial to examine LANID's robustness to such errors and explore potential mitigation strategies. The framework incorporates several design choices that inherently provide resilience to label noise.

First, the iterative sampling and training procedure helps mitigate the impact

of individual labeling errors. In each iteration, new utterance pairs are sampled and labeled, providing multiple opportunities to learn correct relationships. If the LLM makes an error on a particular pair in one iteration, subsequent iterations may sample different pairs involving the same utterances, potentially correcting the error through majority voting effects. Additionally, as the encoder improves through training, the quality of sampled pairs also improves, leading to more reliable pseudo-labels in later iterations.

Second, the contrastive learning framework with triplet loss provides natural robustness to noisy labels. Unlike classification tasks where a single mislabeled example directly corrupts the decision boundary, contrastive learning focuses on relative relationships between samples. The margin-based loss allows some tolerance for imperfect pairwise relationships, as long as the overall clustering structure is preserved. Furthermore, by sampling multiple negative examples for each positive pair, the model learns from aggregate patterns rather than individual labels, reducing sensitivity to isolated errors.

To further enhance robustness, we propose several potential mitigation strategies for future work: (1) **Confidence-based filtering**, where the LLM provides confidence scores for its judgments, and only high-confidence labels are used for training; (2) **Ensemble labeling**, where multiple LLMs or multiple prompts are used to label the same pairs, with disagreements resolved through voting or confidence weighting; (3) **Active learning**, where the model identifies uncertain or potentially mislabeled pairs and requests human verification for critical cases; (4) **Noise-robust loss functions**, such as symmetric cross-entropy or bootstrapping losses, which have been shown effective in learning from noisy labels.

Our ablation studies (detailed in the Experiment section) demonstrate that

LANID maintains strong performance even when artificial noise is injected into the pseudo-labels, validating the framework's inherent robustness. This robustness is essential for practical deployment, as it reduces the dependency on perfect LLM performance and enables the use of smaller, more efficient LLMs that may have slightly lower accuracy but offer better cost-performance trade-offs.

Training and Optimization To optimize the representation of the text encoder on domain data, we collect pairs of positive samples from $\mathcal{M}_i^{Near}$ and/or $\mathcal{M}_i^{Core}$, with their relationships $r(i, j)$ determined by the LLM manager. For each positive sample pair $\{(x_i, x_j)\}$, we randomly sample k_n utterances from D_{train} as negative samples to better represent the distribution of the entire dataset, assuming the dataset distribution is not extreme. This approach forms a dataset $D_f = (x_i, p_i, n_i)$ of triplets. We then fine-tune the model using a triplet margin loss defined as:

$$\mathcal{L}(x_i, p_i, n_i) = max(d(x_i, p_i) - d(x_i, n_i) + mgn, 0), \tag{5.1}$$

where x_i serves as the anchor point, mgn. represents a hyperparameter, and p_i and n_i are its positive and negative samples, respectively. The function $d(x_i, y_j) = \|x_i - y_j\|$ represents the distance, and the margin is a hyperparameter that determines the minimum desired difference between $d(x_i, p_i)$ and $d(x_i, n_i)$.

As training progresses, the quality of the text encoder's representation improves, leading to enhanced sampling outcomes. In practice, the process of selecting utterance pairs and querying the LLM manager recurs every T epochs, with the fine-tuning dataset D_f and the text encoder being incrementally updated during this iterative process.

Finally, upon completing the training stage, we apply the optimized encoder on the test set D_{test} and employ the k-means algorithm to do clustering. Utterances that fall within the same cluster are considered to share the same intent.

Relationship to Explicit Intent Representations While LANID focuses on learning latent representations of intents through contrastive learning, it is worth discussing the complementary approach of explicit intent representations. Recent work such as FolkScope [134] constructs intention knowledge graphs that explicitly model intent concepts and their relationships in e-commerce domains. These explicit representations provide structured, interpretable knowledge about user intentions, including hierarchical relationships, attribute constraints, and commonsense associations.

The explicit representation approach offers several advantages: (1) interpretability, as the intent structure is directly observable and can be inspected by domain experts; (2) incorporation of domain knowledge, as the knowledge graph can be enriched with expert-defined relationships; and (3) reasoning capabilities, enabling logical inference about intent relationships. However, constructing and maintaining such explicit knowledge graphs requires significant manual effort and domain expertise, limiting scalability across diverse domains.

LANID's latent representation approach complements these explicit methods by automatically learning intent representations from data without requiring manual knowledge engineering. The contrastive learning framework enables LANID to capture nuanced semantic similarities that may be difficult to explicitly encode. Future work could explore hybrid approaches that combine the interpretability of explicit knowledge graphs with the scalability and adaptability of learned latent

Table 5.1: Hyper-parameter Settings: MinPts denotes the minimum number of points needed within a specified radius (epsilon) to form a dense region in DBSCAN.

	K	p	n_k	m	k_n	T	#Epoch	MinPts
Banking	50	0.1	2	5	2	3	10	4
Stackoverflow	50	0.05	2	8	2	2	10	4
M-CID	50	0.2	2	5	2	3	20	4

representations, potentially using LLMs to bridge between these two paradigms.

5.4 Experiment

5.4.1 Experimental Details

We evaluate LANID using three intent recognition benchmarks. The BANKING dataset [15] includes 13,083 utterances distributed across 77 intents within the banking domain. The StackOverflow dataset [125] comprises 20,000 queries collected from an online question-answering platform[1], categorized into 20 different categories. The M-CID dataset [5] consists of 1,745 utterances associated with 16 intents specifically related to COVID-19 services.

Experimental Setup. Our proposed method is evaluated under both unsupervised and semi-supervised settings. We use three clustering evaluation metrics: normalized mutual information (NMI) [31], adjusted Rand index (ARI) [131], and accuracy (ACC).

[1]https://stackoverflow.com/

Table 5.2: Performance on Semi-Supervised NID with Varying Known Class Ratios: Best results for each dataset are highlighted in bold. The Known Class Ratio (KCR) is defined as $\frac{|\mathcal{C}_k|}{(|\mathcal{C}_k|+\mathcal{C}_u)}$. A 10% subset of each known class was randomly sampled to create $D_{labeled}$. Results are shown for the three LANID variants as described in Table 5.3.

		Banking			**StackOverflow**			**M-CID**		
	Methods	**NMI**	**ARI**	**ACC**	**NMI**	**ARI**	**ACC**	**NMI**	**ARI**	**ACC**
KCR-25%	BERT-KCL	53.85	20.07	28.79	35.47	16.80	32.88	29.35	11.58	24.76
	DAC	69.85	37.16	49.67	53.97	36.46	53.96	49.83	27.21	43.72
	MTP	79.17	50.83	62.05	74.86	62.27	77.20	70.53	45.76	64.76
	MTP-CLNN	83.88	60.76	70.91	78.38	65.80	80.10	78.30	65.32	78.30
	LANID-Near	85.28	63.48	**72.47**	**80.83**	**65.86**	**83.30**	81.91	70.30	81.09
	LANID-DBSCAN	84.74	62.22	70.13	74.74	60.54	73.70	80.04	69.69	83.09
	LANID	**85.51**	**64.23**	71.40	79.55	63.23	81.80	**85.11**	**75.66**	**86.82**
		Banking			**StackOverflow**			**M-CID**		
	Methods	**NMI**	**ARI**	**ACC**	**NMI**	**ARI**	**ACC**	**NMI**	**ARI**	**ACC**
KCR-50%	BERT-KCL	62.86	30.16	40.81	57.63	41.90	56.58	42.48	22.83	38.11
	DAC	76.41	47.28	59.32	70.78	56.44	73.76	63.27	43.52	57.19
	MTP	82.12	56.43	67.34	76.58	65.55	82.50	70.53	45.76	64.76
	MTP-CLNN	86.42	66.66	74.97	81.41	**72.15**	**86.00**	79.34	66.18	78.80
	LANID-Near	**86.83**	**67.41**	**76.10**	81.62	64.32	81.30	81.20	69.54	81.95
	LANID-DBSCAN	85.62	64.35	72.44	81.19	65.75	81.40	79.16	67.85	80.80
	LANID	86.31	66.53	75.49	**82.07**	70.51	83.00	**81.58**	**70.66**	**82.81**
		Banking			**StackOverflow**			**M-CID**		
	Methods	**NMI**	**ARI**	**ACC**	**NMI**	**ARI**	**ACC**	**NMI**	**ARI**	**ACC**
KCR-75%	BERT-KCL	72.18	44.29	58.70	70.38	57.98	71.50	54.22	34.60	52.15
	DAC	79.99	54.57	65.87	75.31	60.02	78.84	71.41	54.22	69.11
	MTP	84.61	63.23	72.76	80.41	70.01	81.10	77.90	64.57	77.65
	MTP-CLNN	87.24	68.77	77.14	80.99	72.14	85.80	80.12	67.40	79.37
	LANID-Near	87.59	**70.13**	**78.51**	82.14	73.05	85.80	81.13	69.75	**83.09**
	LANID-DBSCAN	86.79	67.18	74.35	**83.74**	**76.45**	**88.30**	80.65	70.24	82.52
	LANID	**87.64**	68.89	76.56	82.80	74.33	87.50	**82.16**	**70.56**	82.23

Baselines. We compare LANID with both unsupervised and semi-supervised NID state-of-the-art (SOTAs). The unsupervised NID SOTAs include the SAE series [123], MTP, and CLNN [146]. The semi-supervised baselines include BERT-KCL [52], DAC [138], MTP, and CLNN [146].

Implementation Details. We use the default settings of CLNN [146] and continue to train the model pretrained by MTP-CLNN as a post-finetuning stage. Further training can also be conducted on other NID baselines. For LLMs, we use the *gpt-3.5-turbo*[2] model. The hyperparameters of LANID as shown in Table 5.1 are selected based on performance on the validation set.

5.4.2 Result Analysis

Tables 5.3 and 5.2 summarize the performance of LANID compared to state-of-the-art (SOTA) methods in both unsupervised and semi-supervised settings across three intent recognition benchmarks. The results reveal several key observations: (1) LANID and its variants excel in unsupervised learning, beating other baselines due to effective LLM labeling that compensates for missing supervised signals. (2) LANID consistently surpasses baselines in most cases, proving its effectiveness. (3) Although the combination of two neighborhood sampling strategies generally works well, relying solely on the DBSCAN-based sampling strategy can sometimes hinder performance. This is because the LLM is constrained to make binary judgments and retain only positive pairs. For many outliers, their nearest cores may not express the same intent, thereby reducing the size of D_f and leading to overfitting problems.

[2]https://platform.openai.com/docs/models/gpt-3-5

Discussion: Direct LLM Usage vs. Knowledge Distillation An important question regarding LANID's design is why we employ knowledge distillation from LLMs to lightweight encoders rather than directly using LLMs for end-to-end intent discovery. This design choice is motivated by several practical considerations that are critical for real-world deployment.

First, **computational efficiency and latency** are paramount in production dialogue systems. Direct LLM inference for clustering thousands of utterances would require extensive API calls or local GPU resources, resulting in prohibitive costs and latency. For example, clustering 10,000 utterances with pairwise comparisons would require $O(n^2)$ LLM calls, which is impractical. In contrast, LANID uses the LLM only during offline training to label a small subset of representative pairs (typically ¡1% of all possible pairs), and then performs efficient inference using the lightweight encoder. This approach reduces inference time by orders of magnitude while maintaining comparable performance.

Second, **privacy and data security** concerns make it challenging to send sensitive user utterances to external LLM APIs in many enterprise settings. LANID addresses this by using LLMs only during the training phase, which can be conducted in secure environments, while the deployed lightweight encoder operates entirely on-premise without external dependencies. Third, **cost considerations** are significant: continuous LLM API usage for production traffic would incur substantial ongoing costs, whereas LANID's one-time training cost is amortized across all future inferences.

Finally, **domain adaptation and fine-tuning** are more feasible with lightweight encoders. The distilled model can be further fine-tuned on domain-specific data or incrementally updated as new intents emerge, operations that would be impractical

Table 5.3: Performance on Unsupervised NID: Best results for each dataset are highlighted in bold. Results are shown for three LANID variants: LANID-Near (KNN-based sampling), LANID-DBSCAN (DBSCAN sampling), and LANID (combination of both strategies)

	Banking			**StackOverflow**			**M-CID**		
Methods	**NMI**	**ARI**	**ACC**	**NMI**	**ARI**	**ACC**	**NMI**	**ARI**	**ACC**
SAE-KM	60.12	24.00	37.38	48.72	23.36	37.16	51.03	43.51	52.95
SAE-DEC	62.92	25.68	39.35	61.32	21.17	57.09	50.69	44.52	53.07
SAE-DCN	62.94	25.69	39.36	61.34	34.98	57.09	50.69	44.52	53.07
MTP	77.25	47.80	59.12	61.35	45.77	61.90	70.53	45.76	64.76
MTP-CLNN	82.15	57.68	66.88	75.20	63.13	79.20	80.03	67.39	79.94
LANID-Near	83.44	58.28	66.75	79.56	66.67	83.40	80.80	69.86	81.38
LANID-DBSCAN	83.21	58.02	65.78	81.25	72.86	85.30	80.41	68.10	79.08
LANID	**84.12**	**60.40**	**70.58**	**81.25**	**72.96**	**86.60**	**82.64**	**71.36**	**82.52**

or impossible with large frozen LLMs. Our design philosophy prioritizes practical deployability while leveraging LLMs' knowledge during training, achieving a favorable balance between performance, efficiency, and operational feasibility.

5.5 Chapter Summary

This chapter presented LANID, a novel framework for New Intent Discovery (NID) in Task-Oriented Dialogue Systems (TODS), addressing the persistent challenge of identifying emerging user intents in dynamic conversational environments. The proposed methodology leverages the semantic reasoning capabilities of Large Language Models (LLMs) to enhance the representational power of lightweight NID encoders through contrastive learning, thereby balancing the need for both adaptability and computational efficiency.

