Reimagining Intent Prediction: Insights from Graph-Based Dialogue Modeling and Sentence Encoders

Daria Ledneva, Denis Kusnetsov
Neural Networks and Deep Learning Lab,
Moscow Institute of Physics and Technology, Dolgoprudny, Russia
ledneva.dr@mipt.ru, kuznetsov.den.p@phystech.edu

Abstract

This paper presents a innovative approach tailored to the specific characteristics of closed-domain dialogue systems. Leveraging scenario dialog graphs, our method effectively addresses the challenges posed by highly specialized fields, where context comprehension is of paramount importance. By modeling dialogues as sequences of transitions between intents, representing distinct goals or requests, our approach focuses on accurate intent prediction for generating contextually relevant responses. The study conducts a thorough evaluation, comparing the performance of state-of-the-art sentence encoders in conjunction with graph-based models across diverse datasets encompassing both open and closed domains. The results highlight the superiority of our methodology, offering fresh perspectives on the integration of advanced sentence encoders and graph models for precise and contextually-driven intent prediction in dialogue systems. Additionally, the use of this approach enhances the transparency of generated output, enabling a deeper understanding of the reasoning behind system responses. This study significantly advances the field of dialogue systems, providing valuable insights into the effectiveness and potential limitations of the proposed approaches.

Keywords: intent prediction, dialogue systems, graph neural networks

1. Introduction

In recent years, dialogue systems have undergone a remarkable transformation, revolutionizing the way humans communicate with computers and becoming an essential part of our daily lives (Patlan et al., 2021). These systems, computer programs capable of engaging with humans in conversational manners and emulating human-like responses (Burtsev et al., 2018), have gained widespread adoption (Chen et al., 2017). Their applications range from virtual assistants to customer service chatbots, showcasing their versatility.

One of the fundamental tasks in the field of dialogue systems is intent prediction (Lang et al., 2022), which involves the identification of the underlying intention or purpose behind a dialog participant's utterance. Precise intent prediction is crucial as it enables dialogue systems to generate contextually relevant and effective responses during the ongoing conversation (Goyal et al., 2022).

The surge in popularity of Large Language Models (LLMs) in dialog systems is noteworthy (Deng et al., 2023). However, solving the task of intent prediction using them is particularly challenging (He and Garner, 2023) due to the limitations of LLMs in grasping context, especially in highly specialized fields common to closed-domain dialog systems (Hudeček and Dušek, 2023; Finch et al., 2023). Their adaptability in such fields is restricted by this drawback in contextual understanding.

Henceforth, this paper introduces an alternative approach to constructing dialog systems, employing scenario dialog graphs to effectively address these challenges (Nagovitsin and Kuznetsov, 2022). This approach also resolves another concern related to LLMs: the transparency of their generated output (Wu et al., 2023). With scenario dialog graphs, it becomes possible to understand the reasoning behind a specific response generated by the dialog system.

By leveraging on the structured nature of closed-domain dialog systems, we represent dialogs as sequences of transitions between intents (Theodoridis, 2015), with each intent signifying a goal or request from the dialog participants. This makes accurately predicting the intent of the next statement crucial. Achieving high precision in this task empowers dialogue systems to consistently produce contextually pertinent and effective responses throughout the ongoing conversation (Goyal et al., 2022).

In light of this, our study contributes significantly to the advancement of dialog systems. We introduce an innovative methodology, harnessing sentence encoders and dialog structure to achieve precise and contextually-driven intent prediction. Our evaluation includes a comprehensive analysis of various state-of-the-art sentence encoders, assessing their performance in conjunction with graph-based models across diverse datasets encompassing both open and closed domains.

The contributions of our study is as follows: (i) the introduction of novel methodologies integrating

LREC-COLING 2024, pages 13847–13860
20-25 May, 2024.

advanced sentence encoders and graph models for accurate prediction in English-language dialog systems, (ii) an overview of various graph-based approaches that can be used to address challenges in dialogue graphs, and (iii) a meticulous analysis of the results, offering critical insights into the effectiveness and potential limitations of the proposed approaches.

All code is available here (*https://github.com/LadaNikitina/Dialog-Graph-Intent-Prediction*).

2. Related Work

2.1. Generation of Unsupervised Intents

Precise intent detection is a critical component of goal-oriented dialogue systems, significantly enhancing the accuracy of response selection models (Larson and Leach, 2022; Cai and Chen, 2020). Its primary aim is to predict the intent behind the user's next utterance based on the user's current input (Fernández-Martínez et al., 2021).

One of the main challenges in this field arises from the absence of intent annotations in many dialogue datasets. The manual markup process is not only labor-intensive but also resource-demanding. To address this, extensive research has been conducted on the formation of unsupervised clusters using clustering techniques (Du et al., 2023). These clusters represent the intents of the dialogue participants and serve as foundational elements in constructing deep learning models that predict the next intent and underlie the scenario architecture of the dialogue system. Among the various methods, the co-clustering technique (Guigourès, 2013), based on the MODL approach (Bouraoui and Lemaire, 2017), has emerged as a prominent technique for utterance clustering. It effectively utilizes a text/word adjacency matrix to define clusters. Additionally, alternative clustering approaches have been explored, including the application of K-means (Steinley, 2006) or HDBSCAN (Costa et al., 2023).

However, current clustering algorithms often struggle to capture contextual nuances, a critical aspect in understanding dialogue structures. This limitation prompted the development of a two-stage clustering algorithm (Nagovitsin and Kuznetsov, 2022), which enables the creation of clusters comprising semantically similar dialogue replicas occurring within comparable contexts.

Moreover, the selection of an appropriate sentence encoder for generating vector representations of dialogue replicas is a crucial task. It plays a pivotal role in the subsequent clustering process. The ability to predict the intent of the next utterance and form high-quality clusters is intricately linked to the semantic proximity of replicas, the nuanced capacity to encapsulate context within vector representations, and the overall quality of replica embeddings (Zhang et al., 2020). Specifically, the choice of a particular sentence encoder, among the multitude available (Muennighoff et al., 2022), exerts a substantial impact on the final intent prediction outcome (see Section 5.3 in the Experiments).

2.2. Graph-Based Intent Prediction

A distinctive characteristic of goal-oriented dialogue systems is their inherent regular structure. In essence, dialogues can be viewed as a sequence of intents expressed by participants, where each intent signifies the speaker's request or objective (Nagovitsin and Kuznetsov, 2022). Thus, the regular structure of dialogue systems enables the construction of a scenario dialogue graph based on a given set of dialogues (Bouraoui et al., 2019). Within this graph, vertices represent states within the dialogue system, while edges denote transitions between these states. Each state corresponds to specific intents of the dialogue participants. This representation allows us to frame the challenge of predicting the next utterance's intent as a link prediction problem (Wang et al., 2021; Zamini et al., 2022) within the scenario graph.

The integration of graph models signifies an emerging trend in the field of dialogue systems, enabling the potential of graph structures to enhance various aspects of dialog system functionality (He et al., 2023). Currently, several studies are dedicated to addressing the challenge of predicting the next intent in diverse domains, leveraging knowledge graphs and graph models as foundational tools (Arčan et al., 2023; Yang et al., 2020). Among these, a prevalent approach involves the use of Graph Neural Networks (GNNs) (Zhou et al., 2018), renowned for their ability to capture dependencies between vertices.

Graph methods can be categorized into homogeneous and heterogeneous approaches. Homogeneous methods, exemplified by Graph Convolutional Networks (GCNs) (Zhang et al., 2019; Zhou et al., 2023) and Graph Attention Networks (GATs) (Veličković et al., 2017), are noted for their effectiveness in modeling interdependencies among vertices. Meanwhile, handling heterogeneous graphs requires specialized techniques such as Heterogeneous Graph Attention Networks (HANs) and Graph Transformer Networks (GTNs) (Yun et al., 2019, 2022). HANs extend the GAT architecture to accommodate diverse data types, while GTNs identify useful links between vertices to generate new graph structures in an end-to-end manner.

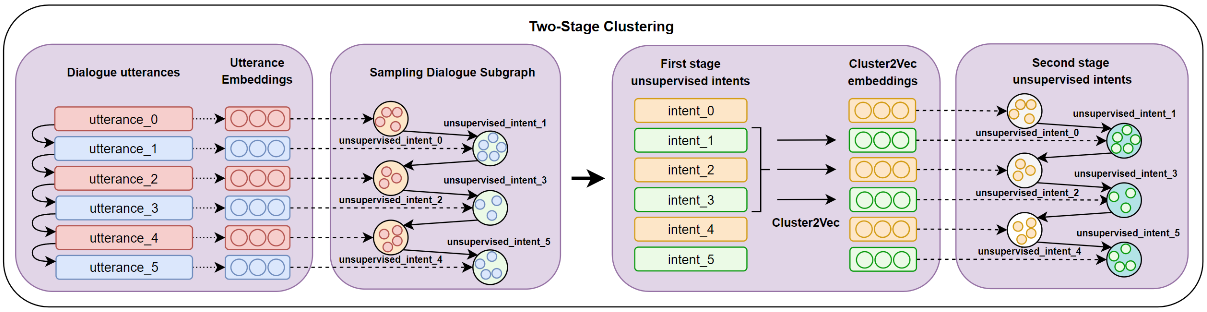

Figure 1: A two-stage algorithm for clustering dialogue utterances based on their embeddings. The first stage uses K-means clustering to group similar embeddings together. In the second stage, context embeddings are generated for each cluster using the cluster2vec method. This algorithm forms vertices in a multipartite dialogue graph.