LANID operates by iteratively sampling informative utterance pairs from the training data using two complementary unsupervised strategies: K-nearest neigh-

bors (KNN) and Density-Based Spatial Clustering of Applications with Noise (DBSCAN). These strategies ensure that the sampled pairs are both representative and diverse, capturing local and global semantic relationships within the data. The relationships between sampled utterance pairs are then determined by querying a powerful LLM, which acts as a teacher to provide pseudo-labels. These pseudo-labels serve as supervision signals for constructing contrastive learning tasks, wherein a lightweight encoder is fine-tuned using a triplet loss objective. This process is repeated iteratively, allowing the encoder to incrementally distill semantic knowledge from the LLM and adapt to new intent categories.

Finally, the experimental evaluation, conducted on three benchmark datasets (BANKING, StackOverflow, and M-CID), demonstrates that LANID consistently outperforms state-of-the-art baselines in both unsupervised and semi-supervised NID settings. The results highlight several key observations: (1) LANID's LLM-guided labeling effectively compensates for the absence of annotated data, particularly in unsupervised scenarios; (2) the integration of both KNN and DBSCAN sampling strategies generally yields superior performance, though reliance solely on global density sampling may introduce overfitting in the presence of outliers; and (3) the framework's iterative design supports incremental updates, enhancing scalability and flexibility for real-world deployment.

Chapter 6

Modeling User Behavior for Product Search

6.1 Motivation

Convenience drives the growth of e-commerce platforms such as Taobao[1] or Amazon[2]. Product search is an essential module in online shopping platforms, which guides users to browse and purchase products from a huge collection of commodities. Product search has its unique characteristics, making it distinct from web search, where information retrieval has made considerable progress. First, in web search engines, web pages are usually represented by long descriptive texts, while in e-commerce platforms, products are mainly represented by short texts such as titles and reviews, which may not always be informative. Second, other than textual representations, products are also associated with diverse relational

[1] https://world.taobao.com/
[2] https://www.amazon.com/

data including ontology, spec sheet, figures, etc. Third, there are various types of user-item interactions in e-commerce platforms. A user can browse, click, review, or purchase a product, or simply put it in his/her cart. Besides, there exist other structural information such as query-reformulation, shop browsing or category browsing, and shopping cart checkout.

It would be highly desirable yet challenging to utilize such rich information for personalized product search. Existing methods mainly exploit text data. Among them, a recent line of research [45, 3, 1, 9] proposes to projects queries, items, and users into the same latent space and learn the representations of all entities with language modeling and information retrieval tasks, which enables the model to learn domain-specific semantic representations. However, they are limited in their ability to model user preferences, which is the core problem in product search. A common way to represent users is by the products they've visited during a period of time, but long-term historical user behavior normally contains noisy preference signals. HEM [3] suffers from this problem since it represents a user with all his/her reviews of purchased products. ZAM [1], TEM [9], and RTM [10] employ attentive models such as Transformer-based encoder to model user preferences and take into account both user behavior and query. For computational efficiency, user behavior sequences are usually truncated, and only recent behaviors are considered. While this helps to eliminate noisy preference signals, short-term user behavior may not contain sufficient preference signals.

To capture more useful user preference signals, it is a natural idea to explore various user-product interactions and product co-occurrence relationships, which are usually encoded in a graph. Some recent efforts [75, 2] have been devoted to exploiting structural graph information for personalized product search. Ai et al. [2]

proposed a dynamic relation embedding model (DREM). DREM constructs a unified knowledge graph to encode diverse relations and dynamic user-search/purchase behaviors and models the structural relationships via graph regularization. Liu et al. [75] proposed graph embedding based structural relationship representation learning (GraphSRRL), which explicitly models the structural relationships, such as two users visiting the same product by a same query or a user visiting the same product by two different queries. While DREM and GraphSRRL can model complex relationships, they include all previous user behaviors for preference modeling and may suffer from the noise induced by overly diverse signals.

In this work, we propose to explore local and global user behavior patterns on a user successive behavior graph (SBG) for user preference modeling. The SBG is constructed by utilizing *short-term* actions of all users, which collectively form a global behavior graph with rich relations among products. To capture implicit user preference signals and collaborative patterns, we employ graph convolution to learn enriched product representations, which can be subsequently used for user preference modeling. Since user purchase behaviors are often sparse, it is helpful to explore high-order information on the SBG to model potential user interest, which requires stacking many graph convolution layers and leads to the well-known over-smoothing problem. To address this issue, we adopt an efficient jumping graph convolution layer that can effectively alleviate the over-smoothing effect. To showcase the usefulness of our approach, we integrate it into a state-of-the-art latent space based model ZAM [1] and evaluate its performance on eight Amazon public benchmarks. The results show that our approach can significantly improve upon the base model and achieve better performance than other graph-based methods including DREM [2] and GraphSRRL [75]. It is worth noting that our approach is

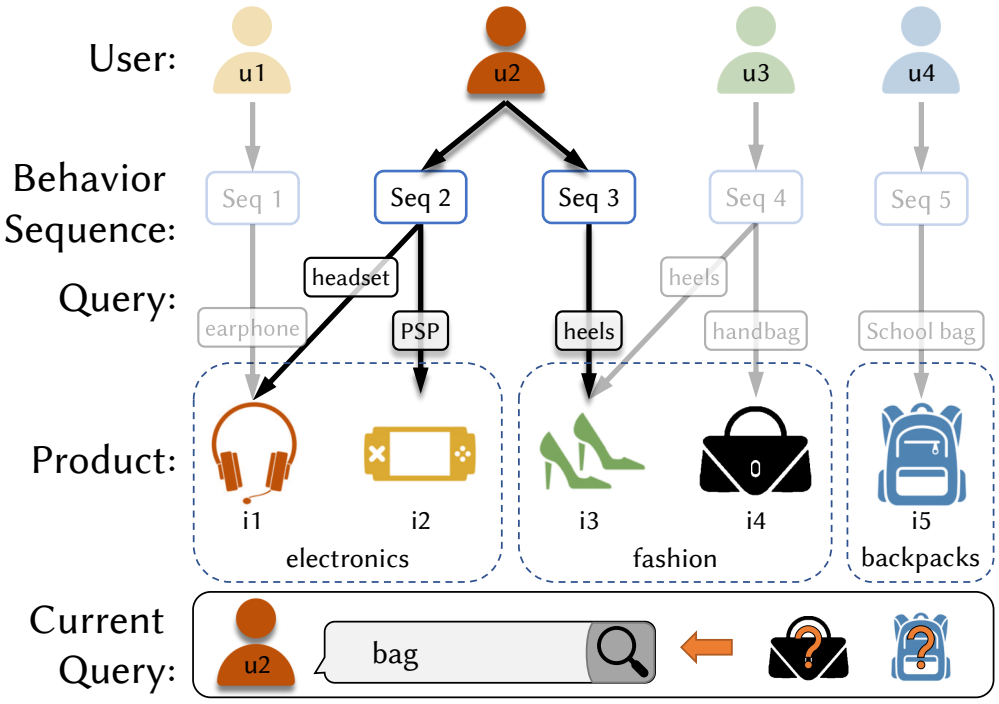


Figure 6.1: Illustration of exploiting global user successive behavior for personalized product search. Given the query *bag* issued by user u_2, the system is expected to retrieve suitable bags satisfying the user intent. By modeling the global user successive behavior graph with graph convolution, our proposed approach can capture implicit preference signals and yield desirable results.

generic and can be potentially applied in any product retrieval method that models users using their purchase history.

6.2 Motivation and Insight

In this section, we discuss the limitations of existing product search methods in user preference modeling and provide insight on our proposed approach.

6.2.1 Limitations of Existing Methods

Existing product search methods commonly model user preferences by considering their long-term or short-term behavior, but both of them have limitations.

Long-term user behavior may contain noisy preference signals. Typically, the long-term preference of a user is represented by all his/her historical interactions during a long period of time, which may contain many items and be overly diverse. For example, as shown in Figure 6.1, the current query of user u_2 is "*bag*", but she has visited a couple of electronic devices in her purchase history, which are not related to bags. Therefore, representing u_2's preference with all his/her historical engagement may impose negative effect on the current search induced by the irrelevant products in the purchase history. This problem is even more severe in HEM [3], which models users independently with their previous purchase reviews that may contain a lot noisy textual information.

Short-term user behavior may not contain enough preference signals. To address the above-mentioned issue, recent works including ZAM [1], TEM [9] and RTM [10] only consider recent user actions in a short span of time and adopt attentive models such as Transformer-based encoder to put more emphasis on relevant products in the purchase record. However, a user's recent behavior may not contain enough preference signals for product retrieval. For example, when user u_2 searches for "bag", the remotely relevant item in his/her purchase history is "heels", which is not directly relevant to bags. Therefore, it is hard to tell whether the user is looking for handbags, backpacks, or other kinds of bags.

6.2.2 Insight of Our Proposed Approach

In this work, we propose to overcome the limitations of prior works in user preference modeling by exploring local and global user behavior patterns on a user successive behavior graph (SBG), which is constructed by utilizing *short-term* actions of *all* users. We then exploit *high-order* relations in the SBG to capture implicit collaborative patterns and preference signals with an efficient jumping graph convolution and learn enriched product representations for user preference modeling. Our approach addresses the aforementioned problems in the following two aspects.

Expanding the set of potentially intended products. While short-term user behavior usually contains a limited number of products which may be inadequate to reflect user preferences, the *global* SBG connects the products recently purchased by a user to other relevant or similar ones on the graph. In effect, it expands the set of potentially intended products of the user. For example, in Figure 6.1, the heels i_3 is connected to the handbag i_4 because of the co-occurrence of i_3 and i_4 in the behavior sequence Seq 4 of user u_3. On the global SBG, the behavior sequences Seq 3 and Seq 4 are connected by i_3, and hence i_4 could be a potentially intended product for user u_2.

Making connected products more similar in the latent space. Since a user is often represented as the ensemble of the products he/she bought, learning better product representations is crucial for user preference modeling. We leverage graph convolution to exploit the connectivity patterns in the SBG and make the embeddings of connected products more similar. The enriched product representations can better reflect user preference. For example, by graph convolution, the fashion

items i_3 and i_4 will be more similar. Therefore, when user u_2 searches for "bag", the handbag i_4 would be ranked higher than the backpack i_5, because i_4 is more similar to i_3 than i_5, and i_3 partially represents user u_2.

6.3 Method

In this section, we present our proposed approach in detail, which is built on the popular latent space based product search framework that learns semantic representations for products, users, and text tokens in the same embedding space. On top of it, we aim to learn enriched product representations to better represent users for search personalization. As shown in Figure 6.2, we first construct a successive behavior graph from observed user behavior sequences. Then, we employ an efficient graph convolution with jumping connections to enrich product representations. The enriched product vectors can subsequently be used to represent users fro better preference modeling.

Latent Space Based Product Search Framework The goal of product search is to retrieve the most relevant products from a candidate pool C for a user with his/her current query. In this work, we adopt a typical latent space based generative framework for product search [45, 3, 1, 9], which learns embeddings for all products, users, and text tokens. The most relevant products are retrieved by matching product embeddings with the current context, which is usually represented by the user and query.

More formally, after receiving a query q from a user u, the system is expected to predict the probability $P(i|u, q)$ of user u purchasing product i, for each product in

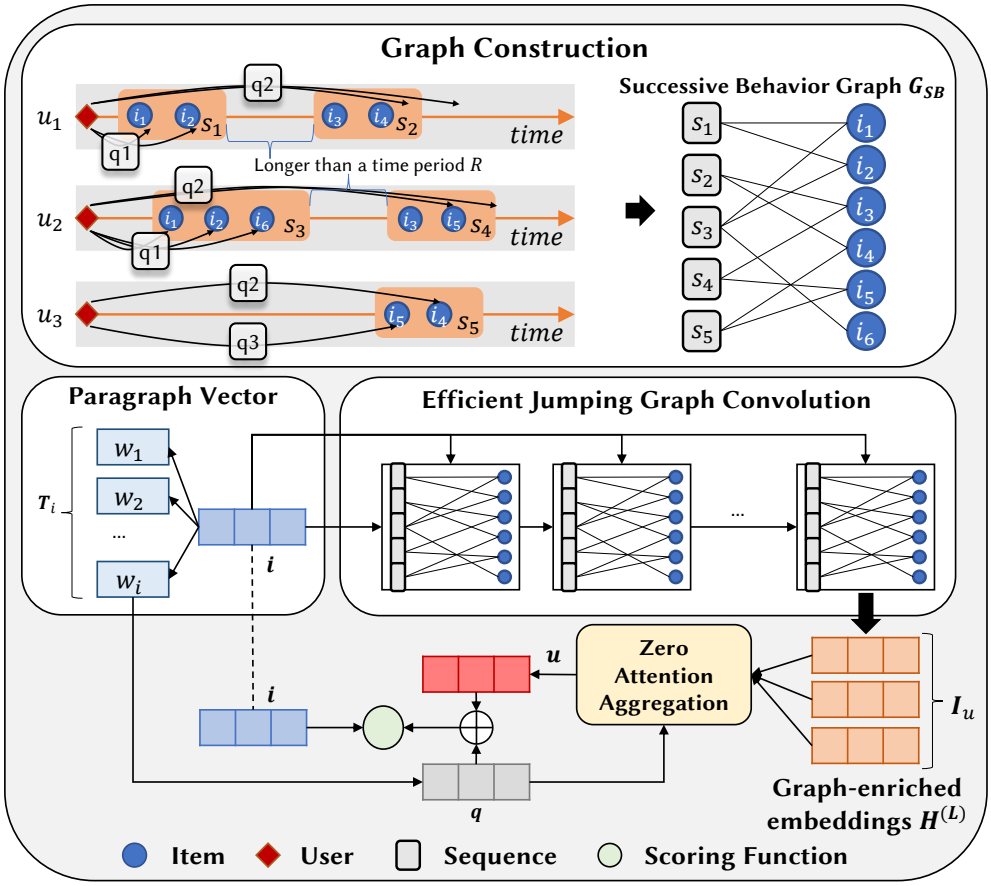


Figure 6.2: The framework of our proposed SBG model.

the candidate set C, and then rank all the products by the probability. The available information includes the purchasing history of all the users and the text associated with the products, such as product name, product description, and product review. The latent space based product search framework learns entity embeddings from two tasks: the product retrieval task and the language modeling task.

Product Retrieval Task.

It aims to retrieve relevant products w.r.t. the current query. In [3], the user intent $\boldsymbol{M}_{uq}$ is represented by a mix of the query $\boldsymbol{q}$ and the user's preference vector

$\boldsymbol{u}$:

$$\boldsymbol{M}_{uq} = \lambda \boldsymbol{q} + (1 - \lambda)\boldsymbol{u}, \tag{6.1}$$

where $\lambda \in [0, 1]$ is a balancing parameter. As such, $\boldsymbol{M}_{uq}$ encodes both the semantic meaning of the query and user preference. Then, the probability of purchasing product i is computed as

$$P(i|u, q) = \frac{\exp(f(\boldsymbol{i}, \boldsymbol{M}_{uq}))}{\sum\limits_{i' \in C} \exp(f(\boldsymbol{i'}, \boldsymbol{M}_{uq}))}, \tag{6.2}$$

where C is the set of all possible candidate products and f is a similarity measure such as cosine similarity.

Language Modeling Task. It aims to learn the embeddings of queries and products by modeling the text information. [3] proposes to jointly learn the word embedding $\boldsymbol{w}$ and product embedding $\boldsymbol{i}$ from the product's associated text by the paragraph vector (PV) model [64]. The PV model assumes that words or tokens can be generated from the entity and maximizes the likelihood

$$P(T_i|i) = \prod_{w \in T_i} \frac{\exp(\tau(w, i))}{\sum_{w' \in V} \exp(\tau(w', i))}, \tag{6.3}$$

where τ denotes a scoring function of product i and its associated word w, and T_i is the set of words associated with i.

With the learned word embeddings, a query is then represented by a function of the word embeddings:

$$\boldsymbol{q} = \phi(\{\boldsymbol{w}_q | w_q \in q\}), \tag{6.4}$$

where w_q is a word of query q, and ϕ can be a non-linear sequential encoder such as LSTM or Transformer. Since queries are usually short and the order of words often does not matter, we simply use an average function to obtain the query embedding.

Efficient Graph Convolution with Jumping Connections As mentioned in Sec. 6.2, to improve user preference modeling and learn better product representations, we propose to utilize a global successive behavior graph and perform graph convolution over the graph to capture implicit and complex collaborative signals. Here, we adopt an efficient graph convolution layer with jumping connections and provide theoretical analysis to show its advantage.

Efficient Graph Convolution. In the past few years, graph convolutional networks (GCN) and variants [59, 108, 46, 19] have been successfully applied to learn useful graph node representations for various graph learning and mining tasks. In each layer of GCN, it performs feature propagation and transformation with connected nodes in the graph:

$$\boldsymbol{H}^{(l)} = \sigma\left(\hat{\boldsymbol{A}}\boldsymbol{H}^{(l-1)}\boldsymbol{W}^{(l)}\right), \tag{6.5}$$

where $\hat{\boldsymbol{A}} = \boldsymbol{I} + \boldsymbol{D}^{-1}\boldsymbol{A}$ is the (normalized) adjacency matrix with self-loops, and $\boldsymbol{D}$ is the degree matrix. $\boldsymbol{H}^{(l)}$ is the node embeddings produced by layer l. $\boldsymbol{W}^{(l)}$ denotes trainable parameters, and σ is a non-linear function such as $\mathrm{ReLU}(\cdot)$.

The projection layers (trainable parameters $\boldsymbol{W}^{(l)}$) and activation layers (σ) as shown in Eq. (6.5) are commonly included in many GCN-based methods. However, as observed from our empirical study, the projection layers may distort the semantic product representations learned by language modeling in methods such as ZAM or

HEM. Hence, we propose to use graph convolution without the projection layers to enrich product representations. Following the efficient design in Li et al. [67], we remove the projection layers and activation layers. Further, we add a balancing parameter ω to control the strength of self-information:

$$\boldsymbol{H}^{(l)} = \left(\omega \boldsymbol{I} + (1-\omega)\boldsymbol{D}^{-1}\boldsymbol{A}\right) \boldsymbol{H}^{(l-1)}. \tag{6.6}$$

Jumping Graph Convolution Layer. Since user purchase behavior is often sparse, it is helpful to aggregate high-order information on the successive behavior graph to model potential user interest, as discussed in Sec. 6.2.2. However, the ordinary graph convolution suffers from the well-known over-smoothing problem [66], i.e., stacking too many convolution layers may make the node features (product representations) indistinguishable. To address this issue, Chen et al. [19] proposed GCNII that adds initial residual connections to each GCN layer. We follow the same design to add jumping connections, i.e., feeding each convolution layer an additional input of the initial product representations $\boldsymbol{H}^{(0)}$:

$$\tilde{\boldsymbol{H}}^{(l)} = \left(\omega \boldsymbol{I} + (1-\omega)\boldsymbol{D}^{-1}\boldsymbol{A}\right) \left(\beta \boldsymbol{H}^{(0)} + (1-\beta)\tilde{\boldsymbol{H}}^{(l-1)}\right), \tag{6.7}$$

where β is a weight parameter determining the portion of initial features. Our experiments in Sec. 6.4.2 verify the effectiveness of jumping connections, which can alleviate the over-smoothing effect and enable utilizing high-order relations on the graph.

Further, we provide a theoretical analysis of jumping group convolution by measuring the diversity of product representations after graph convolution using

Laplacian-Beltrami operator $\Omega(\cdot)$ [23]. We compare the diversity of product representations with jumping connections ($\tilde{\boldsymbol{H}}^{(l)}$ in Eq. (6.7)) and those without jumping connections ($\boldsymbol{H}^{(l)}$ in Eq. (6.6)). We employ Laplacian-Beltrami operator, which measures the total variance of connected nodes:

$$\Omega(\boldsymbol{H}) = \sum_{k} \sum_{i,j} a_{ij} (\boldsymbol{H}_{i,k} - \boldsymbol{H}_{j,k})^2. \tag{6.8}$$

High $\Omega(\boldsymbol{H})$ indicates high diversity, and low $\Omega(\boldsymbol{H})$ indicates severe over-smoothing. The following theorem shows jumping connections can substantially alleviate the over-smoothing effect of graph convolution.

Theorem 1. *If the initial diversity $\Omega(\boldsymbol{H}^{(0)}) > 0$, then for any integer $l > 0$, $\beta \in (0, 1)$, and $\omega \in (0.5, 1)$, $\tilde{\boldsymbol{H}}^{(l)}$ is strictly more diverse than $\boldsymbol{H}^{(l)}$, i.e.,*

$$\Omega\left(\tilde{\boldsymbol{H}}^{(l)}\right) > \Omega\left(\boldsymbol{H}^{(l)}\right). \tag{6.9}$$

When l approaches infinity, jumping connections can prevent the diversity of product representations from collapsing to 0 (over-smoothing), i.e.,

$$\lim_{l \to \infty} \Omega\left(\tilde{\boldsymbol{H}}^{(l)}\right) > \lim_{l \to \infty} \Omega\left(\boldsymbol{H}^{(l)}\right) = 0. \tag{6.10}$$

Proof. The proof is provided in the Appendix. □

Modeling User Behavior with Graph Convolution Here, we show how to utilize local and global user behavior patterns to improve user representations in a zero attention model with the efficient jumping graph convolution. First, we construct a bipartite successive behavior graph. Then, we stack multiple jumping

graph convolution layers to enrich product representations. Finally, we incorporate the graph-enriched product vectors into the zero attention model for user preference modeling.

Graph Construction. To construct the successive behavior graph, we first define successive behavior sequences in the training set. First, we sort all the observed purchased records in a chronological order for each user. If the time interval between two consecutive actions is within a period R (e.g., a day, a week, or a month), the two actions are considered as successive and will be placed in the same successive behavior sequence. Then, we construct a successive behavior graph G_{SB}, which is a bipartite graph between sequences and products. If and only if a product i is in a sequence S, we form an edge between i and S, denoted as $G_{SB}(i, S) = 1$.

Enriching Product Representations with Graph Convolution. To enrich product representations, we apply jumping graph convolution as introduced in Sec. 6.3 on the successive behavior graph G_{SB}. Let $\boldsymbol{h}_j^{(0)}$ denote the input of the first graph convolution layer for any entity j. We use the embeddings learned with PV as $\boldsymbol{h}_i^{(0)}$ for a product i. For a sequence s, $\boldsymbol{h}_s^{(0)}$ is randomly initialized. We then apply L efficient jumping graph convolution layers as defined in Eq. (6.7) and obtain the graph-enriched product embedding $\tilde{\boldsymbol{h}}_i^{(L)}$ for each product.

Using Graph-enriched Product Representations for User Preference Modeling.

Based on the observation that the effect of personalization varies significantly in respect of query characteristics. Ai et al. [1] proposed ZAM that introduces a zero-vector to adaptively control the degree of personalization. The representation

of a user u is composed of his/her recently visited products, which is computed as

$$\boldsymbol{u} = \sum_{i \in I_u \cup \mathbf{0}} \frac{\exp(s(q,i))}{\exp(s(q,\mathbf{0})) + \sum_{i' \in I_u} \exp(s(q,i'))} \boldsymbol{i}, \tag{6.11}$$

where I_u is the product set in user u's history visit records, $\mathbf{0}$ is a zero vector. The attention score $s(q,i)$ of a given product i w.r.t. the current query q is defined as

$$s(q,i) = (\boldsymbol{i}^\top \tanh(\boldsymbol{W}_f^\top \boldsymbol{q} + \boldsymbol{b}_f))^\top \boldsymbol{W}_h, \tag{6.12}$$

where $\boldsymbol{W}_h \in \mathbb{R}^{d_a}$, $\boldsymbol{W}_f \in \mathbb{R}^{d \times d_a \times d}$, $\boldsymbol{b}_f \in \mathbb{R}^{d \times d_a}$ are the trainable parameters, and d_a is the hidden dimension of the user-product attention network. In particular, $exp(s(q,\mathbf{0}))$ is calculated by Eq. (6.12) with $\boldsymbol{i}$ as a learnable inquiry vector $\mathbf{0'} \in \mathbb{R}^d$.

To incorporate the graph-enriched product representations for user preference modeling, we use $\tilde{\boldsymbol{h}}_i^{(L)}$ to substitute $\boldsymbol{i}$ in Eq. (6.11) and Eq. (6.12), i.e.,

$$\begin{aligned} \boldsymbol{u} &= \sum_{i \in I_u \cup \mathbf{0}} \frac{\exp(s(q,i))}{\exp(s(q,\mathbf{0})) + \sum_{i' \in I_u} \exp(s(q,i'))} \tilde{\boldsymbol{h}}_i^{(L)}, \\ s(q,i) &= (\tilde{\boldsymbol{h}}_i^\top \tanh(\boldsymbol{W}_f^\top \boldsymbol{q} + \boldsymbol{b}_f))^\top \boldsymbol{W}_h. \end{aligned} \tag{6.13}$$

Model Optimization Following ZAM [1], we jointly optimize the product retrieval task and the language modeling task. The product retrieval loss is

$$L_{PR} = -\sum_{(u,i,q)} \log P(i|u,q) = -\sum_{(u,i,q)} \log \frac{\exp(f(\boldsymbol{i}, \boldsymbol{M}_{uq}))}{\sum\limits_{i' \in C} \exp(f(\boldsymbol{i'}, \boldsymbol{M}_{uq}))}, \tag{6.14}$$

which is optimized over all triples (u,i,q) in the training set, where a triple (u,i,q) represents a product i purchased by a user u under the submitted query q. The

language modeling loss is

$$L_{LM} = -\sum_{i} \log P(T_i|i) = -\sum_{i} \sum_{w \in T_i} \log \frac{\exp(\tau(w, i))}{\sum\limits_{w' \in V} \exp(\tau(w', i))}. \tag{6.15}$$

Hence, the total loss is

$$L_{total} = L_{PR} + L_{LM} = -\sum_{(u,i,q)} \log P(i|u, q) - \sum_{i} \log P(T_i|i). \tag{6.16}$$

Remark. It is worth noting that we only use the graph-enriched product embeddings to represent users but do not use them to represent products themselves in the product retrieval task (Eq. (6.2)) or the language modeling task (Eq. (6.3)), because the mixed representations may make products lose their uniqueness and hurt performance, which is verified by our empirical study.

Since the candidate set C and vocabulary V are usually extremely large, it is impractical to compute the log likelihood in Eq. (6.14) and Eq. (6.15). A common solution is to sample only a portion of negative products to approximate the denominator of Eq. (6.14) and Eq. (6.15):

$$\begin{aligned} L = -\sum_{(u,i,q)} [&\log \tau(w, i) + k_w \mathbb{E}_{w' \sim P_w} \log \tau(-w', i) \\ &+ \log f(i, M_{uq}) + k_i \mathbb{E}_{i' \sim P_i} (\log f(-i, M_{uq}))], \end{aligned} \tag{6.17}$$

where k_w and k_i are the negative sampling rates for words and products, respectively. In this work, we follow HEM [3] to randomly sample negative words from the vocabulary with P_w as the unigram distribution raised to the 3/4rd power, and randomly sample negative products from all products in the training set.

Table 6.1: Dataset Statistics.

	Magazine	**Software**	**Phones**	**Toys&Games**
#reviews	4,583	25,086	133,792	148,756
#user	694	3,642	17,464	16,370
#query	170	999	163	399
#product	876	5,875	10,278	11,875
#seq	2,337	17,814	79,224	78,616
#edge	3,078	16,391	93,174	111,578
	Instruments	**Clothing**	**Health**	**Home&Kitchen**
# reviews	209,229	270,854	334,025	545,083
#user	23,887	37,914	36,639	65,510
#query	492	2,000	793	900
#product	9,756	23,033	17,956	27,888
#seq	100,945	160,959	201,513	333,709
#edge	149,401	189,985	256,095	408,607

6.4 Experiments

In this section, we introduce our experimental settings and present experimental results. Our experiments try to answer the following research questions:

- **RQ1:** How is the performances of SBG compared to the base model ZAM?
- **RQ2:** How does SBG perform compared to state-of-the-art methods for personalized product search?
- **RQ3:** How useful is graph convolution? Can the proposed jumping connection alleviate over-smoothing?
- **RQ4:** What is the effect of time interval R on the constructed successive behavior graph G_{SB}?