3. Methodology

3.1. Dialogue Graph Auto Construction

As mentioned earlier, addressing the challenge of predicting intent necessitates the development of a scenario dialogue graph. This graph should display the distinct roles of the dialogue participants, along with their corresponding behaviors and interactions. For instance, closed dialogue systems typically differentiate between two roles: user and manager. Within a scenario dialogue graph, each node signifies a dialogue state or the intent of a participant at a specific moment in the conversation. It is imperative to avoid using the same vertices for intents across different participant roles, as they are driven by distinct objectives. To tackle this issue, we propose the concept of a multipartite dialogue graph, where each partite represents a specific role in the conversation. Nevertheless, open domain dialogue systems commonly involve only one role — the role of a dialogue participant. In such cases, employing a single-partite dialogue graph is considered acceptable.

To begin, the vertices of the dialogue graph should be generated based on the vector representations of the dialogue utterances within the dataset being utilized. In this study, we utilize embeddings derived from the DistilRoBERTa sentence encoder (Sanh et al., 2019). The selection of DistilRoBERTa is justified by a comparative analysis of state-of-the-art sentence encoder architectures (refer to Section 5.3 in the Experiments).

The construction of vertices for the multipartite dialog graph involves a two-stage clustering algorithm (refer to Figure 1). In the initial stage, replicas from the dialog dataset are clustered using an implementation of the K-means method from the FAISS (Johnson et al., 2019) library. This specific implementation of the K-means algorithm (Steinley, 2006) was selected for its efficiency on large dialog datasets compared to other K-means implementations. Consequently, clusters comprising dialog utterances with identical semantics and similar vector representations are established.

Moving to the second stage of clustering, context vector representations are generated for each of the clusters from the first stage, utilizing the Cluster2Vec method. The Cluster2Vec process consists of the following: every dialog is interpreted as a sequence of cluster numbers to which the respective dialog utterances belong. Subsequently, Word2Vec (Mikolov et al., 2013) training is conducted based on the obtained sequences, where the numbers representing cluster identifiers play the role of "words". This Cluster2Vec approach yields vector representations of the clusters from the initial clustering stage, encapsulating information about the context in which replicas from each cluster occur in dialogs. These context vector representations, along with an implementation of the K-means method from the FAISS library, are then employed to merge the clusters from the first stage into final clusters. These final clusters subsequently serve as the vertices of the multipartite dialog graph. In this manner, the nodes of the multipartite scenario dialog graph are ultimately established, encompassing dialog utterances with matching semantics that occur in similar contexts within dialogs.

3.2. Data preprocessing

The data preprocessing stage encompasses the preparation of a dialog dataset for training models to predict intents within dialogs. To forecast the next intent, information from the last m utterances of a dialog is utilized. To achieve this, the data is readied employing a sliding window method of length m over the entire dialog. In instances where the dialog history is shorter than m at the time of prediction, a null node is introduced. This node encompasses a single utterance with a zero vector representation, signifying the absence of an utterance in the dialog history.

Each fragment of dialog extracted from this pro-

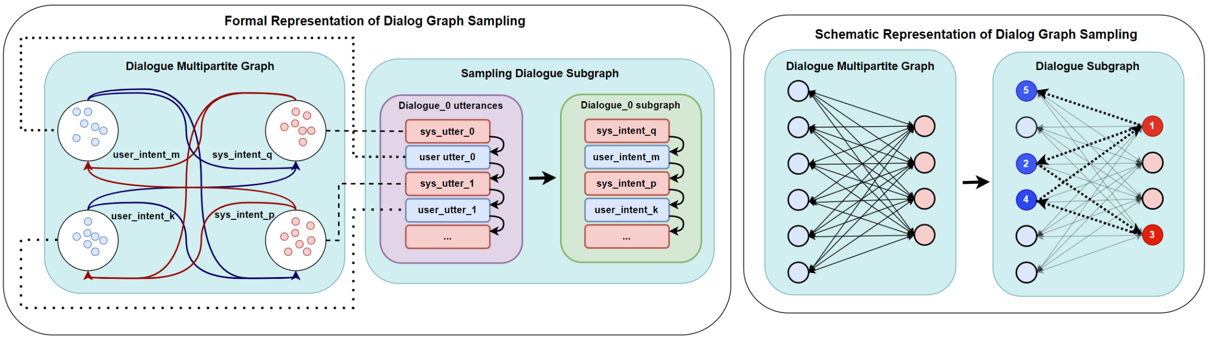

Figure 2: Representation of a dialog fragment as a subgraph of a multipartite dialog graph. The vertices in the subgraph correspond to the vertices of the multipartite graph containing statements of the dialog fragment.

cess is depicted as a directed subgraph within a multipartite dialog graph (see Figure 2). The vertices of this subgraph align with those of the multipartite dialog graph, housing the statements from the dialog fragment on which the subgraph is based. Subsequently, we generate the requisite features for both the vertices and edges of each subgraph.

4. Proposed approaches

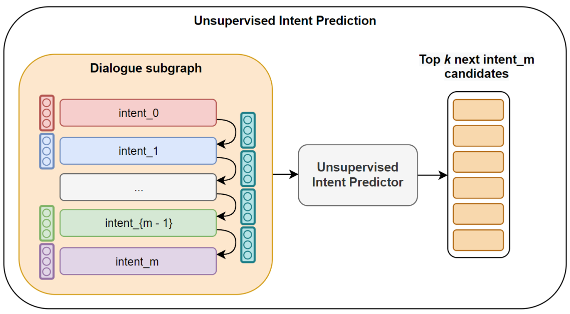

Figure 3: Prediction the intent of the next utterance in a dialogue by utilizing an intent predictor on the dialogue subgraph, along with vertex and edge features.

In this section, we provide a comprehensive overview of the approaches that have been compared within the context of addressing the next-intent prediction task. Each approach shares a common objective: predicting the intent of the next dialogue utterance based on the dialog subgraph. This is similar to predicting a vertex in a multipartite dialog graph, where each vertex represents a distinct intent (see Figure 3).

Mathematically, the problem statement can be articulated as follows: given a dialog $D = \{u_1, u_2, ..., u_t\}$, where t represents the number of utterances in the dialog and u_i denotes the i-th utterance within the dialog. For every dialog utterance u_i, the corresponding vertex v_i denoting the intent of the utterance is known. Consequently, the dialog is represented as a directed subgraph of a multipartite dialog graph $G = (V, E)$, where $V = unique(\{v_1, v_2, ..., v_t\})$ constitutes the set of vertices within the subgraph, and $E = \{(v_1, v_2), (v_2, v_3), ..., (v_{t-1}, v_t)\}$ comprises the set of edges in the subgraph. The goal of each of the proposed approaches is to predict the speaker's intent, which can be expressed mathematically as $Intent(G) = c^*$. This prediction is performed at each step in a goal-oriented dialogue system, where c^* belongs to a predefined set of N classes of intents $C = \{c_1, c_2, ..., c_N\}$ and G is a directed subgraph of a multipartite dialog graph. In scenarios where we need to predict the top-k most probable intents of the speaker, $Intent(G) = C_k^* = \{c_1^*, c_2^*, ..., c_k^*\}$, where $c_j^* \in C$. Also, $\tilde{C}$ denotes all vertices in the multipartite graph with the intents of the speaker to which the next replica in the dialog belongs. These vertices also serve as the classes used to train models for the intent classification task.

4.1. Markov Chain

As a first basic approach, we applied the Markov chain method. This method calculates transition probabilities from each vertex in a scenario dialog graph to vertices in other partitions within a multipartite dialog graph using a dialog dataset. In doing so, it identifies the most probable vertices to transition from the current vertex in the graph, considering them as potential candidates for representing the intent of the next utterance in the dialog. Formally, the approach is expressed as:

$$C_k^* = \underset{c_1, c_2, ..., c_k;\ c \in \tilde{C}}{\arg\max} P(c|v_t) \quad (1)$$

Here, $P(c|v_t)$ signifies the conditional transition probabilities from vertex v_t to vertex c in the multipartite dialog graph. These probabilities are precomputed based on dialogs from the dialog dataset.

4.2. Encoder

An alternative basic approach involves the utilization of pre-trained language models. In this study, the DistilRoBERTa encoder (Johnson et al., 2019) was selected for this purpose, as it demonstrated the most promising outcomes in generating vertices of a multipartite dialogue graph. The technique involves using a sentence encoder to obtain vector representations for prior utterances from the dialogue history and possible future utterances. These representations are then used to predict future utterances and their underlying intents.