6.4.1 Experimental Setup

Datasets Our experiments are conducted on the well-known Amazon review dataset[3] [82], which includes product reviews and metadata such as product titles and categories. It was first introduced for product search by Van Gysel et al. [45, 106] and has become a benchmark dataset for evaluating product search methods as used in many recent studies [3, 1, 2, 75]. Product reviews are generally used as text corpus for representing products or users, and product categories are used as queries to simulate a search scenario.

In our experiments, we use the 5-core data, and for each sub dataset, we filter out products with no positive records. The statistics of the filtered datasets are shown in Table 6.1.

Baselines We compare our proposed approach SBG with the following baselines.

- **HEM [3]** assumes that users and products are independent. It employs PV to learn the representations of users, products, and words jointly. The user-query pair is projected to the same latent space with products.
- **ZAM [1]** also employs PV to learn semantic representations of products and words. Users are represented by the products they visited. In particular, ZAM proposes a zero attention vector to control the degree of personalization.
- **DREM [2]** employs knowledge graph embedding techniques and constructs a unified knowledge graph that represents both static entity features and dynamic user searching behaviors. Embeddings of all entities are learned via a graph regularization loss.

[3] http://jmcauley.ucsd.edu/data/amazon

Table 6.2: Comparison of our proposed method with baselines on eight Amazon review sub datasets. * and † denote the significant differences to ZAM and the best baseline, respectively in paired t-test with $p \leq 0.01$. The best results are highlighted in bold.

	Software					Magazine				
	HR@10	**MRR@100**	**NDCG@10**	**NDCG@20**	**NDCG@100**	**HR@10**	**MRR@100**	**NDCG@10**	**NDCG@20**	**NDCG@100**
HEM	0.3785	0.2181	0.2474	0.2688	0.3024	0.4115	0.2183	0.2526	0.2837	0.3211
ZAM	0.4841	0.2900	0.3289	0.3465	0.3713	0.4349	0.2290	0.2657	0.2968	0.3384
DREM	0.5058	0.3189	0.3555	0.3740	0.4029	0.3985	0.2130	0.2469	0.2725	0.3103
GraphSRRL	0.2555	0.1415	0.1598	0.1795	0.2186	0.2682	0.1353	0.1579	0.1795	0.2190
SBG (ours)	**0.5629***†	**0.3759***†	**0.4144***†	**0.4302***†	**0.4501***†	**0.4679***†	**0.2568***†	**0.2952***†	**0.3255***†	**0.3657***†
	Phones					**Toys&Games**				
	HR@10	**MRR@100**	**NDCG@10**	**NDCG@20**	**NDCG@100**	**HR@10**	**MRR@100**	**NDCG@10**	**NDCG@20**	**NDCG@100**
HEM	0.4049	0.2411	0.2697	0.2913	0.3314	0.2509	0.1330	0.1501	0.1729	0.2217
ZAM	0.5160	0.2995	0.3386	0.3659	0.4083	0.4358	0.2314	0.2653	0.2985	0.3541
DREM	0.5836	0.3365	0.3841	0.4107	0.4419	0.5557	0.3124	0.3554	0.3916	0.4368
GraphSRRL	0.1883	0.0887	0.1035	0.1225	0.1581	0.3811	0.1822	0.2166	0.2475	0.2956
SBG (ours)	**0.5878***	**0.3447***†	**0.3904***	**0.4173***	**0.4553***†	**0.571***†	**0.3182***	**0.3648***†	**0.3976***	**0.4418***
	Instruments					**Clothing**				
	HR@10	**MRR@100**	**NDCG@10**	**NDCG@20**	**NDCG@100**	**HR@10**	**MRR@100**	**NDCG@10**	**NDCG@20**	**NDCG@100**
HEM	0.4754	0.2562	0.2974	0.3262	0.3649	0.5146	0.3013	0.3409	0.3673	0.4065
ZAM	0.4951	0.2806	0.3202	0.3482	0.3911	0.5606	0.3230	0.3671	0.3972	0.4371
DREM	0.4503	0.2475	0.2856	0.3107	0.3523	0.4129	0.2460	0.2741	0.2999	0.3488
GraphSRRL	0.5073	0.2770	0.3194	0.3507	0.3951	0.1789	0.0811	0.0946	0.1152	0.1537
SBG (ours)	**0.5184***†	**0.3052***†	**0.3451***†	**0.3713***†	**0.4116***†	**0.602***†	**0.3528***†	**0.4001***†	**0.4294***†	**0.4663***†
	Health					**Home&Kitchen**				
	HR@10	**MRR@100**	**NDCG@10**	**NDCG@20**	**NDCG@100**	**HR@10**	**MRR@100**	**NDCG@10**	**NDCG@20**	**NDCG@100**
HEM	0.3841	0.2292	0.2572	0.2778	0.3153	0.4507	0.2846	0.3164	0.3349	0.3666
ZAM	0.4528	0.2622	0.2952	0.3235	0.3727	0.5247	0.3219	0.3593	0.3853	0.4275
DREM	0.5667	0.3620	0.3985	0.4276	0.4686	0.5793	0.3894	0.4236	0.4501	0.4904
GraphSRRL	0.5073	0.2770	0.3194	0.3507	0.3951	0.4534	0.2199	0.2604	0.2976	0.3534
SBG (ours)	**0.6181***†	**0.3696***†	**0.4174***†	**0.4458***†	**0.4815***†	**0.6419***†	**0.4095***†	**0.4546***†	**0.4802***†	**0.5152***†

- **GraphSRRL [75]** explicitly utilizes structural patterns in a user-query-product graph. It defines three specific structural patterns that represent three frequent user-query-product interactions.

Evaluation Protocol We partition each sub dataset into a training set, a validation set, and a test set. Similar to the process of constructing G_{SB}, we sort the reviews in chronological order for each user and split the full record into successive behavior sequences. Then, the last sequence is used as the test set, and the second last sequence is used as the validation set.

For performance measurement, we adopt three metrics: hit rate (HR@K), normalized discounted cumulative gain (NDCG@K), and mean reciprocal rank (MRR). For each sequence, the target products are mixed up with candidates

randomly sampled from the entire product set, forming a candidate set of size 1000.

Implementation Details For a fair comparison, we re-implement HEM, ZAM, and DREM using the same encoder, evaluation setting, and negative sampling method. For DREM, we build a unified heterogeneous graph that contains the user-product review relation and the product-category belonging relation. In addition, we connect products and users with their associated words during training. For GraphSRRL, we use the official implementation[4] with our dataset splits and evaluation protocols. For all methods, the batch size is set to 1024, and the ADAM optimizer is used with an initial learning rate of 0.001. All the entity embeddings are initialized randomly with dimension 64. For our SBG, we set the attention dimension d_a to 8, and the user-query balancing parameter λ to 0.5. We employ 4 layers of jumping graph convolution, and the weight of self-loop is set to 0.1. The strength of jumping connection β is also set to 0.1. The negative sampling rate for each word is set to 5, and that for each item is set to 2. We report the evaluation metrics on the converged model. The reported results are averaged over multiple runs, and the significant differences are computed based on the paired t-test with $p \leq 0.01$.

6.4.2 Result Analysis

Main Results Table 6.2 summarizes the overall performance of our SBG and the baselines on eight Amazon review sub datasets. Besides, the improvement percentages of SBG and the best baseline DREM (also graph-based) over ZAM

[4]https://github.com/Shawn-hub-hit/GraphSRRL-master

Table 6.3: The improvement percentages of NDCG@10 over ZAM by DREM (the best baseline) and our SBG.

	Magazine	**Software**	**Phones**	**Toys&Games**
DREM	+8.08%	-7.06%	+13.44%	+33.96%
SBG	+25.99%	+11.10%	+15.30%	+37.50%
	Instruments	**Clothing**	**Health**	**Home&Kitchen**
DREM	-10.83%	-25.34%	+35.00%	+17.89%
SBG	+7.76%	+8.99%	+41.39%	+26.53%

are summarized in Table 6.3. We can make the following observations.

- First of all, SBG significantly improves over ZAM in every tested domain/dataset. Since NDCG reflects the quality of the entire ranking list, we calculate the performance gain of SBG compared to ZAM in NDCG@10 as shown in Table 6.3. SBG improves ZAM by at least 7.76% on *Instruments*, and up to 41.39% on *Health*. **RQ1** is answered.

- SBG achieves significant improvements over other baselines in nearly all cases, which answers **RQ2**.

- As shown in Table 6.3, the performances of DREM and SBG are consistent on all domains. In the domains where DREM fails including *Software*, *Instruments*, and *Clothing*, the improvement percentages of SBG are also less significant. It indicates that the effectiveness of graph-based methods may be affected by the characteristics of different domains.

- GraphSRRL assumes that the pre-defined patterns are frequent, which may not hold in some domains, as evidenced by the experimental results.

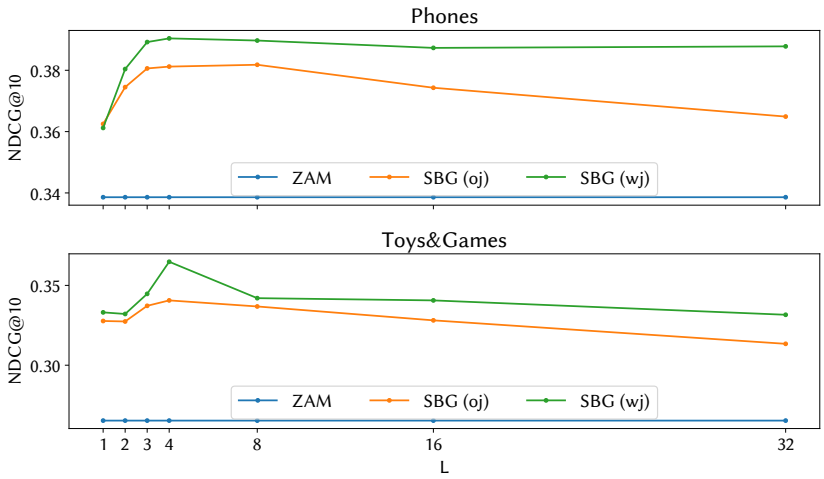


Figure 6.3: Ablation studies of jumping connections and the number L of the graph convolutional layers. The results in NDCG@10 with respect to L on *Toys&Games* and *Phones* are reported. SBG (wj) stands for our proposed SBG with jumping connections, and SBG (oj) stands for SBG without jumping connections.

Effect of the Order of Graph Convolution To answer **RQ3**, we investigate the effect of the number of graph convolution layers L and the jumping connections. We conduct experiments on *Phones* and *Toys&Games* and vary L from 0 to 64. We compare our SBG with jumping connections (denoted as SBG (wj)) with ZAM and a variant SBG (oj), which stands for SBG without jumping connections. The results are shown in Figure 6.3.

It can be seen that a few graph convolution layers can bring substantial performance gains. In our experiments on *Phones* and *Toys&Games*, we achieve the best performance with $L = 4$. However, as L increases, the performances of SBG (wj) and SBG (oj) drop due to the over-smoothing effect. Especially on *Toys&Games*, we observe a significant performance drop when L is larger than 4.

Effect of Jumping Connections. It can be seen from Figure 6.3 that as L increases, the performance of SBG (wj) drops much slower than that of SBG (oj),

especially on *Phones*, demonstrating the effectiveness of jumping connections in alleviating the over-smoothing effect.

Effect of Time Interval R. To answer **RQ4**, we evaluate the performance of SBG with respect to different time scale $R \in \{a\ day, a\ week, a\ month, a\ quater, a\ year\}$. The results shown in Figure 6.4. It can be observed that the best performance is usually achieved when R is $a\ day$ or $a\ week$. When R is $a\ month$ or longer, the performances often drop. This is probably because the behavior sequences are overly diverse and contain noisy preference signals when the time scale is large. The only exception is on *Magazine*, where the performance of SBG slightly increases as R becomes larger. However, *Magazine* is a small dataset that does not have much training data and many products have limited records in the test set. Hence, content-based product features may not be informative enough, and a larger R helps a cold product reach to broader neighborhood and leads to better performance.

Runtime Analysis We evaluate the time efficiency of our SBG and the baselines on *Home&Kitchen*, the largest dataset used in the experiments. As shown in Figure 6.5, the running time of our method is comparable with that of ZAM and DREM. All the experiments are conducted on a platform with Intel(R) Xeon(R) Gold 6226R CPU @ 2.90GHz and GeForce RTX 3090.

6.5 Chapter Summary

This chapter tackled the challenge of user preference modeling in personalized product search by introducing a novel approach that leverages both local and global

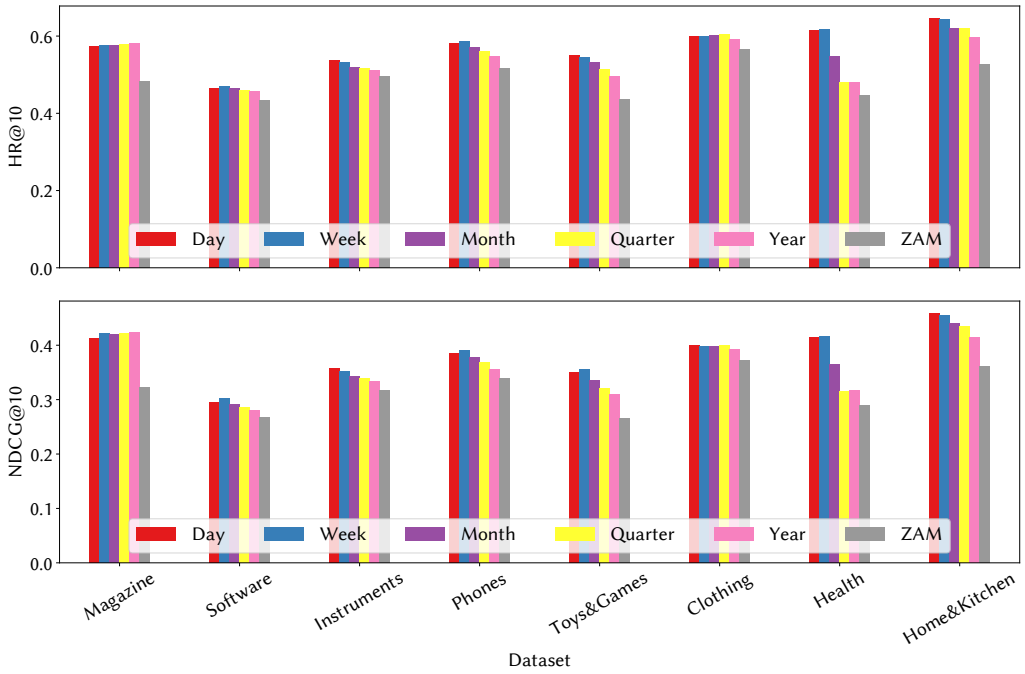


Figure 6.4: Ablation study of the time interval R for successive graph construction. The results in HR@10 and NDCG@10 are reported. R is varied from a day to a year.

user behavior patterns. Traditional latent space methods often fail to fully capture the complexity of user-product interactions, relying on limited behavioral data.

To address this, the chapter proposed constructing a user successive behavior graph (SBG) from short-term user actions, enabling the capture of richer relational and collaborative signals. An efficient jumping graph convolutional network (GCN) was integrated into the product search framework to enrich product representations while mitigating the over-smoothing problem through jumping connections.

Finally, extensive experiments on eight Amazon datasets demonstrated that this method consistently outperforms state-of-the-art baselines. Ablation studies confirmed the effectiveness of the graph convolution approach and highlighted the importance of modeling high-order relations.

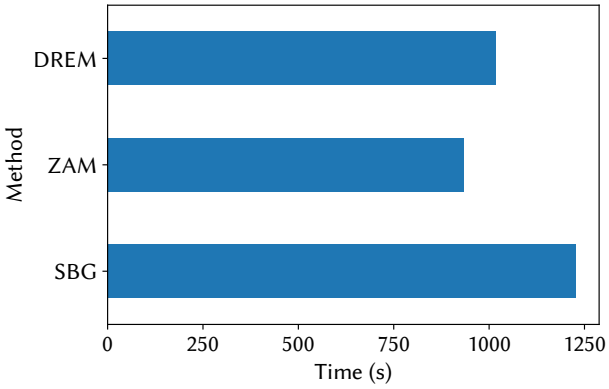


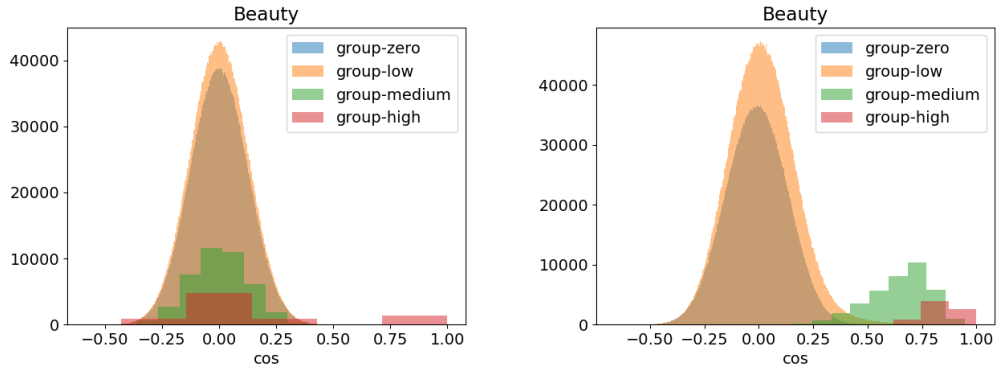
Figure 6.5: Comparison of the average training time (in seconds) on *Home&Kitchen*.

Chapter 7

Hard Negative Mining in Recommendation

7.1 Motivation

Sequential recommendation (SR) is a task that predicts the next item that a user may be interested in given his or her past behavior sequence. Most SR models use Noise Contrastive Estimation (NCE), which brings positive items closer to the sequence and pushes negative ones away. Negative sampling strategies play a crucial role in the development of successful SR models. However, while the majority of SR models adopt uniform sampling strategies and focus on designing sequence encoders or incorporating additional training tasks, a few attempts have been made to focus on mining informative negative samples for SR. These attempts can be broadly categorized into two groups: (1) sampling-based methods [22, 41, 118, 56, 124] that aim to learn a proper distribution for negative mining, and (2) generation-based methods [114, 135, 115, 54] that develop generative models



(a) (b)

Figure 7.1: Visualization of the distributions of node-pair similarities in different groups on the *Beauty* dataset at the training epoch 0 and 20, respectively. Nodes are divided into four groups w.r.t. their Jaccard similarity $J(i, j)$ on a global weighted item transition graph. Node/item pairs in *group-zero* have no common neighbors, those in *group-low* have $J(i, j) \in (0, 0.15]$, those in *group-medium* have $J(i, j) \in (0.15, 0.3]$, and those in *group-high* have $J(i, j) \in (0.3, 1.0]$. Note that due to the considerable differences in group size, we use different bin values for different groups.

that synthesize negative samples for model training. Despite the various types of negative mining methods, to our knowledge, there is no effort that explicitly utilizes structural information concealed in user behavior sequences to mine hard negatives via graph analysis for SR. In this work, we develop a new negative sampling approach for SR by utilizing the structural information of behavior sequences and creating a negative sampler based on neighborhood analysis.

Our approach is motivated by a pilot experiment in Figure 7.1, which visualizes the distributions of node-pair similarities in different groups at the different epochs during training an SR model. The nodes on a global weighted item transition graph (WITG) are divided into four groups w.r.t. the Jaccard similarity defined in Eq. 7.2 that measures the degree of neighborhood overlap between two nodes

on the graph. Specifically, the four groups are defined as: **group-zero** ($J = 0$, no neighborhood overlap), **group-low** ($0 < J \leq 0.33$, low overlap), **group-medium** ($0.33 < J \leq 0.67$, medium overlap), and **group-high** ($J > 0.67$, high overlap). Higher $J(i, j)$ indicates the neighborhoods of i and j have a larger overlap.

Figure 7.1 shows that for each group, the distribution of embedding similarity shifts as training progresses. The distributions of *group-zero* and *group-low* exhibit similar patterns that they only change slightly. The two groups represent easy negatives, which, although less informative, are essential for training the SR model. Item pairs within *group-medium* display some overlap in their respective neighborhoods. As training progresses, a significant shift in the distribution is observed. In this study, we treat item pairs in *group-medium* as hard negative pairs, positing that they contribute more information during the training process. Since item pairs in *group-high* have strong connections and are likely to be false negatives, we propose to exclude them from the item pool for negative sampling.

Specifically, we propose a *g*raph-based *n*egative sampling approach based on *n*eighborhood *o*verlap (GNNO). GNNO first constructs a WITG as described in Sec 7.3. Utilizing the built WITG, it selects negative samples for each target item by considering the extent of the neighborhood overlap between the target item and any other item. Additionally, GNNO employs curriculum learning (CL) to adjust the maximum hardness of negative samples, ranging from easy to hard. Among existing graph-based negative sampling strategies, GNNO is most similar to RecNS [127] in that we also perform region division and propose sampling variations that account for unique characteristics in each region. The key differences between GNNO and RecNS are three-fold: (1) GNNO is designed for sequential recommendation and explicitly utilizes structural information in user behavior

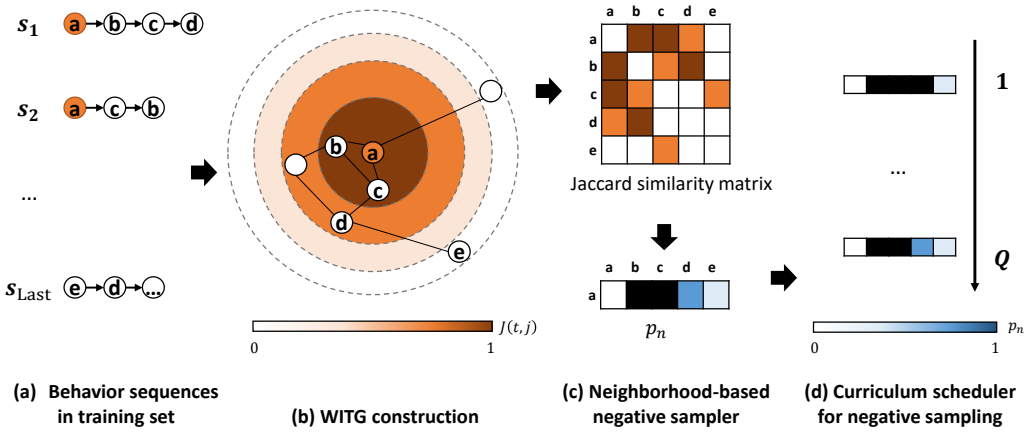


Figure 7.2: Illustration of our proposed GNNO framework. The false negatives for a given target node are annotated as black blocks. For example, for the target node a, nodes b and c are false negatives and hence excluded from the sampling distribution for a.

sequences; (2) Rather than dividing the sampling region by k-hop distance, GNNO creates a negative sampler that samples from a distribution based on the relative Jaccard index; (3) GNNO suggests negative samples from distant regions are indispensable for training an SR model.

7.2 Problem Formulation

Formally, let $\mathcal{U}$ and $\mathcal{I}$ denote the user and item sets from the interaction sequences, respectively. For each user $u \in \mathcal{U}$, we use a chronologically ordered list $\mathbf{s} = [i_1, i_2, ..., i_N]$ to denote his or her behavior sequence, where i_t is the t^{th} interaction item of u and N is the length of the sequence. The task of SR is to predict the subsequent user-item interactions with the given sequence $\mathbf{s}$.

Training an SR model requires choosing an objective function L and a negative sampling distribution p_n. The commonly adopted Bayesian Personalized Ranking

(BPR) loss [92] w.r.t. all training sequences and time steps is defined as follows:

$$L_{BPR} = \sum_{(\mathbf{s}_t, i_t)} -\log \sigma(\hat{y}(\mathbf{s}_t, i_t) - \hat{y}(\mathbf{s}_t, i_t^-)), \tag{7.1}$$

where $\mathbf{s}_t = [i_1, i_2, ..., i_{t-1}]$ is the historical sub-sequence of $\mathbf{s}$ at time step t, i_t is the target item, σ denotes the sigmoid function, $\hat{y}(\mathbf{s}, i)$ is a trainable network that predicts a matching score between a sequence $\mathbf{s}$ and an item i. $i_t^- \sim p_n$ is a negative item sampled from a distribution p_n. Optimizing Eq 7.1 offers the model ability to score the target item i_t higher than the negative item i_t^- for a given $\mathbf{s}_t$.

7.3 Method

In this section, we start with the problem statement of sequential recommendation. Subsequently, we present our proposed approach. As illustrated in Figure 7.2, GNNO comprises three modules: (1) WITG construction; (2) overlapping-based negative sampling distribution generator; and (3) curriculum scheduler for negative sampling.

Weighted Item Transition Graph (WITG) In contrast to the commonly used user-item graph, to make use of global information in behavior sequences, we propose to mine hard negatives on a WITG $\mathcal{G}(\mathcal{I}, \mathcal{E})$ [144] where $\mathcal{E}$ is the edge set. A WITG contains global item transition patterns extracted from all user behavior sequences in the training set $\mathcal{D}$. $\mathcal{G}$ is constructed by traversing every sequence in $\mathcal{D}$. For a sequence $\mathbf{s} \in \mathcal{D}$, if there exists no edge between the items i_m and $i_{(m+k)}$ in $\mathcal{G}$, we connect them and set the edge weight $w(i_m, i_{(m+k)}) = 1/k$, where

k represents the importance of a target item i_m to its k-hop neighbor $i_{(m+k)}$ in $\mathbf{s}$. Otherwise, if there is already an edge between them, we update the edge weight as $w(i_m, i_{(m+k)}) \leftarrow w(i_m, i_{(m+k)}) + 1/k$.

Neighborhood-based Negative Sampler Given a target item, we can divide the other items into groups w.r.t. their neighborhood overlap with the target item. As analyzed in Sec. 7.1, these groups demonstrate distinct characteristics during the training of a SR model, thus it can be advantageous to differentiate and treat them differently. Here, we propose to use the Jaccard similarity to measure the degree of neighborhood overlap between the target item and the others. Let $\mathcal{N}(i)$ denote the neighbor set of an item i on $\mathcal{G}$. The Jaccard similarity between any two items i and j can be defined as:

$$J(i,j) = \frac{\mathcal{N}(i) \cap \mathcal{N}(j)}{\mathcal{N}(i) \cup \mathcal{N}(j)}. \tag{7.2}$$

If i and j have many common neighbors, they will have a large similarity. Meanwhile, the denominator acts as a normalizer, and the similarity will be small if the degree of either i or j is large.

As explained in Figure 7.1, *group-medium* is likely to be quality hard negatives, hence we want to give them higher sampling weight. Meanwhile, we want to keep *group-zero* and *group-medium* for diversity, but we want to give them low sampling probability. Finally, we want to exclude *group-high* as they are likely to be false negatives. Therefore, we propose to sample the negatives for a target item i from the following distribution:

$$p_n(i^-|i) = \frac{e^{J(i,i^-)}}{\sum_{j \in \mathcal{N}'(i)} e^{J(i,j)}}, \tag{7.3}$$

where $\mathcal{N}'(i) = \mathcal{I} \backslash \{j | J(i,j) > \lambda\}$ is the item set without *group-high*, and λ is the threshold for *group-high* at the current training step.