Formally, this approach can be formulated as follows. We aim to predict k candidate vertices in a multipartite graph for the next dialogue utterance u_{t+1}. We have a set of all replicas included in the training sample U. By computing the cosine similarity $cos(h_{dr\_bert}(u_t), h_{dr\_bert}(s))$ between all utterances $s \in U$ and the previous utterance in the dialogue u_t, we arrange the utterances from U in descending order of cosine similarity. Then, for each utterance from U, the cluster number associated with the utterance's intent is determined. Candidate vertices are selected from the beginning of the sorted list of cosine similarity values until the number of unique candidate vertices reaches k.

4.3. ConveRT

As highlighted earlier, context is pivotal in dialogue systems. Consequently, we introduced a ConveRT-based approach (Henderson et al., 2020) as an alternative baseline method. This is an addition to the Encoder approach, which considers only a single utterance from the dialogue history as context. In contrast, ConveRT is a dual-encoder model crafted to accommodate multiple utterances from a dialogue history. By incorporating ConveRT alongside the Encoder approach, we aim to achieve a more profound comprehension of the impact of utterance context in our study.

This approach is identical to the Encoder approach, but it calculates the cosine similarity as $cos(h_{mc}(\{u_1, ..., u_t\}), h_r(s))$, where h_{mc} retrieves the full dialog history representation and h_r obtains the response representation.

4.4. ConveRT-MAP

Without the approach based on fine-tuning the language model for our task, our comparison of intent prediction methods would be incomplete. Hence, we introduced the ConveRT-MAP approach, in which we fine-tuned the ConveRT model. Fine-tuning the model is crucial in creating more relevant vector representations of utterances, thus enhancing the accuracy of intent prediction (see Section 5.3 in the Experiments).

This method involves using ConveRT as a base model and extending it with three fully connected non-linear feed-forward layers, followed by a linear layer. We trained the resulting model using contrastive loss. In this case, consecutive dialogue replicas from the dialog dataset served as positive pairs, while negative pairs consisted of replicas randomly selected from other positive pairs within the batches.

This approach is otherwise identical to the previous two approaches but it calculates the cosine similarity as $cos(h_{mc\_map}(\{u_1, ..., u_t\}), h_{r\_map}(s))$, where h_{mc_map} retrieves the full dialog history representation and h_{r_map} obtains the response representation.

4.5. CatBoost

Experimental results show that among gradient boosting libraries, CatBoost (Prokhorenkova et al., 2017; Dorogush et al., 2018) exhibits the best performance for the intent prediction task. In the implementation of this approach, a vector is generated for each subgraph, which is a concatenation of the features of all the vertices included in that subgraph. The resulting vector is then used as input embedding for the CatBoost algorithm. Formally, the approach can be represented as follows:

$$C_k^* = \underset{c_1, c_2, ..., c_k;\ c \in \tilde{C}}{\arg\max} \ CatBoost(||_{v_i \in V}\ h_f(v_i)), \quad (2)$$

Here, $h_f(v_i)$ is a function that generates a vector representation of the vertex's features from the dialog subgraph.

4.6. Message Passing

Graph Neural Networks (GNNs) are a class of neural networks designed to operate on graph-structured data. They enable the integration of information from a node's neighbors, allowing for the modeling of complex relationships and dependencies within the graph. Of the various GNN models, Graph Attention Networks (GATs) stand out for their ability to assign different importance values to messages from neighboring vertices during the aggregation process using an attention mechanism, making them the most effective.

Formally, the approach can be represented as follows:

$$h_v^l = \Big\Vert_{k=1}^{K} \sigma\left(\sum_{\tilde{v} \in N_v} \alpha_{v\tilde{v}} W^k h_v^{l-1}\right) \quad (3)$$

$$P(c \mid G) = softmax(W(\underset{v_i \in V}{||}\ h_v^L) + b) \quad (4)$$

$$C_k^* = \underset{c_1, c_2, \ldots, c_k;\ c \in \tilde{C}}{\arg\max} P(c \mid G) \quad (5)$$

Here, K is the number of heads in the GAT and σ is an activation function.

4.7. FastGTN

Dialog graphs contain vertices of different types depending on their correspondence to the vertices in a multipartite dialog graph. Graph Transformation Networks (GTNs) are employed to address problems related to such graphs. This paper introduces FastGTN, an enhanced implementation of GTN that requires significantly fewer resources and less training time. Formally, the approach can be represented as follows:

$$P(c \mid G) = softmax(W(\underset{v_i \in V}{\|} h_{gtn}(v)) + b) \quad (6)$$

$$C_k^* = \underset{c_1, c_2, \ldots, c_k;\ c \in \tilde{C}}{\arg\max} P(c \mid G) \quad (7)$$

Here, h_{gtn} represents the output vectors of vertices obtained from GTN.

5. Experiments

For more explanations of the implementation details of approaches, readers are encouraged to refer to Section B in the Appendix.

5.1. Utilized Datasets

We evaluated our intent prediction models using a diverse set of datasets from both open and closed domains.

5.1.1. Open Domain Datasets

PersonaChat Zhang et al. (2018): Designed for chitchat-oriented dialogue systems, this dataset comprises over $160,000$ conversational exchanges covering a wide range of topics.

DailyDialog Li et al. (2017): With $13,118$ dialogues, this dataset encompasses discussions on various topics like life events and personal interests.

5.1.2. Closed Domain Datasets

MultiWOZ 2.2 Zang et al. (2020): This dataset includes over $10,000$ dialogues across seven domains, such as hotels, restaurants, hospitals, and transportation.

FoCus Jang et al. (2022): Encompassing $14,452$ dialogues, this dataset focuses on discussions about geographical landmarks, leveraging Wikipedia knowledge.

Taskmaster Byrne et al. (2019): This dataset features $13,215$ dialogues in six domains, including $7,708$ written and $5,507$ spoken dialogues.

5.2. Metric

The model's performance was evaluated using the MAR ($Mean\ Average\ Recall$) and $Recall@k$ metrics, quantifying the accuracy of predicting the intent of the next utterance. For each subgraph within the test sample, this metric assigns a score of 1 if the vertex corresponding to the intent of the next utterance is among the top-k predicted vertices based on the transition probabilities. Otherwise, a score of 0 is assigned. These scores are then averaged across all utterances and dialogues.

Acknowledging the non-obvious choice of the Recall metric, it is imperative to explain our rationale for choosing Recall over Accuracy. In future experimental design involving dialogue statements with multiple different intents, it becomes imperative not only to identify the correct candidate but also to assess how many candidates the model has selected from the correct ones. This consideration led us to prefer Recall, and in particular the Recall@k metric, as it fits seamlessly with our experimental goals. In addition, we would like to emphasise that our Recall@k metric is conceptually aligned with the Accuracy@k metric within the parameters of our ongoing research.

By employing various values of k in the set $\{1, 3, 5, 10\}$, the $Recall@k$ metric provides an estimation of the $Recall$ metrics distribution and offers insights into the effectiveness of predicting candidate vertices. For enhanced comparability between the approaches, we utilized the MAR metric, calculated as the arithmetic mean of $Recall@k$ values where k is drawn from the set $\{1, 3, 5, 10\}$. This approach strikes a balance between computational feasibility and providing a meaningful approximation of MAR across the entire spectrum of k values ranging from 1 to 10.

In order to underscore the distinctions in cluster formation for each dialogue participant on closed-domain datasets, we present separate metrics for predicting user and dialogue system intents.

5.3. Sentence Encoder Selection

The selection of an appropriate sentence encoder plays a pivotal role in our research, as it directly impacts the generation of vector representations for dialogue responses. This choice is critical in shaping the dialogue graph nodes, subsequently influencing the model's capability to predict the intention behind the next utterance. To tackle this challenge, we conducted a thorough comparative analysis of various sentence encoder architectures, meticulously assessing their performance on dialogue data.

Our evaluation metrics, outlined in Table 1, offer a comprehensive examination of how the selection of a text encoder influences the formation of

Models	MPNet	MPNet-one-stage	DistilRoBERTa	S-BERT	MiniLM	GloVe	GPT	T5
# of Parameters	109M	109M	82M	22M	33M	120M	125M	335M
				Encoder				
Recall@1	**23.63 ± 0.531**	19.18 ± 0.421	**23.92 ± 0.806**	21.22 ± 1.417	**23.15 ± 1.489**	13.35 ± 0.341	21.01 ± 1.233	**23.08 ± 0.884**
Recall@3	**47.87 ± 0.469**	41.31 ± 0.435	**47.57 ± 0.219**	43.55 ± 1.086	**47.13 ± 1.508**	32.51 ± 0.890	44.36 ± 1.241	**48.95 ± 0.719**
Recall@5	**58.92 ± 0.738**	53.99 ± 0.157	**58.81 ± 0.405**	53.67 ± 1.012	**59.50 ± 0.419**	44.07 ± 0.840	54.90 ± 1.223	**60.01 ± 0.343**
Recall@10	**74.19 ± 1.109**	72.21 ± 0.023	**73.75 ± 1.164**	68.28 ± 0.914	**74.35 ± 0.372**	61.97 ± 1.046	71.72 ± 1.541	**73.70 ± 0.271**
				Message Passing				
Recall@1	**46.94 ± 1.135**	37.79 ± 0.818	**46.55 ± 1.288**	45.82 ± 1.263	**46.33 ± 0.766**	38.77 ± 1.726	44.78 ± 0.633	**48.23 ± 0.614**
Recall@3	**74.40 ± 0.277**	67.12 ± 0.386	**74.36 ± 0.533**	71.80 ± 0.804	**72.82 ± 1.033**	64.07 ± 0.797	71.07 ± 0.212	**74.29 ± 0.687**
Recall@5	**83.45 ± 0.136**	80.46 ± 0.470	**83.63 ± 0.558**	81.62 ± 0.756	82.15 ± 0.670	76.47 ± 0.336	81.50 ± 0.211	**83.90 ± 0.532**
Recall@10	**92.74 ± 0.352**	92.61 ± 0.703	**93.17 ± 0.758**	92.27 ± 0.541	**92.35 ± 0.486**	89.99 ± 0.534	92.37 ± 0.345	**93.31 ± 0.752**
				Markov Chain				
Recall@1	**37.62 ± 0.503**	27.56 ± 1.007	**37.99 ± 0.599**	36.66 ± 1.207	**37.47 ± 0.648**	28.66 ± 1.735	36.98 ± 1.105	**36.81 ± 0.735**
Recall@3	63.86 ± 0.282	55.20 ± 0.993	**65.52 ± 0.469**	63.43 ± 0.965	**64.65 ± 0.513**	52.76 ± 1.503	61.29 ± 0.940	**65.28 ± 0.588**
Recall@5	75.19 ± 0.474	70.81 ± 1.164	**76.96 ± 0.269**	74.45 ± 0.977	**76.20 ± 0.322**	64.97 ± 1.106	72.83 ± 0.452	**76.38 ± 0.638**
Recall@10	**88.56 ± 0.728**	88.23 ± 0.483	**89.62 ± 0.564**	87.78 ± 0.730	**88.48 ± 0.223**	82.92 ± 0.151	86.71 ± 0.294	**89.37 ± 0.727**

Table 1: Evaluation of text encoders in generating vector representations for dialogue utterances in the MultiWOZ dataset and their impact on the three primary approaches: Message Passing, Encoder, and Markov Chain.

dialogue graph nodes and the accuracy of predicting the next intention across different approaches. These metrics provide valuable insights into the text encoders' performance in generating vector representations for dialogue utterances within the MultiWOZ dataset. Furthermore, they also highlight on the performance implications for three primary approaches: Message Passing, Encoder and Markov Chain, representing significant methods in intention prediction, encompassing probabilistic, encoder-based, and graph-based methods.

Upon analyzing the experimental results, it becomes evident that both DistilRoBERTa and T5 exhibit exceptional performance. However, considering the significantly lower computational requirements of DistilRoBERTa – four times less than T5 – we opted for its utilization in our research. This decision not only aligns with our research goals but also reflects a balance between performance and computational efficiency.

5.4. Cluster Number

In our study, we utilized different quantities of clusters in the first and second stages of clustering. Specifically, we utilized 200, 400, and 800 clusters in the first stage, and 30, 60, and 120 clusters in the second stage.

It's important to recognize that each dataset carries its own unique characteristics. The choice of the exact number of clusters, in both the initial and subsequent stages of clustering, is fundamentally dependent on the specific task at hand. Hence, we opted for the minimum, average, and maximum quantities of clusters, taking into consideration the specific attributes of the datasets used for approach comparison, such as the number of supervised clusters in the MultiWOZ dataset.

By carefully examining the metrics obtained from different cluster configurations, valuable insights can be gained about the relationship between the number of clusters and the accuracy of predicting the next intent. This, in turn, allows choosing the most appropriate number of clusters for the first and second stages of clustering when replicating the proposed techniques on other datasets with identical intent distribution.

Approach	# Parameters	Relative Training Time	# Clusters First Stage	# Clusters Second Stage	PersonaChat	DailyDialog
Markov Chain	10K	0.13	200	30	52.50 ± 2.27	49.91 ± 0.85
			400	60	41.67 ± 2.28	40.53 ± 2.66
			800	120	32.72 ± 1.03	31.48 ± 0.91
Message Passing	82M + 3.7M	0.47	200	30	**58.86 ± 1.06**	**57.13 ± 2.28**
			400	60	**48.79 ± 0.68**	**47.15 ± 0.71**
			800	120	**42.96 ± 0.68**	**38.52 ± 0.42**
CatBoost	82M + 2.2M	1.00	200	30	**59.31 ± 1.24**	**58.67 ± 0.90**
			400	60	**50.12 ± 0.78**	**47.55 ± 1.20**
			800	120	**42.56 ± 0.63**	**39.50 ± 0.60**
FastGTN	82M + 1.9M	0.49	200	30	**60.21 ± 2.29**	55.88 ± 0.54
			400	60	**49.11 ± 0.45**	**46.35 ± 0.71**
			800	120	**41.68 ± 1.35**	**38.92 ± 0.96**
Encoder	82M	0.50	200	30	43.45 ± 2.20	48.92 ± 0.58
			400	60	30.95 ± 2.02	39.95 ± 1.61
			800	120	24.10 ± 4.06	31.16 ± 0.66
ConveRT	46M	0.36	200	30	45.39 ± 1.46	50.24 ± 2.35
			400	60	35.01 ± 2.96	40.65 ± 0.92
			800	120	27.32 ± 2.33	32.27 ± 0.57
ConveRT MAP	46M + 2M	0.78	200	30	47.08 ± 2.01	50.51 ± 2.03
			400	60	39.97 ± 1.69	38.41 ± 2.15
			800	120	20.78 ± 2.01	29.66 ± 1.82

Table 2: Experimental results for Mean Average Recall metric: various intent prediction approaches on the open domain datasets. The training time of the models was counted from the start of training until the Early Stopping. To ensure stability of results, all approaches were trained on 3 different sets of clusters and the resulting metrics were averaged.

6. Results and Discussion

This section provides an overview of the outcomes obtained through various approaches applied to both closed-domain and open-domain dia-