Curriculum Scheduler for Negative Sampling Concentrating too much on hard negatives in the early training stage may bring negative effects to the model [16]. We therefore employ curriculum learning (CL) techniques to schedule the hardness of the negatives. The main idea of CL is to order negative samples during training based on their difficulty [116]. Let Q be the maximum training step, then for each training step $q \in [1, ..., Q]$, we update λ to form $\mathcal{N}'(i)$ in Eq. 7.3. The maximum Jaccard score is updated from an initial hardness b with a linear pacing function $f(q)$:

$$f(q) = c * q + b, \tag{7.4}$$

where q is the current time step and c is the pace coefficient. For simplicity, we set $b = 0$ in our experiments. Moreover, we set a maximum hardness λ_{max} to clip the growth of λ. Finally, λ is updated as:

$$\lambda(q) = min(f(q), \lambda_{max}). \tag{7.5}$$

7.4 Experiments

7.4.1 Experimental Setup

Datasets Our experiments are conducted on three subsets of the well-known Amazon dataset[1] [82], which includes rich user-item review interactions. Specifi-

[1] http://jmcauley.ucsd.edu/data/amazon

Table 7.1: Dataset statistics.

	#user	#item	#entry	#edge	sparseness
Beauty	22,363	12,101	198,502	530,266	20.01%
Toys	19,413	11,925	148,455	486,740	16.32%
Phone	22,364	12,102	176,139	530,266	24.55%

Table 7.2: Hyper-parameter settings. $\#neg_{hard}$ and $\#neg_{rand}$ indicate the number of hard negatives and random negatives, respectively.

	$\#neg_{hard}$	$\#neg_{rand}$	c (Eq.7.4)	λ_{max} (Eq.7.5)
Beauty	9	16	0.04	0.5
Toys	2	10	0.05	0.2
Phones	4	10	0.01	0.9

cally, we choose the sub datasets *Beauty, Phone, and Toy* for our empirical study. In our experiments, we use the 5-core data. Items and users with no positive records are filtered out for each subset. The statistics of the filtered datasets are shown in Table 7.1.

Baselines We compare our method with both state-of-the-art sequential recommendation methods and negative sampling strategies. As shown in Table 7.4, we compare our method with the following baselines for sequential recommendation: **BPRMF** [92], **Caser** [105], **GRU4Rec** [50], **BERT4Rec** [103], **SASRec** [55], **TimiRec** [112], and **ContraRec** [111]. Meanwhile, as shown in Table 7.3, we also compare our method with the following negative sampling approaches: **DNS** [142] that adaptively samples the negative item scored highest by the recommender, **MCNS** [128]that samples negatives with the distribution sub-linearly related to the positive distribution and accelerates the sampling process by Metropolis-Hastings, and **MixGCF** [54] that injects positive samples into negatives via hop mixing to

Table 7.3: Comparison of our proposed method with hard negative mining baselines on three Amazon review sub datasets. The best results are highlighted in bold.

Dataset	Method	HR@5	NDCG@5	HR@20	NDCG@20
Beauty	**DNS**	0.3858	0.2984	0.5890	0.3561
	MCNS	0.4101	0.3141	0.6135	0.3721
	MixGCF	0.4123	0.3168	0.6140	0.3743
	GNNO	**0.4149**	**0.3215**	**0.6174**	**0.3792**
Phone	**DNS**	0.4307	0.3300	0.6563	0.3943
	MCNS	0.4794	0.3646	0.7179	0.4334
	MixGCF	0.4810	0.3655	0.7197	0.4342
	GNNO	**0.4897**	**0.3754**	**0.7249**	**0.4430**
Toy	**DNS**	0.3470	0.2666	0.5598	0.3266
	MCNS	0.4035	0.3084	0.6238	0.3711
	MixGCF	0.4049	0.3097	**0.6252**	0.3723
	GNNO	**0.4074**	**0.3148**	0.6245	**0.3762**

synthesize hard negatives. Note that MCNS and MixGCF are graph-based but DNS is not.

Implementation details All recommendation baselines are implemented with ReChorus[2], a framework for top-k recommendation. GNNO and other negative sampling strategies are implemented on the state-of-the-art SR method ContraRec. We use the default settings of ContraRec. Specifically, we use Adam as the optimizer and BERT4Rec as the sequence encoder. The batch size is set to 4096, and the dimension of hidden units is set to 64. The hyper-parameters of GNNO, as shown in Table 7.2, are selected based on the performances on the validation set using grid search.

Evalutaion Protocals Following [111], we adopt the commonly used leave-one-out strategy to evaluate model performance. We employ two evaluation metrics:

[2]https://github.com/THUwangcy/ReChorus

hit rate (HR@K) and **normalized discounted cumulative gain (NDCG@K)**. For each sequence, the target products are mixed up with candidates randomly sampled from the entire product set, forming a candidate set of size 1000.

7.4.2 Result Analysis

Comparisons with Negative Sampling Baselines Table 7.3 summarizes the recommendation performance of GNNO in comparison to existing negative sampling approaches on three Amazon benchmarks. GNNO outperforms the baselines in almost all cases, demonstrating its efficacy. It can be seen that all graph-based methods perform better than DNS, which highlights the advantages of incorporating structural information for negative sampling.

Comparisons with Baselines for Sequential Recommendation Table 7.4 shows the recommendation performance of GNNO against state-of-the-art methods for SR on three Amazon benchmarks. GNNO improves over ContraRec on every dataset. It demonstrates the effectiveness of the proposed negative sampling strategy, showing that pushing away negatives with some neighborhood overlap is beneficial for SR. In addition, both ContraRec and GNNO outperform other baselines consistently, where the performance gain is probably brought by the context-context contrastive learning task.

7.5 Chapter Summary

This chapter investigates the critical role of negative sampling in sequential recommendation (SR) and introduces a novel approach that leverages structural infor-

mation embedded in user behavior sequences. The key observation underpinning this work is that, during the training of SR models, the distribution of embedding similarities between item pairs varies significantly depending on the degree of neighborhood overlap within a global weighted item transition graph (WITG). Specifically, item pairs with medium neighborhood overlap emerge as informative hard negatives, while those with high overlap are likely to be false negatives and should be excluded from negative sampling.

Building on this insight, the chapter proposes the Graph-based Negative sampling approach based on Neighborhood Overlap (GNNO). The GNNO framework first constructs a WITG to capture global item transition patterns from user sequences. It then employs the Jaccard similarity to quantify the neighborhood overlap between items, enabling the adaptive selection of negative samples. To further enhance training efficacy and mitigate the risks associated with static hard negative sampling, GNNO integrates curriculum learning to dynamically adjust the hardness of negative samples throughout the training process.

Extensive experiments conducted on three Amazon benchmark datasets demonstrate that GNNO consistently outperforms existing negative sampling strategies and state-of-the-art SR models. The empirical results validate the effectiveness of incorporating structural information for hard negative mining and highlight the benefits of curriculum learning in managing sample hardness. Collectively, this chapter advances the understanding of negative sampling in sequential recommendation by elucidating the importance of neighborhood structure and proposing a principled, graph-based sampling methodology.

Table 7.4: Comparison of our proposed method with baselines for sequential recommendation on three Amazon review sub-datasets. The best results are highlighted in bold.

Dataset	Method	HR@5	NDCG@5	HR@20	NDCG@20
Beauty	**BPRMF**	0.3588	0.2593	0.5716	0.3202
	Caser	0.3198	0.2238	0.5772	0.2970
	GRU4Rec	0.3254	0.2318	0.5762	0.3030
	BERT4Rec	0.3590	0.2658	0.5734	0.3266
	SASRec	0.3653	0.2780	0.5744	0.3372
	TimRec	0.3781	0.2812	0.5958	0.3433
	ContraRec	0.4112	0.3158	0.6111	0.3727
	GNNO	**0.4149**	**0.3215**	**0.6174**	**0.3792**
Phone	**BPRMF**	0.3690	0.2709	0.5945	0.3352
	Caser	0.3873	0.2745	0.6727	0.3560
	GRU4Rec	0.4122	0.2973	0.6935	0.3777
	BERT4Rec	0.4209	0.3145	0.6664	0.3849
	SASRec	0.4432	0.3349	0.6809	0.4032
	TimRec	0.4338	0.3241	0.6712	0.3925
	ContraRec	0.4831	0.3673	0.7210	0.4358
	GNNO	**0.4897**	**0.3754**	**0.7249**	**0.4430**
Toys	**BPRMF**	0.3107	0.2255	0.5255	0.2864
	Caser	0.2921	0.2000	0.5584	0.2757
	GRU4Rec	0.3113	0.2168	0.5802	0.2934
	BERT4Rec	0.3457	0.2561	0.5582	0.3164
	SASRec	0.3594	0.2731	0.5694	0.3328
	TimRec	0.3535	0.2628	0.5849	0.3287
	ContraRec	0.4013	0.3065	0.6177	0.3676
	GNNO	**0.4074**	**0.3148**	**0.6245**	**0.3762**

Chapter 8

Conclusions and Future Work

8.1 Conclusions

We are now moving close to the end of the thesis. In this chapter, we first summarize the core contributions of this work to the two key research directions: (a) user intent analysis for conversation and (b) user behavior modeling for recommendation. Beyond summarizing contributions, this chapter also points out the limitations of the current work. Finally, building on the identified limitation and the broader trends in conversational AI and recommender systems, we will outline future work directions that extend the scope of this research.

User Intent Analysis for Conversation Understanding user intent is the cornerstone of enabling conversational AI systems to deliver accurate and adaptive services, yet challenges such as emerging unknown intents, polysemy, and resource-intensive model deployment persist. To address these, this thesis advances the field through three progressive studies under different settings.

First, for unknown intent detection, we proposed the Semantic-Enhanced Gaussian Mixture Model (SEG). Unlike traditional softmax-based classifiers, SEG integrates semantic information from class labels into Gaussian mixture distributions, forcing embeddings to form dense, cluster-like structures in the feature space. This design makes it highly compatible with density-based outlier detection algorithms (e.g., LOF). Experiments on benchmark datasets (SNIPS, ATIS, SMP-2018) demonstrated that SEG outperforms conventional baselines (e.g., MSP, DOC, LMCL), validating the effectiveness of semantic enhancement and Gaussian mixture modeling for unknown intent identification. Second, to improve zero-shot intent classification, we reconstructed capsule networks into ReCapsNet-ZS. This model introduces a dimensional attention mechanism to resolve polysemy (by assigning context-aware weights to different dimensions of word embeddings) and a Mahalanobis distance-based strategy to construct transformation matrices for unknown intents. In both zero-shot and generalized zero-shot scenarios, ReCapsNet-ZS outperforms state-of-the-art methods, particularly addressing the generalization limitation of prior capsule network-based approaches in recognizing unknown intents. Third, to balance the powerful generalization of large language models (LLMs) and the efficiency of real-world deployment, we proposed LANID (LLM-Assisted New Intent Discovery). LANID distills semantic knowledge from LLMs into lightweight encoders via contrastive learning, using KNN-based and DBSCAN-based sampling strategies to select informative utterance pairs. Across datasets like Banking, StackOverflow, and M-CID, LANID achieved better performance than existing methods in both semi-supervised and unsupervised settings, proving its ability to discover new intents in low-resource environments.

User Behavior Modeling for Recommendation In the domain of recommender systems, this thesis addresses the challenge of modeling complex and evolving user behaviors to enhance recommendation quality. We proposed graph-based approaches that capture both local and global behavioral patterns. Specifically, we constructed user successive behavior graphs (SBG) that aggregate short-term actions across all users, forming a global structure rich in relational information. By applying efficient jumping graph convolutional networks to these graphs, we learned enriched product representations that encapsulate both direct and high-order collaborative signals. Extensive experiments on Amazon review datasets demonstrate that our SBG-based approach consistently outperforms strong baselines, including ZAM, DREM, and GraphSRRL, with significant improvements in NDCG and hit rate metrics across diverse domains.

Additionally, we tackled the sequential recommendation task by introducing GNNO, a graph-based negative sampling strategy that leverages the neighborhood overlap in a global weighted item transition graph (WITG) to identify informative hard negatives. By employing curriculum learning to balance the hardness of negatives during training, GNNO enhances the discriminative power of recommendation models. Our results show that GNNO outperforms existing negative sampling methods and state-of-the-art sequential recommendation models, highlighting the value of structural information in user-item interactions.

Through these contributions, our work demonstrates the critical role of both user intent and user behavior modeling in advancing the intelligence, adaptability, and user-centricity of modern dialogue and recommendation systems.

8.2 Limitations of This Thesis

Despite the progress made in this thesis, there are still notable limitations. Most of the proposed methods, including intent analysis and user behavior modeling, are fundamentally based on relatively small-scale models or lightweight architectures. While these approaches are efficient and practical for many real-world applications, the landscape of natural language processing has been fundamentally transformed by the advent of large language models (LLMs). LLMs possess strong zero-shot capabilities and a remarkable ability to understand complex and nuanced information, which has made them the new paradigm in NLP research and applications. Although this thesis has made initial explorations in Chap 4.6, LANID leverages LLMs to guide the training of small models for NID problem. However, the broader question of how to fully utilize LLMs for intent analysis and user behavior modeling, while overcoming their high computational cost and deployment complexity, remains largely open. Effectively integrating the strengths of LLMs without incurring prohibitive resource requirements is a key challenge for future research.

8.3 Future Work: Harnessing LLMs for Personalized User Understanding through Historical Data Reorganization and Compression

In the context of rapidly evolving technological advancements, a promising future direction is to explore how to more deeply and efficiently integrate LLMs into both

user intent analysis and user behavior modeling. By systematically extracting, condensing, and synthesizing relevant information from these longitudinal interactions, LLMs can enable a more nuanced and personalized understanding of each user. This, in turn, paves the way for delivering tailored services that are not only more accurate but also dynamically adaptive to evolving user preferences and contexts.

Chapter 9

References

[1] Qingyao Ai et al. “A Zero Attention Model for Personalized Product Search”. In: *Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, Beijing, China, November 3-7, 2019*. ACM, 2019, pp. 379–388.

[2] Qingyao Ai et al. “Explainable Product Search with a Dynamic Relation Embedding Model”. In: *ACM Trans. Inf. Syst.* 38.1 (2020), 4:1–4:29.