Approach	# Parameters	Relative Training Time	Dataset # Clusters First Stage	Dataset # Clusters Second Stage	MultiWOZ User	MultiWOZ Dialog System	MultiWOZ All	FoCus User	FoCus Dialog System	FoCus All	Taskmaster User	Taskmaster Dialog System	Taskmaster All
Markov Chain	10K	0.13	200	30	59.47 ± 0.77	75.57 ± 0.59	67.52 ± 0.48	52.55 ± 1.30	52.15 ± 2.06	52.35 ± 0.98	57.79 ± 0.45	59.63 ± 0.67	58.77 ± 0.51
			400	60	47.05 ± 1.88	66.19 ± 1.50	56.61 ± 1.60	46.67 ± 0.70	44.46 ± 0.71	45.57 ± 0.56	49.84 ± 0.86	49.06 ± 0.29	49.52 ± 0.52
			800	120	30.90 ± 1.26	48.33 ± 1.47	39.62 ± 0.43	39.67 ± 1.91	39.86 ± 0.76	39.77 ± 0.81	42.60 ± 0.44	43.57 ± 0.24	43.14 ± 0.18
Message Passing	82M + 3.7M	0.47	200	30	**65.24 ± 1.09**	**83.62 ± 0.64**	**74.43 ± 0.78**	**66.34 ± 2.31**	**68.80 ± 0.70**	**67.57 ± 1.46**	**72.04 ± 0.70**	**78.69 ± 0.60**	**75.41 ± 0.45**
			400	60	**52.66 ± 0.44**	**75.88 ± 0.78**	**64.27 ± 0.33**	**59.56 ± 1.67**	**63.36 ± 0.72**	**61.46 ± 0.71**	64.73 ± 0.53	**69.98 ± 0.47**	**67.40 ± 0.33**
			800	120	**35.93 ± 0.72**	**58.35 ± 0.92**	**47.14 ± 0.67**	**54.64 ± 1.05**	**56.07 ± 0.90**	**55.35 ± 0.61**	**57.56 ± 0.41**	**64.00 ± 0.37**	**60.83 ± 0.32**
CatBoost	82M + 2.2M	1.00	200	30	**65.88 ± 0.54**	**83.09 ± 0.56**	**74.48 ± 0.45**	**65.71 ± 0.37**	**69.09 ± 0.31**	**67.41 ± 0.20**	71.57 ± 0.30	**78.23 ± 0.52**	**74.94 ± 0.24**
			400	60	51.07 ± 1.07	73.09 ± 0.81	62.08 ± 0.83	**59.61 ± 1.47**	60.91 ± 0.46	**60.26 ± 0.77**	**65.03 ± 0.34**	68.93 ± 0.33	67.01 ± 0.24
			800	120	**37.16 ± 0.58**	55.45 ± 0.74	46.30 ± 0.59	**54.55 ± 0.35**	53.94 ± 0.74	54.25 ± 0.49	56.53 ± 0.35	62.60 ± 0.29	59.61 ± 0.30
FastGTN	82M + 1.9M	0.49	200	30	**65.55 ± 0.64**	**83.04 ± 0.48**	**74.30 ± 0.26**	**65.12 ± 2.73**	**68.98 ± 1.16**	**67.05 ± 1.38**	**72.53 ± 0.41**	**78.30 ± 0.51**	**75.46 ± 0.36**
			400	60	**51.84 ± 0.66**	**75.94 ± 0.95**	**63.89 ± 0.55**	55.89 ± 1.93	61.76 ± 0.58	58.82 ± 1.04	**65.84 ± 0.50**	**70.11 ± 0.36**	**68.01 ± 0.29**
			800	120	**36.40 ± 0.90**	**58.38 ± 1.29**	**47.39 ± 0.41**	**54.19 ± 1.50**	**55.91 ± 0.28**	**55.05 ± 0.77**	**57.52 ± 0.51**	**64.27 ± 0.47**	**60.93 ± 0.43**
Encoder	82M	0.50	200	30	34.69 ± 1.20	67.33 ± 0.90	51.01 ± 0.65	39.01 ± 1.63	59.11 ± 0.80	49.06 ± 0.77	46.08 ± 0.72	49.05 ± 0.42	47.56 ± 0.19
			400	60	24.67 ± 0.44	53.40 ± 2.03	39.04 ± 0.90	32.50 ± 0.87	50.39 ± 0.73	41.45 ± 0.56	36.35 ± 0.24	40.88 ± 0.20	38.61 ± 0.19
			800	120	15.31 ± 0.33	36.35 ± 0.74	25.83 ± 0.41	28.55 ± 0.41	43.16 ± 0.43	35.86 ± 0.26	27.82 ± 0.14	31.21 ± 0.14	29.52 ± 0.11
ConveRT	46M	0.36	200	30	32.81 ± 0.78	57.94 ± 0.94	45.38 ± 0.81	38.13 ± 0.85	60.62 ± 0.32	49.38 ± 0.50	47.52 ± 0.36	59.80 ± 0.78	53.66 ± 0.34
			400	60	21.10 ± 0.23	46.25 ± 1.00	33.67 ± 0.53	33.19 ± 0.63	52.53 ± 0.87	42.86 ± 0.45	37.87 ± 0.57	45.92 ± 0.64	41.90 ± 0.44
			800	120	12.71 ± 0.56	29.38 ± 0.69	21.04 ± 0.27	28.59 ± 0.23	45.80 ± 0.85	37.20 ± 0.47	29.54 ± 0.31	38.52 ± 0.18	34.03 ± 0.23
ConveRT MAP	46M + 2M	0.78	200	30	51.75 ± 1.87	75.97 ± 1.08	63.86 ± 1.38	55.74 ± 1.33	60.11 ± 1.49	57.92 ± 0.86	63.18 ± 0.68	70.82 ± 0.90	67.00 ± 0.68
			400	60	39.39 ± 1.33	61.44 ± 1.31	50.41 ± 1.32	44.31 ± 1.38	47.52 ± 1.40	45.92 ± 1.25	54.54 ± 0.61	58.59 ± 0.88	56.56 ± 0.53
			800	120	22.20 ± 1.21	39.75 ± 0.36	31.35 ± 0.58	37.62 ± 0.42	36.99 ± 1.43	37.29 ± 0.61	43.61 ± 1.09	49.61 ± 0.90	46.61 ± 0.99

Table 3: Experimental results for Mean Average Recall metric: various intent prediction approaches on the closed domain datasets. The training time of the models was counted from the start of training until the Early Stopping. The all metric is the average of the user metric and the dialogue system metric. To ensure stability of results, all approaches were trained on 3 different sets of clusters and the resulting metrics were averaged.

log datasets (see Table 2 and Table 3), evaluated using the MAR ($Mean\ Average\ Recall$) metric.

To visually highlight the distinctions between the approaches, we provide a comparative results table. This table offers a comparison (refer to Table 4) of the different methods based on the evaluation results using the $Mean\ Average\ Recall$ metric. Each method is assigned a score of 1 if it outperforms the others on a particular metric; otherwise, it receives a score of 0. Then, all the obtained scores for each method and dataset are summarized.

Closed Domain Datasets Results. The comparative table highlights that, in closed-domain datasets, the Message Passing (MP) approach demonstrated superior performance. Additionally, the Graph Transformer Network (GTN) exhibited commendable results, surpassing both gradient-based boosting and encoder-based techniques. It's noteworthy that both MP and GTN approaches excelled in terms of execution speed and demanded fewer computational resources compared to alternative methods. This underscores their effectiveness and practicality in utilizing dialog graphs for intent prediction.

Open Domain Datasets Results. The comparative table indicates that the approach employing gradient boosting demonstrated the most promising performance. This suggests that open-domain dialog systems comprise a much larger number of states in dialogues and lack a distinct regular structure, making it challenging to obtain a high-quality graph representation of such dialog systems.

Graph Models' Superiority over Text-Based Approaches. The study confirms the superiority of graph models over text-based architectures in addressing the challenge of intent prediction in dialog systems. Specifically, graph models outperformed both a simple text-based encoder and an additionally trained ConveRT-MAP text-based encoder. This underscores the criticality of accounting for structural relationships among dialog elements.

Asymmetry in Dialogue Roles. When analyzing the metrics on closed-domain datasets, a significant distinction became apparent between user metrics and dialog system metrics. This disparity occurs from the asymmetric roles that participants play in a dialog, emphasizing the importance of considering role asymmetry in the future research.

7. Conclusion

In conclusion, our research sheds light on the efficacy of graph-based models in intent prediction for dialog systems. In closed-domain datasets, both MP and GTN approaches proved to be robust performers, excelling not only in accuracy but also in computational efficiency. On the other hand, open-domain datasets present a unique challenge due to their inherent complexity and lack of regular structure, which makes them less amenable to graph-based representation.

Furthermore, our findings emphasize the superiority of graph models over text-based approaches, underscoring the significance of capturing structural relationships among dialog elements. It's worth noting that the choice of sentence encoder significantly impacts the accuracy of the approaches.

Dataset	Markov Chain	Message Passing	CatBoost	FastGTN	Encoder	ConveRT	ConveRT-MAP	Max Score
MultiWOZ	0	**9**	4	**9**	0	0	0	**9**
FoCus	0	**9**	6	6	0	0	0	**9**
Taskmaster	0	8	3	**9**	0	0	0	**9**
DailyDialog	0	**3**	**3**	2	0	0	0	**3**
PersonaChat	0	**3**	**3**	**3**	0	0	0	**3**
Closed Domain Summary	0	**26**	13	24	0	0	0	**27**
Open Domain Summary	0	**6**	**6**	5	0	0	0	**6**

Table 4: The table shows how different intent prediction methods performed in research. Each method gets a score of 1 if it does better than others on a specific metric; otherwise, it gets a score of 0. The table summarizes all the scores for each method and dataset.

Overall, this study provides valuable insights into the application of graph-based models in enhancing the accuracy and efficiency of intent prediction in dialog systems across various domains.

8. Limitations

While our study provides valuable insights, there are several considerations:

Language Focus. Our experiments primarily centered on English dialog datasets. Generalizing our findings to multilingual settings may require further exploration.

Participant Pool Size. The datasets involved a relatively small number of participants, potentially limiting representation of real-world dialog dynamics. Larger, more diverse datasets would enhance model evaluation.

Traditional Dialogue Emphasis. We focused on conventional dialogues, excluding non-standard formats like social media conversations. Adapting models for these unique patterns warrants further investigation.

Clustering Impact. The quality of clustering affects our graph-based approaches. Future work should refine clustering techniques for more reliable results. It is very essential to improve clustering methods, especially when dealing with large datasets with multiple topics. Future research should focus on optimising clustering methods to provide robust and scalable results, and on conducting experiments with large number of cluster on large datasets.

Encoder Selection Sensitivity. Our experiments highlighted the critical role of sentence encoders. Further research should explore domain-specific encoder adaptation for optimal performance.

In conclusion, while our study offers valuable insights into graph-based dialog modeling, it's important to acknowledge these limitations. Addressing them in future research will broaden the applicability and effectiveness of our models across diverse settings.

9. Funding

This work was supported by a grant for research centers in the field of artificial intelligence, provided by the Analytical Center for the Government of the Russian Federation in accordance with the subsidy agreement (agreement identifier 000000D730321P5Q0002) and the agreement with the Moscow Institute of Physics and Technology dated November 1, 2021 No. 70-2021-00138.

10. Ethics Statement

Our work on Context-Aware Unsupervised Intent Prediction has ethical considerations that we would like to address.

Firstly, in our research, we have strictly adhered to ethical guidelines regarding data collection and usage. We have used publicly available datasets and have ensured the privacy and anonymity of the participants involved in the dialogues. Any personally identifiable information has been carefully removed or anonymized to protect the privacy of individuals.

Secondly, we acknowledge the importance of maintaining fairness and avoiding bias in dialogue systems. Our models have been trained on diverse datasets to ensure inclusivity and mitigate biases that may arise from imbalanced data. We have made efforts to minimize any potential bias in our models and aim for fair representation of all individuals and groups in dialogue interactions.

While we have taken these ethical considerations into account, we also acknowledge that the field of AI and dialogue systems is continually evolving, and new ethical challenges may arise. We remain committed to upholding ethical standards, staying informed about emerging ethical guidelines, and addressing any ethical concerns that may arise as our work progresses.

It is our belief that by considering and addressing ethical considerations, we can contribute to the development of AI systems that have a positive impact on society and promote responsible and ethical dialogue interactions.

11. Bibliographical References

Mihael Arčan, Sampritha Manjunath, Cécile Robin, Ghanshyam Verma, Devishree Pillai, Simon Sarkar, Sourav Dutta, Haytham Assem, John McCrae, and Paul Buitelaar. 2023. Intent classification by the use of automatically generated knowledge graphs. *Information*, 14:288.

Jean-Leon Bouraoui, Sonia Le Meitour, Romain Carbou, Lina M. Rojas Barahona, and Vincent Lemaire. 2019. Graph2Bots, unsupervised assistance for designing chatbots. In *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue*, pages 114–117, Stockholm, Sweden. Association for Computational Linguistics.

Jean-Léon Bouraoui and Vincent Lemaire. 2017. Cluster-based graphs for conceiving dialog systems. In *Workshop DMNLP at European Conference on Machine Learning (ECML)*.

Mikhail S Burtsev, Alexander V Seliverstov, Rafael Airapetyan, Mikhail Arkhipov, Dilyara Baymurzina, Nickolay Bushkov, Olga Gureenkova, Taras Khakhulin, Yuri Kuratov, Denis Kuznetsov, et al. 2018. Deeppavlov: Open-source library for dialogue systems. In *ACL (4)*, pages 122–127.

Wanling Cai and Li Chen. 2020. Predicting user intents and satisfaction with dialogue-based conversational recommendations. In *Proceedings of the 28th ACM Conference on User Modeling, Adaptation and Personalization*, UMAP '20, page 33–42, New York, NY, USA. Association for Computing Machinery.

Hongshen Chen, Xiaorui Liu, Dawei Yin, and Jiliang Tang. 2017. A survey on dialogue systems: Recent advances and new frontiers. *ACM SIGKDD Explorations Newsletter*, 19.

Rita Costa, Bruno Martins, Sérgio Viana, and Luisa Coheur. 2023. Towards a fully unsupervised framework for intent induction in customer support dialogues.

Yang Deng, Lizi Liao, Liang Chen, Hongru Wang, Wenqiang Lei, and Tat-Seng Chua. 2023. Prompting and evaluating large language models for proactive dialogues: Clarification, target-guided, and non-collaboration.

Anna Veronika Dorogush, Vasily Ershov, and Andrey Gulin. 2018. Catboost: gradient boosting with categorical features support.

Bingzhu Du, Nan Su, Yuchi Zhang, and Yongliang Wang. 2023. A two-stage progressive intent clustering for task-oriented dialogue. In *Proceedings of The Eleventh Dialog System Technology Challenge*, pages 48–56, Prague, Czech Republic. Association for Computational Linguistics.

Fernando Fernández-Martínez, David Griol, Zoraida Callejas, and Cristina Luna Jiménez. 2021. An approach to intent detection and classification based on attentive recurrent neural networks.

Sarah E. Finch, Ellie S. Paek, and Jinho D. Choi. 2023. Leveraging large language models for automated dialogue analysis.

Abhinav Goyal, Anupam Singh, and Nikesh Garera. 2022. End-to-end speech to intent prediction to improve E-commerce customer support voicebot in Hindi and English. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 579–586, Abu Dhabi, UAE. Association for Computational Linguistics.

Romain Guigourès. 2013. The application of coclustering in exploratory data analysis.

Mutian He and Philip N. Garner. 2023. Can chatgpt detect intent? evaluating large language models for spoken language understanding.

Yacheng He, Qianghuai Jia, Lin Yuan, Ruopeng Li, Yixin Ou, and Ningyu Zhang. 2023. A concept knowledge graph for user next intent prediction at alipay. In *Companion Proceedings of the ACM Web Conference 2023*. ACM.

Matthew Henderson, Iñigo Casanueva, Nikola Mrkšić, Pei-Hao Su, Tsung Hsien Wen, and Ivan Vulić. 2020. Convert: Efficient and accurate conversational representations from transformers. pages 2161–2174.

Vojtěch Hudeček and Ondřej Dušek. 2023. Are llms all you need for task-oriented dialogue?

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547.

Hao Lang, Yinhe Zheng, Jian Sun, Fei Huang, Luo Si, and Yongbin Li. 2022. Estimating soft labels for out-of-domain intent detection. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 261–276, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Stefan Larson and Kevin Leach. 2022. A survey of intent classification and slot-filling datasets for task-oriented dialog.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space.

Niklas Muennighoff, Nouamane Tazi, Loïc Magne, and Nils Reimers. 2022. Mteb: Massive text embedding benchmark. *arXiv preprint arXiv:2210.07316*.

Mark Nagovitsin and Denis Kuznetsov. 2022. *DGAC: Dialogue Graph Auto Construction Based on Data with a Regular Structure*, pages 508–529.

Atharv Patlan, Shiven Tripathi, and Shubham Korde. 2021. A review of dialogue systems: From trained monkeys to stochastic parrots.

Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. 2017. Catboost: unbiased boosting with categorical features.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *ArXiv*, abs/1910.01108.

Douglas Steinley. 2006. K-means clustering: A half-century synthesis. *The British journal of mathematical and statistical psychology*, 59:1–34.

S. Theodoridis. 2015. *Probabilistic Graphical Models*, pages 795–843.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Y. Bengio. 2017. Graph attention networks.

Meihong Wang, Linling Qiu, and Xiaoli Wang. 2021. A survey on knowledge graph embeddings for link prediction. *Symmetry*, 13:485.

Zhaofeng Wu, William Merrill, Hao Peng, Iz Beltagy, and Noah A. Smith. 2023. Transparency helps reveal when language models learn meaning.

Kai Yang, Xinyu Kong, Jie Zhang, and Gerard de Melo. 2020. Reinforcement learning over knowledge graphs for explainable dialogue intent mining. *IEEE Access*, PP:1–1.

Seongjun Yun, Minbyul Jeong, Raehyun Kim, Jaewoo Kang, and Hyunwoo Kim. 2019. Graph transformer networks.

Seongjun Yun, Minbyul Jeong, Sungdong Yoo, Seunghun Lee, Sean S. Yi, Raehyun Kim, Jaewoo Kang, and Hyunwoo J. Kim. 2022. Graph transformer networks: Learning meta-path graphs to improve gnns. *Neural Networks*, 153:104–119.

Mohamad Zamini, Hassan Reza, and Minou Rabiei. 2022. A review of knowledge graph link prediction using graph neural networks.

Jian-Guo Zhang, Kazuma Hashimoto, Wenhao Liu, Chien-Sheng Wu, Yao Wan, Philip S. Yu, Richard Socher, and Caiming Xiong. 2020. Discriminative nearest neighbor few-shot intent detection by transferring natural language inference.

Si Zhang, Hanghang Tong, Jiejun Xu, and Ross Maciejewski. 2019. Graph convolutional networks: a comprehensive review. *Computational Social Networks*, 6.

Jie Zhou, Ganqu Cui, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, and Maosong Sun. 2018. Graph neural networks: A review of methods and applications.

Yuchen Zhou, Hongtao Huo, Zhiwen Hou, and Fanliang Bu. 2023. A deep graph convolutional neural network architecture for graph classification. *PloS one*, 18:e0279604.

12. Language Resource References

Bill Byrne, Karthik Krishnamoorthi, Chinnadhurai Sankar, Arvind Neelakantan, Daniel Duckworth, Semih Yavuz, Ben Goodrich, Amit Dubey, Kyu-Young Kim, and Andy Cedilnik. 2019. Taskmaster-1: Toward a realistic and diverse dialog dataset.

Yoonna Jang, Jungwoo Lim, Yuna Hur, Dongsuk Oh, Suhyune Son, Yeonsoo Lee, Donghoon Shin, Seungryong Kim, and Heuiseok Lim. 2022. Call for customized conversation: Customized conversation grounding persona and knowledge. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 10803–10812.

Yanran Li, Hui Su, Xiaoyu Shen, Wenjie Li, Ziqiang Cao, and Shuzi Niu. 2017. Dailydialog: A manually labelled multi-turn dialogue dataset. In *Proceedings of The 8th International Joint Conference on Natural Language Processing (IJCNLP 2017)*.