[3] Qingyao Ai et al. “Learning a Hierarchical Embedding Model for Personalized Product Search”. In: *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Shinjuku, Tokyo, Japan, August 7-11, 2017*. ACM, 2017, pp. 645–654.

[4] Wenbin An et al. “New user intent discovery with robust pseudo label training and source domain joint training”. In: *IEEE Intelligent Systems* 38.4 (2023), pp. 21–31.

[5] Abhinav Arora et al. “Cross-lingual transfer learning for intent detection of covid-19 utterances”. In: (2020).

[6] Lei Jimmy Ba et al. “Predicting Deep Zero-Shot Convolutional Neural Networks Using Textual Descriptions”. In: *IEEE International Conference on Computer Vision (ICCV)*. 2015, pp. 4247–4255.

[7] Ismail Badache. “Exploring Differences in the Impact of Users’ Traces on Arabic and English Facebook Search”. In: *2019 IEEE/WIC/ACM International Conference on Web Intelligence, WI 2019, Thessaloniki, Greece, October 14-17, 2019*. 2019, pp. 225–232.

[8] Ismail Badache and Mohand Boughanem. “Fresh and Diverse Social Signals: Any Impacts on Search?” In: *Proceedings of the 2017 Conference on Conference Human Information Interaction and Retrieval, CHIIR 2017, Oslo, Norway, March 7-11, 2017*. 2017.

[9] Keping Bi, Qingyao Ai, and W. Bruce Croft. “A Transformer-based Embedding Model for Personalized Product Search”. In: *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*. ACM, 2020, pp. 1521–1524.

[10] Keping Bi, Qingyao Ai, and W. Bruce Croft. “Learning a Fine-Grained Review-based Transformer Model for Personalized Product Search”. In: *SIGIR ’21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021*. ACM, 2021, pp. 123–132.

[11] Keping Bi et al. “Leverage Implicit Feedback for Context-aware Product Search”. In: *Proceedings of the SIGIR 2019 Workshop on eCommerce, co-located with the 42st International ACM SIGIR Conference on Research and Development in Information Retrieval, eCom@SIGIR 2019, Paris, France, July 25, 2019*. Vol. 2410. CEUR Workshop Proceedings. CEUR-WS.org, 2019.

[12] Piotr Bojanowski et al. “Enriching Word Vectors with Subword Information”. In: *Transactions of the Association for Computational Linguistics (TACL)* 5 (2017), pp. 135–146.

[13] Markus M Breunig et al. “LOF: identifying density-based local outliers”. In: *ACM SIGMOD RECORD*. Vol. 29. 2. 2000, pp. 93–104.

[14] Xuxiao Bu, Jihua Zhu, and Xueming Qian. “Personalized product search based on user transaction history and hypergraph learning”. In: *Multim. Tools Appl.* 79.31-32 (2020), pp. 22157–22175.

[15] Inigo Casanueva et al. “Efficient Intent Detection with Dual Sentence Encoders”. In: *ACL 2020* (2020), p. 38.

[16] Thibault Castells, Philippe Weinzaepfel, and Jérôme Revaud. “SuperLoss: A Generic Loss for Robust Curriculum Learning”. In: *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*. 2020.

[17] Ajay Chatterjee and Shubhashis Sengupta. “Intent mining from past conversations for conversational agent”. In: *arXiv preprint arXiv:2005.11014* (2020).

[18] Sergiu Chelaru, Claudia Orellana-Rodriguez, and Ismail Sengör Altingövde. "How useful is social feedback for learning to rank YouTube videos?" In: *World Wide Web* (2014).

[19] Ming Chen et al. "Simple and Deep Graph Convolutional Networks". In: *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*. Vol. 119. Proceedings of Machine Learning Research. PMLR, 2020, pp. 1725–1735.

[20] Yun-Nung Chen, Asli Çelikyilmaz, and Dilek Hakkani-Tür. "Deep Learning for Dialogue Systems". In: *Annual Meeting of the Association for Computational Linguistics (ACL)*. 2017, pp. 8–14.

[21] Yun-Nung Chen, Dilek Z. Hakkani-Tür, and Xiaodong He. "Zero-shot learning of intent embeddings for expansion by convolutional deep structured semantic models". In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2016, pp. 6045–6049.

[22] Kyunghyun Cho et al. "Learning Phrase Rep resentations using RNN Encoder-Decoder for Statistical Machine Translation". In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*. ACL, 2014, pp. 1724–1734. DOI: 10.3115/v1/d14-1179. URL: https://doi.org/10.3115/v1/d14-1179.

[23] Fan RK Chung and Fan Chung Graham. *Spectral graph theory*. 92. American Mathematical Soc., 1997.

[24] Alice Coucke et al. “Snips Voice Platform: an embedded Spoken Language Understanding system for private-by-design voice interfaces”. In: *arXiv* abs/1805.10190 (2018).

[25] Thomas Cover and Peter Hart. “Nearest neighbor pattern classification”. In: *IEEE transactions on information theory* 13.1 (1967), pp. 21–27.

[26] Nick Craswell and Martin Szummer. “Random walks on the click graph”. In: *SIGIR 2007: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Amsterdam, The Netherlands, July 23-27, 2007*. ACM, 2007, pp. 239–246.

[27] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. “Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering”. In: *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*. 2016, pp. 3837–3845.

[28] Jingtao Ding et al. “Simplify and Robustify Negative Sampling for Implicit Collaborative Filtering”. In: *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*. 2020.

[29] Chi Thang Duong et al. “Graph embeddings for one-pass processing of heterogeneous queries”. In: *2020 IEEE 36th International Conference on Data Engineering (ICDE)*. IEEE. 2020, pp. 1994–1997.

[30] Martin Ester et al. “A density-based algorithm for discovering clusters in large spatial databases with noise.” In: *kdd*. Vol. 96. 1996, pp. 226–231.

[31] Pablo A Estévez et al. “Normalized mutual information feature selection”. In: *IEEE Transactions on neural networks* 20.2 (2009), pp. 189–201.

[32] Lu Fan et al. “LANID: LLM-assisted New Intent Discovery”. In: *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*. 2024, pp. 10110–10116.

[33] Lu Fan et al. “Unknown intent detection using Gaussian mixture model with an application to zero-shot intent classification”. In: *Proceedings of the 58th annual meeting of the association for computational linguistics*. 2020, pp. 1050–1060.

[34] Geli Fei and Bing Liu. “Breaking the Closed World Assumption in Text Classification”. In: *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*. 2016, pp. 506–514.

[35] Emmanuel Ferreira, Bassam Jabaian, and Fabrice Lefèvre. “Online adaptative zero-shot learning spoken language understanding using word-embedding”. In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2015, pp. 5321–5325.

[36] Emmanuel Ferreira, Bassam Jabaian, and Fabrice Lefèvre. “Zero-shot semantic parser for spoken language understanding”. In: *Annual Conference of the International Speech Communication Association (INTERSPEECH)*. 2015, pp. 1403–1407.

[37] Chelsea Finn, Pieter Abbeel, and Sergey Levine. “Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks”. In: *International Conference on Machine Learning (ICML)*. 2017, pp. 1126–1135.

[38] Andrea Frome et al. “DeViSE: A Deep Visual-Semantic Embedding Model”. In: *Advances in Neural Information Processing Systems (NeurIPS)*. 2013, pp. 2121–2129.

[39] Jianfeng Gao et al. “Smoothing clickthrough data for web search ranking”. In: *Proceedings of the 32nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2009, Boston, MA, USA, July 19-23, 2009*. ACM, 2009, pp. 355–362.

[40] Ruiying Geng et al. “Few-Shot Text Classification with Induction Network”. In: *Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 2019.

[41] Dan Gillick et al. “Learning Dense Representations for Entity Retrieval”. In: *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*. 2019, pp. 528–537.

[42] Aditya Grover and Jure Leskovec. “node2vec: Scalable Feature Learning for Networks”. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*. ACM, 2016, pp. 855–864.

[43] Yangyang Guo et al. “Attentive long short-term preference modeling for personalized product search”. In: *ACM Transactions on Information Systems (TOIS)* 37.2 (2019), pp. 1–27.

[44] Yangyang Guo et al. “Multi-modal preference modeling for product search”. In: *Proceedings of the 26th ACM international conference on Multimedia*. 2018, pp. 1865–1873.

[45] Christophe Van Gysel, Maarten de Rijke, and Evangelos Kanoulas. “Learning Latent Vector Spaces for Product Search”. In: *Proceedings of the 25th ACM International Conference on Information and Knowledge Management, CIKM 2016, Indianapolis, IN, USA, October 24-28, 2016*. ACM, 2016, pp. 165–174.

[46] Will Hamilton, Zhitao Ying, and Jure Leskovec. “Inductive representation learning on large graphs”. In: *NeurIPS*. 2017, pp. 1024–1034.

[47] Michael Heck et al. “ChatGPT for Zero-shot Dialogue State Tracking: A Solution or an Opportunity?” In: *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. 2023, pp. 936–950.

[48] Charles T. Hemphill, John J. Godfrey, and George R. Doddington. “The ATIS Spoken Language Systems Pilot Corpus”. In: *Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania, USA, June 24-27, 1990*. Morgan Kaufmann, 1990.

[49] Dan Hendrycks and Kevin Gimpel. “A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks”. In: *International Conference on Learning Representations (ICLR)*. 2017.

[50] Balázs Hidasi et al. “Session-based Recommendations with Recurrent Neural Networks”. In: *4th International Conference on Learning Repre-*

sentations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings. 2016.

[51] Sepp Hochreiter and Jürgen Schmidhuber. "Long Short-Term Memory". In: *Neural Computation* 9.8 (1997), pp. 1735–1780.

[52] Yen-Chang Hsu et al. "Multi-class classification without multi-class labels". In: *arXiv preprint arXiv:1901.00544* (2019).

[53] Jian Hu et al. "Understanding user's query intent with wikipedia". In: *International Conference on World Wide Web (WWW)*. 2009, pp. 471–480.

[54] Tinglin Huang et al. "MixGCF: An Improved Training Method for Graph Neural Network-based Recommender Systems". In: *KDD '21: The 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, Singapore, August 14-18, 2021*. ACM, 2021, pp. 665–674.

[55] Wang-Cheng Kang and Julian McAuley. "Self-attentive sequential recommendation". In: *2018 IEEE international conference on data mining (ICDM)*. IEEE. 2018, pp. 197–206.

[56] Vladimir Karpukhin et al. "Dense Passage Retrieval for Open-Domain Question Answering". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*. Association for Computational Linguistics, 2020, pp. 6769–6781.

[57] Siwon Kim et al. "Propile: Probing privacy leakage in large language models". In: *Advances in Neural Information Processing Systems* 36 (2023), pp. 20750–20762.

[58] Diederik P. Kingma and Jimmy Ba. "Adam: A Method for Stochastic Optimization". In: *International Conference on Learning Representations (ICLR)*. 2015.

[59] Thomas N Kipf and Max Welling. "Semi-supervised classification with graph convolutional networks". In: *International Conference on Learning Representations*. 2017.

[60] Anjishnu Kumar et al. "Zero-Shot Learning Across Heterogeneous Overlapping Domains". In: *Annual Conference of the International Speech Communication Association (INTERSPEECH)*. 2017, pp. 2914–2918.

[61] Amy Nicole Langville and Carl Dean Meyer. *Google's PageRank and beyond - the science of search engine rankings*. Princeton University Press, 2006. ISBN: 978-0-691-12202-1.

[62] Hugo Larochelle, Dumitru Erhan, and Yoshua Bengio. "Zero-data Learning of New Tasks". In: *AAAI Conference on Artificial Intelligence (AAAI)*. 2008, pp. 646–651.

[63] Stefan Larson et al. "An Evaluation Dataset for Intent Classification and Out-of-Scope Prediction". In: *Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 2019.

[64] Quoc Le and Tomas Mikolov. "Distributed representations of sentences and documents". In: *International conference on machine learning*. PMLR. 2014, pp. 1188–1196.

[65] Fei-Fei Li, Robert Fergus, and Pietro Perona. "One-Shot Learning of Object Categories". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28.4 (2006), pp. 594–611.

[66] Qimai Li, Zhichao Han, and Xiao-Ming Wu. “Deeper Insights Into Graph Convolutional Networks for Semi-Supervised Learning”. In: *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*. AAAI Press, 2018, pp. 3538–3545.

[67] Qimai Li et al. “Label efficient semi-supervised learning via graph filtering”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 9582–9591.

[68] Shen Li et al. “Analogical Reasoning on Chinese Morphological and Semantic Relations”. In: *Annual Meeting of the Association for Computational Linguistics (ACL)*. 2018, pp. 138–143.

[69] Xiangsheng Li et al. “Learning Better Representations for Neural Information Retrieval with Graph Information”. In: *CIKM ’20: The 29th ACM International Conference on Information and Knowledge Management, Virtual Event, Ireland, October 19-23, 2020*. ACM, 2020, pp. 795–804.

[70] Ting-En Lin and Hua Xu. “Deep Unknown Intent Detection with Margin Loss”. In: *arXiv preprint arXiv:1906.00434* (2019).

[71] Ting-En Lin, Hua Xu, and Hanlei Zhang. “Discovering new intents via constrained deep adaptive clustering with cluster refinement”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 34. 2020, pp. 8360–8367.

[72] Bing Liu and Ian Lane. "Attention-Based Recurrent Neural Network Models for Joint Intent Detection and Slot Filling". In: *Annual Conference of the International Speech Communication Association (INTERSPEECH)*. 2016, pp. 685–689.

[73] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. "Isolation forest". In: *IEEE International Conference on Data Mining (ICDM)*. 2008, pp. 413–422.

[74] Han Liu et al. "Reconstructing Capsule Networks for Zero-shot Intent Classification". In: *Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 2019, pp. 4798–4808.

[75] Shang Liu et al. "Structural Relationship Representation Learning with Graph Embedding for Personalized Product Search". In: *CIKM '20: The 29th ACM International Conference on Information and Knowledge Management, Virtual Event, Ireland, October 19-23, 2020*. ACM, 2020, pp. 915–924.