Zang, Xiaoxue and Rastogi, Abhinav and Sunkara, Srinivas and Gupta, Raghav and Zhang, Jianguo and Chen, Jindong. 2020. *MultiWOZ 2.2: A Dialogue Dataset with Additional Annotation Corrections and State Tracking Baselines*.

Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. 2018. Personalizing dialogue agents: I have a dog, do you have pets too? *arXiv preprint arXiv:1801.07243*.

A. Examples Of Graph Nodes

In this section, we present Table 5 with samples from the nodes of the graph constructed using the MultiWOZ 2.2 dataset.

B. Implementation Details

We employed various techniques and strategies to optimize our graph-based approaches. To ensure efficient training, we utilized the Adam optimizer for each approach. The Adam optimizer accurately updates the model parameters during training, facilitating faster convergence and enhancing overall performance.

To capture the information from vertex representations and create a complete graph representation, we incorporated a pooling module into all graph-based approaches. This pooling module aggregated the features of vertices, providing an embedding of the overall graph structure. Additionally, a linear layer was included to handle graph classification task.

To prevent overfitting, we employed two techniques. The first technique was Early Stopping, which monitored the model's performance on a validation set and halted training if the performance did not improve. This helped prevent the model from memorizing the training data and improved its generalization ability. The second technique was the Reduce Learning Rate on Plateau scheduler, which automatically reduced the learning rate if the model's performance plateaued during training. This fine-tuning of the learning rate ensured better convergence and avoided overshooting the optimal solution.

For consistency and effective information processing, we set the hidden dimension to 512 for all graph-based approaches. Regarding hyperparameters, we adopted default values based on the specific graph topology. For example, the FastGTN model consisted of three FastGTN layers and two FastGT layers, while the GAT model utilized two GATv2Conv layers. These hyperparameters were chosen based on their effectiveness in capturing relevant graph patterns and achieving good performance on our specific tasks.

To account for the potential influence of metric values and cluster sets, we trained all approaches on three different cluster sets. This approach allowed us to evaluate the models' performance across various scenarios and mitigate the impact of specific clusters set configurations. We then averaged the resulting metric values to obtain a more robust evaluation of the models' performance.

C. Detailed results of the study

This section presents detailed results for proposed approaches assessed on both open-domain (refer to Table 6) and closed-domain (refer to Table 7) dialogue datasets. The evaluation employs the $Recall@k$ metric, $k \in \{1, 3, 5, 10\}$.

D. Resources

One NVIDIA GeForce GTX 1080 Ti was required for the graph-based approaches, and four such graphics cards were required for the gradient boosting approach.

Samples from the graph nodes, two-stage clustering method			
User cluster #1	**User cluster #2**	**Dialogue system cluster #1**	**Dialogue system cluster #2**
Can I please have the phone number and address for that place?	Yes, please book a table for 4 people at 12:15 on Tuesday.	Thank you for contacting us and have a nice day.	I'm sorry. There is still no availability. Would you like to try a different hotel then?
Could you tell me the price, address and phone number?	Book it for the same number of people at 14:30 on the same day.	Thank you for using Cambridge Town Info centre, have a great day!	I'm sorry, there were no rooms available. Perhaps you'd like to find another hotel?
How about Jesus Green Outdoor pool. Could I have their address and phone number?	I don't have a preference for food type. I do need reservations for 8 at 12:00 on Thursday.	You're very welcome, enjoy your time in Cambridge!	I'm sorry, there are no rooms available for that length of stay. Could you shorten your stay or book a different day possibly?
Yes, please. Can I get the address and phone number for the one you recommend?	Can you see if there's anything at 20:00?	Great! I'm happy to help. Goodbye!	The booking for the Acorn Guest House was unsuccessful. Would you like me to look for another hotel for you?
Do you have there phone number?	La Mimosa sounds good. Can your reserve me a table for 1 on Saturday at 11:15?	I'm glad I was able to help. Please call back if you have any more questions!	I am sorry, but the Leverton House was not available for your party on Tuesday. Would you like me to look for another hotel?

Table 5: Samples from the user and dialogue system MultiWOZ 2.2 graph nodes.

Approach	# Parameters	Relative Training Time	Datasets	PersonaChat			DailyDialog		
			# Clusters, First Stage	200	400	800	200	400	800
			# Clusters, Second Stage	30	60	120	30	60	120
Markov Chain	10K	0.13	Recall@1	29.41 ± 1.866	26.19 ± 0.802	20.70 ± 0.728	27.55 ± 0.821	**23.62 ± 0.683**	18.21 ± 0.620
			Recall@3	46.97 ± 2.250	37.48 ± 2.008	29.74 ± 0.529	44.22 ± 0.991	35.97 ± 2.419	28.01 ± 1.214
			Recall@5	57.84 ± 2.841	44.94 ± 3.053	35.10 ± 0.977	54.93 ± 0.849	44.13 ± 3.537	34.39 ± 1.044
			Recall@10	75.78 ± 2.111	58.09 ± 3.242	45.35 ± 1.905	72.94 ± 0.747	58.41 ± 3.998	45.32 ± 0.779
Message Passing	82M + 3.7M	0.47	Recall@1	**36.04 ± 0.348**	**30.27 ± 1.175**	**25.87 ± 0.559**	**29.03 ± 1.761**	23.59 ± 0.235	19.63 ± 0.315
			Recall@3	**54.07 ± 1.350**	44.95 ± 0.745	**40.33 ± 0.598**	**52.43 ± 2.647**	**42.67 ± 0.601**	34.57 ± 0.428
			Recall@5	**65.03 ± 1.445**	53.75 ± 0.247	**47.71 ± 0.843**	**65.09 ± 2.646**	**53.28 ± 0.912**	43.24 ± 0.236
			Recall@10	**80.32 ± 1.095**	66.20 ± 0.570	**57.92 ± 0.738**	**81.98 ± 2.050**	**69.04 ± 1.099**	**56.62 ± 0.681**
CatBoost	82M + 2.2M	1.00	Recall@1	**36.95 ± 1.848**	**29.85 ± 1.499**	**25.20 ± 0.548**	**31.66 ± 0.873**	**24.86 ± 0.653**	**20.94 ± 0.479**
			Recall@3	**54.35 ± 1.114**	**47.21 ± 0.604**	**40.01 ± 0.612**	**53.99 ± 0.664**	**43.39 ± 1.349**	**35.88 ± 0.388**
			Recall@5	**64.76 ± 0.868**	**55.21 ± 0.528**	**47.06 ± 0.728**	**66.13 ± 0.998**	**53.55 ± 1.378**	**44.27 ± 0.730**
			Recall@10	**81.17 ± 1.130**	**68.19 ± 0.481**	**57.99 ± 0.640**	**82.89 ± 1.061**	**68.41 ± 1.407**	**56.91 ± 0.808**
FastGTN	82M + 1.9M	0.49	Recall@1	**36.12 ± 1.306**	**29.52 ± 0.330**	**25.65 ± 1.537**	26.76 ± 0.655	23.05 ± 0.222	19.03 ± 0.877
			Recall@3	**55.77 ± 3.387**	**46.03 ± 0.623**	**38.66 ± 1.390**	51.21 ± 0.646	**42.16 ± 0.231**	**34.85 ± 0.921**
			Recall@5	**66.78 ± 3.085**	**54.25 ± 0.463**	**46.01 ± 1.420**	64.08 ± 0.764	**52.52 ± 0.710**	**43.79 ± 1.103**
			Recall@10	**82.17 ± 1.389**	66.64 ± 0.372	**56.40 ± 1.035**	81.47 ± 0.100	**67.69 ± 1.669**	**58.00 ± 0.957**
Encoder	82M	0.50	Recall@1	18.29 ± 0.957	13.94 ± 0.518	12.70 ± 3.887	24.01 ± 1.029	18.82 ± 1.152	15.07 ± 0.454
			Recall@3	36.50 ± 2.127	26.48 ± 1.166	21.15 ± 4.063	43.84 ± 0.395	35.51 ± 1.567	27.71 ± 0.462
			Recall@5	49.74 ± 3.656	33.79 ± 1.508	26.72 ± 4.058	54.70 ± 0.445	45.18 ± 1.918	34.87 ± 0.603
			Recall@10	69.26 ± 2.075	49.57 ± 4.880	35.83 ± 4.246	73.12 ± 0.440	60.30 ± 1.816	46.98 ± 1.129
ConveRT	46M	0.36	Recall@1	17.98 ± 0.496	14.10 ± 0.351	10.48 ± 0.439	22.40 ± 1.199	17.58 ± 0.218	13.90 ± 0.163
			Recall@3	40.37 ± 2.252	31.21 ± 3.709	22.04 ± 0.611	44.66 ± 2.404	35.70 ± 0.935	28.65 ± 0.560
			Recall@5	53.19 ± 1.405	39.52 ± 3.689	31.60 ± 4.283	57.32 ± 2.757	46.37 ± 1.280	36.74 ± 0.662
			Recall@10	70.01 ± 1.669	55.19 ± 4.072	45.15 ± 3.978	76.60 ± 3.057	62.95 ± 1.254	49.80 ± 0.877
ConveRT MAP	46M + 2M	0.78	Recall@1	22.94 ± 2.473	21.57 ± 1.862	7.32 ± 1.140	21.48 ± 1.312	14.63 ± 1.954	10.92 ± 1.544
			Recall@3	41.53 ± 2.066	34.71 ± 2.828	17.03 ± 1.489	44.90 ± 1.538	32.93 ± 1.342	25.41 ± 1.079
			Recall@5	53.11 ± 2.175	44.16 ± 0.737	23.53 ± 3.343	58.23 ± 2.394	44.22 ± 2.117	34.73 ± 2.358
			Recall@10	70.72 ± 1.33	59.42 ± 1.342	35.24 ± 2.083	77.42 ± 2.89	61.85 ± 3.192	47.57 ± 2.302

Table 6: The experimental results of the various intent prediction approaches on the open domain datasets. The training time of the models was counted from the start of training until the Early Stopping. To ensure stability of results, all approaches were trained on 3 different sets of clusters and the resulting metrics were averaged.