[76] Yukun Ma, Erik Cambria, and Sa Gao. "Label embedding for zero-shot fine-grained named entity typing". In: *International Conference on Computational Linguistics (COLING)*. 2016, pp. 171–180.

[77] Tomas Mikolov et al. "Efficient Estimation of Word Representations in Vector Space". In: *International Conference on Learning Representations (ICLR)*. 2013.

[78] Tomás Mikolov et al. "Efficient Estimation of Word Representations in Vector Space". In: *1st International Conference on Learning Representa-*

tions, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings. 2013.

[79] Nikhil Mishra et al. "A Simple Neural Attentive Meta-Learner". In: *International Conference on Learning Representations (ICLR)*. 2018.

[80] Jinseok Nam, Eneldo Loza Menc'ia, and Johannes Fürnkranz. "All-in Text: Learning Document, Label, and Word Representations Jointly". In: *AAAI Conference on Artificial Intelligence (AAAI)*. 2016, pp. 1948–1954.

[81] Thanh V. Nguyen, Nikhil Rao, and Karthik Subbian. "Learning Robust Models for e-Commerce Product Search". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*. Association for Computational Linguistics, 2020, pp. 6861–6869.

[82] Jianmo Ni, Jiacheng Li, and Julian McAuley. "Justifying recommendations using distantly-labeled reviews and fine-grained aspects". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. 2019, pp. 188–197.

[83] Xichuan Niu et al. "A Dual Heterogeneous Graph Attention Network to Improve Long-Tail Performance for Shop Search in E-Commerce". In: *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2020, pp. 3405–3415.

[84] OpenAI. *GPT-4 Technical Report*. 2023. arXiv: 2303.08774 [cs.CL].

[85] Mark Palatucci et al. “Zero-shot Learning with Semantic Output Codes”. In: *Advances in Neural Information Processing Systems (NeurIPS)*. 2009, pp. 1410–1418.

[86] Wenbo Pan et al. “A preliminary evaluation of chatgpt for zero-shot dialogue understanding”. In: *arXiv preprint arXiv:2304.04256* (2023).

[87] Hugh Perkins and Yi Yang. “Dialog intent induction with deep multi-view clustering”. In: *arXiv preprint arXiv:1908.11487* (2019).

[88] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. “DeepWalk: online learning of social representations”. In: *The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD ’14, New York, NY, USA - August 24 - 27, 2014*. ACM, 2014, pp. 701–710.

[89] Jiashu Pu et al. “Dialog intent induction via density-based deep clustering ensemble”. In: *arXiv preprint arXiv:2201.06731* (2022).

[90] Yuanyuan Qi et al. “CGTR: Convolution Graph Topology Representation for Document Ranking”. In: *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 2020, pp. 2173–2176.

[91] Suman Ravuri and Andreas Stolcke. “Recurrent Neural Network and LSTM Models for Lexical Utterance Classification”. In: *Annual Conference of the International Speech Communication Association (INTERSPEECH)*. 2015, pp. 135–139.

[92] Steffen Rendle et al. “BPR: Bayesian Personalized Ranking from Implicit Feedback”. In: *UAI 2009, Proceedings of the Twenty-Fifth Conference on*

Uncertainty in Artificial Intelligence, Montreal, QC, Canada, June 18-21, 2009. AUAI Press, 2009, pp. 452–461.

[93] Anthony Rios and Ramakanth Kavuluru. "Few-Shot and Zero-Shot Multi-Label Learning for Structured Label Spaces". In: *Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 2018, pp. 3132–3142.

[94] Peter J Rousseeuw and Katrien Van Driessen. "A fast algorithm for the minimum covariance determinant estimator". In: *Technometrics* 41.3 (1999), pp. 212–223.

[95] Sara Sabour, Nicholas Frosst, and Geoffrey E. Hinton. "Dynamic Routing Between Capsules". In: *Advances in Neural Information Processing Systems (NeurIPS)*. 2017, pp. 3859–3869.

[96] Adam Santoro et al. "Meta-Learning with Memory-Augmented Neural Networks". In: *International Conference on Machine Learning (ICML)*. 2016, pp. 1842–1850.

[97] Bernhard Schölkopf et al. "Estimating the support of a high-dimensional distribution". In: *Neural Computation* 13.7 (2001), pp. 1443–1471.

[98] Tao Shen et al. "DiSAN: Directional Self-Attention Network for RNN/CNN-Free Language Understanding". In: *AAAI Conference on Artificial Intelligence (AAAI)*. 2018, pp. 5446–5455.

[99] Chen Shi et al. "Auto-dialabel: Labeling dialogue data with unsupervised learning". In: *Proceedings of the 2018 conference on empirical methods in natural language processing*. 2018, pp. 684–689.

[100] Lei Shu, Hu Xu, and Bing Liu. "Doc: Deep open classification of text documents". In: *arXiv preprint arXiv:1709.08716* (2017).

[101] Jake Snell, Kevin Swersky, and Richard S. Zemel. "Prototypical Networks for Few-shot Learning". In: *Advances in Neural Information Processing Systems (NeurIPS)*. 2017, pp. 4077–4087.

[102] Richard Socher et al. "Zero-Shot Learning Through Cross-Modal Transfer". In: *Advances in Neural Information Processing Systems (NIPS)*. 2013, pp. 935–943.

[103] Fei Sun et al. "BERT4Rec: Sequential Recommendation with Bidirectional Encoder Representations from Transformer". In: *Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, Beijing, China, November 3-7, 2019*. ACM, 2019, pp. 1441–1450.

[104] Flood Sung et al. "Learning to Compare: Relation Network for Few-Shot Learning". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018, pp. 1199–1208.

[105] Jiaxi Tang and Ke Wang. "Personalized Top-N Sequential Recommendation via Convolutional Sequence Embedding". In: *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, WSDM 2018, Marina Del Rey, CA, USA, February 5-9, 2018*. ACM, 2018, pp. 565–573.

[106] Christophe Van Gysel, Maarten de Rijke, and Evangelos Kanoulas. "Semantic Entity Retrieval Toolkit". In: *SIGIR 2017 Workshop on Neural Information Retrieval (Neu-IR'17)*. 2017.

[107] Sappadla Prateek Veeranna et al. “Using semantic similarity for multi-label zero-shot classification of text documents”. In: *Proceeding of european symposium on artificial neural networks, computational intelligence and machine learning. bruges, belgium: Elsevier*. 2016, pp. 423–428.

[108] Petar Velickovic et al. “Graph Attention Networks”. In: *ICLR*. 2018.

[109] Oriol Vinyals et al. “Matching Networks for One Shot Learning”. In: *Advances in Neural Information Processing Systems (NeurIPS)*. 2016, pp. 3630–3638.

[110] Weitao Wan et al. “Rethinking feature distribution for loss functions in image classification”. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018, pp. 9117–9126.

[111] Chenyang Wang et al. “Sequential Recommendation with Multiple Contrast Signals”. In: *ACM Trans. Inf. Syst.* 41.1 (2023), 11:1–11:27.

[112] Chenyang Wang et al. “Target Interest Distillation for Multi-Interest Recommendation”. In: *Proceedings of the 31st ACM International Conference on Information & Knowledge Management, Atlanta, GA, USA, October 17-21, 2022*. ACM, 2022, pp. 2007–2016.

[113] Jinpeng Wang, Jieming Zhu, and Xiuqiang He. “Cross-Batch Negative Sampling for Training Two-Tower Recommenders”. In: *SIGIR ’21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021*. ACM, 2021, pp. 1632–1636.

[114] Jun Wang et al. “IRGAN: A Minimax Game for Unifying Generative and Discriminative Information Retrieval Models”. In: *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Shinjuku, Tokyo, Japan, August 7-11, 2017*. ACM, 2017, pp. 515–524.

[115] Qinyong Wang et al. “Neural Memory Streaming Recommender Networks with Adversarial Training”. In: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, UK, August 19-23, 2018*. ACM, 2018, pp. 2467–2475.

[116] Xin Wang, Yudong Chen, and Wenwu Zhu. “A Survey on Curriculum Learning”. In: *IEEE Trans. Pattern Anal. Mach. Intell.* 44.9 (2022), pp. 4555–4576.

[117] Ziyang Wang et al. “Global Context Enhanced Graph Neural Networks for Session-based Recommendation”. In: *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*. 2020.

[118] Ledell Wu et al. “Scalable Zero-shot Entity Linking with Dense Entity Retrieval”. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*. Association for Computational Linguistics, 2020, pp. 6397–6407.

[119] Congying Xia et al. “Zero-shot User Intent Detection via Capsule Neural Networks”. In: *Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 2018, pp. 3090–3099.

[120] Jun Xia et al. “ProGCL: Rethinking Hard Negative Mining in Graph Contrastive Learning”. In: *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*. Vol. 162. Proceedings of Machine Learning Research. PMLR, 2022, pp. 24332–24346.

[121] Yongqin Xian et al. “Latent Embeddings for Zero-Shot Classification”. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 69–77.

[122] Teng Xiao et al. “Dynamic Bayesian Metric Learning for Personalized Product Search”. In: *Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, Beijing, China, November 3-7, 2019*. ACM, 2019, pp. 1693–1702.

[123] Junyuan Xie, Ross Girshick, and Ali Farhadi. “Unsupervised deep embedding for clustering analysis”. In: *International conference on machine learning*. PMLR. 2016, pp. 478–487.

[124] Lee Xiong et al. “Approximate Nearest Neighbor Negative Contrastive Learning for Dense Text Retrieval”. In: *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

[125] Jiaming Xu et al. “Short text clustering via convolutional neural networks”. In: *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*. 2015, pp. 62–69.

[126] Puyang Xu and Ruhi Sarikaya. “Convolutional neural network based triangular CRF for joint intent detection and slot filling”. In: *IEEE Workshop*

on Automatic Speech Recognition and Understanding (ASRU Workshop). 2013, pp. 78–83.

[127] Zhen Yang et al. “Region or global a principle for negative sampling in graph-based recommendation”. In: *IEEE Transactions on Knowledge and Data Engineering* (2022).

[128] Zhen Yang et al. “Understanding Negative Sampling in Graph Representation Learning”. In: *KDD ’20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*. ACM, 2020, pp. 1666–1676.

[129] Shaowei Yao et al. “Learning a Product Relevance Model from Click-Through Data in E-Commerce”. In: *WWW ’21: The Web Conference 2021, Virtual Event / Ljubljana, Slovenia, April 19-23, 2021*. ACM / IW3C2, 2021, pp. 2890–2899.

[130] Majid Yazdani and James Henderson. “A Model of Zero-Shot Learning of Spoken Language Understanding”. In: *Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 2015, pp. 244–249.

[131] Ka Yee Yeung and Walter L Ruzzo. “Details of the adjusted rand index and clustering algorithms, supplement to the paper an empirical study on principal component analysis for clustering gene expression data”. In: *Bioinformatics* 17.9 (2001), pp. 763–774.

[132] Wenpeng Yin et al. “Comparative study of CNN and RNN for natural language processing”. In: *arXiv preprint arXiv:1702.01923* (2017).

[133] Yiming Ying and Peng Li. "Distance Metric Learning with Eigenvalue Optimization". In: *Journal of Machine Learning Research* 13 (2012), pp. 1–26.

[134] Changlong Yu et al. "FolkScope: Intention Knowledge Graph Construction for E-commerce Commonsense Discovery". In: *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL)*. 2023.

[135] Lantao Yu et al. "SeqGAN: Sequence Generative Adversarial Nets with Policy Gradient". In: *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*. AAAI Press, 2017, pp. 2852–2858.

[136] Mo Yu et al. "Diverse Few-Shot Text Classification with Multiple Metrics". In: *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*. 2018, pp. 1206–1215.

[137] Chenwei Zhang et al. "Mining User Intentions from Medical Queries: A Neural Network Based Heterogeneous Jointly Modeling Approach". In: *International Conference on World Wide Web (WWW)*. 2016, pp. 1373–1384.

[138] Hanlei Zhang et al. "Discovering new intents with deep aligned clustering". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 35. 2021, pp. 14365–14373.

[139] Hanlei Zhang et al. "TEXTOIR: An Integrated and Visualized Platform for Text Open Intent Recognition". In: *Proceedings of the 59th Annual*

Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations. 2021, pp. 167–174.

[140] Jingqing Zhang, Piyawat Lertvittayakumjorn, and Yike Guo. "Integrating Semantic Knowledge to Tackle Zero-shot Text Classification". In: *arXiv* abs/1903.12626 (2019).

[141] Si Zhang et al. "Graph convolutional networks: a comprehensive review". In: *Computational Social Networks* 6.1 (2019), pp. 1–23.

[142] Weinan Zhang et al. "Optimizing top-n collaborative filtering via dynamic negative item sampling". In: *The 36th International ACM SIGIR conference on research and development in Information Retrieval, SIGIR '13, Dublin, Ireland - July 28 - August 01, 2013*. ACM, 2013, pp. 785–788.

[143] Weinan Zhang et al. "The First Evaluation of Chinese Human-Computer Dialogue Technology". In: *arXiv* abs/1709.10217 (2017).

[144] Yixin Zhang et al. "Enhancing Sequential Recommendation with Graph Contrastive Learning". In: *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI 2022, Vienna, Austria, 23-29 July 2022*. ijcai.org, 2022, pp. 2398–2405.

[145] Yuan Zhang, Dong Wang, and Yan Zhang. "Neural IR Meets Graph Embedding: A Ranking Model for Product Search". In: *The World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019*. ACM, 2019, pp. 2390–2400.

[146] Yuwei Zhang et al. “New Intent Discovery with Pre-training and Contrastive Learning”. In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2022, pp. 256–269.

[147] Yunhua Zhou, Guofeng Quan, and Xipeng Qiu. “A probabilistic framework for discovering new intents”. In: *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2023, pp. 3771–3784.

[148] Yanqiao Zhu et al. “Structure-enhanced heterogeneous graph contrastive learning”. In: *Proceedings of the 2022 SIAM International Conference on Data Mining (SDM)*. SIAM. 2022, pp. 82–90.