Approach	# Parameters	Relative Training Time	Dataset			MultiWOZ			FoCus			Taskmaster		
			# Clusters First Stage	# Clusters Second Stage	Metric	User	Dialog System	All Metric	User	Dialog System	All Metric	User	Dialog System	All Metric
Markov Chain	10K	0.13	200	30	Recall@1	29.53 ± 0.571	46.45 ± 0.628	37.99 ± 0.599	30.55 ± 1.579	28.45 ± 0.537	29.50 ± 0.654	31.04 ± 0.196	32.20 ± 0.481	31.73 ± 0.282
					Recall@3	54.57 ± 0.452	76.47 ± 1.220	65.52 ± 0.469	46.23 ± 1.272	47.41 ± 2.754	46.82 ± 1.920	52.54 ± 0.907	55.42 ± 0.659	54.06 ± 0.784
					Recall@5	68.21 ± 0.959	85.71 ± 0.425	76.96 ± 0.269	58.17 ± 1.173	57.81 ± 2.367	57.99 ± 0.601	64.91 ± 0.528	67.13 ± 0.806	66.06 ± 0.660
					Recall@10	85.58 ± 1.086	93.66 ± 0.077	89.62 ± 0.564	75.24 ± 1.178	74.92 ± 2.595	75.08 ± 0.750	82.65 ± 0.166	83.77 ± 0.731	83.24 ± 0.306
			400	60	Recall@1	22.31 ± 3.217	36.22 ± 3.511	29.26 ± 3.242	28.52 ± 0.855	25.71 ± 0.820	27.12 ± 0.825	24.39 ± 0.273	25.15 ± 0.353	24.81 ± 0.313
					Recall@3	42.28 ± 2.099	63.92 ± 2.011	53.10 ± 2.054	44.19 ± 0.702	40.97 ± 0.398	42.58 ± 0.549	44.83 ± 1.263	45.38 ± 0.068	45.21 ± 0.624
					Recall@5	54.14 ± 1.252	77.15 ± 0.218	65.64 ± 0.711	50.77 ± 0.724	49.16 ± 0.671	49.97 ± 0.621	56.77 ± 1.024	55.28 ± 0.058	56.10 ± 0.523
					Recall@10	69.45 ± 0.954	87.47 ± 0.254	78.46 ± 0.405	63.19 ± 0.506	62.01 ± 0.950	62.60 ± 0.230	73.39 ± 0.862	70.45 ± 0.672	71.95 ± 0.639
			800	120	Recall@1	11.66 ± 0.882	20.33 ± 1.713	15.99 ± 0.709	27.16 ± 1.168	23.66 ± 0.825	25.41 ± 0.172	20.34 ± 0.246	22.32 ± 0.161	21.36 ± 0.165
					Recall@3	25.17 ± 1.081	41.83 ± 1.443	33.50 ± 0.310	36.22 ± 1.823	37.49 ± 0.556	36.86 ± 0.921	38.11 ± 0.158	40.75 ± 0.324	39.51 ± 0.089
					Recall@5	35.16 ± 1.500	55.53 ± 1.545	45.35 ± 0.086	42.56 ± 3.475	44.49 ± 0.898	43.53 ± 1.704	48.67 ± 0.589	49.33 ± 0.133	49.07 ± 0.230
					Recall@10	51.62 ± 1.596	75.63 ± 1.194	63.62 ± 0.596	52.75 ± 1.184	53.80 ± 0.773	53.27 ± 0.457	63.27 ± 0.786	61.89 ± 0.343	62.62 ± 0.253
Message Passing	82M + 3.7M	0.47	200	30	Recall@1	**34.90 ± 0.555**	**58.19 ± 2.035**	**46.55 ± 1.288**	**42.13 ± 0.525**	**43.62 ± 1.067**	**42.88 ± 0.721**	**45.71 ± 0.858**	**55.25 ± 0.794**	**50.57 ± 0.634**
					Recall@3	**62.72 ± 1.239**	**86.00 ± 0.224**	**74.36 ± 0.533**	**63.70 ± 2.667**	**67.02 ± 0.774**	**65.36 ± 1.719**	**69.90 ± 1.087**	**77.78 ± 0.857**	**73.89 ± 0.632**
					Recall@5	**74.61 ± 1.222**	**92.65 ± 0.106**	**83.63 ± 0.558**	**73.30 ± 3.030**	**76.49 ± 0.420**	**74.90 ± 1.689**	**80.38 ± 0.635**	**86.67 ± 0.547**	**83.55 ± 0.354**
					Recall@10	**88.73 ± 1.363**	**97.62 ± 0.202**	**93.17 ± 0.758**	**86.22 ± 3.028**	**88.08 ± 0.529**	**87.15 ± 1.709**	92.18 ± 0.233	**95.06 ± 0.188**	**93.63 ± 0.184**
			400	60	Recall@1	**26.05 ± 0.392**	**47.89 ± 1.306**	**36.97 ± 0.467**	**38.85 ± 0.161**	**40.27 ± 0.219**	**39.56 ± 0.185**	**38.00 ± 0.843**	**45.72 ± 0.135**	**41.90 ± 0.486**
					Recall@3	**49.39 ± 0.467**	**76.11 ± 0.485**	**62.75 ± 0.349**	**56.02 ± 1.663**	**62.17 ± 1.017**	**59.10 ± 0.886**	62.07 ± 0.694	**68.51 ± 0.567**	**65.36 ± 0.257**
					Recall@5	**60.59 ± 0.322**	**85.61 ± 0.910**	**73.10 ± 0.300**	**65.81 ± 2.381**	**70.51 ± 0.841**	**68.16 ± 0.770**	72.76 ± 0.389	**77.50 ± 0.614**	75.17 ± 0.221
					Recall@10	**74.59 ± 0.581**	**93.92 ± 0.429**	**84.25 ± 0.204**	**77.54 ± 2.479**	**80.49 ± 0.814**	**79.01 ± 1.009**	86.09 ± 0.202	**88.19 ± 0.564**	87.15 ± 0.366
			800	120	Recall@1	14.11 ± 0.535	**28.34 ± 0.763**	**21.22 ± 0.647**	**36.57 ± 0.897**	**34.31 ± 0.725**	**35.44 ± 0.087**	**33.34 ± 0.291**	**40.67 ± 0.552**	**37.02 ± 0.391**
					Recall@3	**30.81 ± 0.926**	**53.77 ± 1.903**	**42.29 ± 1.034**	50.56 ± 0.400	**54.72 ± 0.834**	**52.64 ± 0.237**	**54.73 ± 0.458**	**62.35 ± 0.399**	**58.61 ± 0.430**
					Recall@5	41.20 ± 0.969	**66.67 ± 0.826**	**53.94 ± 0.687**	**59.18 ± 2.513**	**62.69 ± 1.076**	**60.93 ± 1.427**	**64.69 ± 0.466**	**71.26 ± 0.193**	**68.03 ± 0.287**
					Recall@10	57.60 ± 0.458	**84.63 ± 0.174**	71.11 ± 0.299	**72.27 ± 0.385**	**72.55 ± 0.971**	**72.41 ± 0.676**	**77.49 ± 0.415**	**81.73 ± 0.318**	79.64 ± 0.169
CatBoost	82M + 2.2M	1.00	200	30	Recall@1	**35.63 ± 0.854**	**58.35 ± 0.535**	**46.99 ± 0.684**	40.35 ± 0.644	**44.08 ± 0.416**	**42.22 ± 0.349**	**45.02 ± 0.118**	**54.25 ± 1.005**	**49.73 ± 0.485**
					Recall@3	**62.84 ± 0.791**	84.69 ± 0.929	**73.76 ± 0.773**	**63.31 ± 0.348**	**67.37 ± 0.347**	**65.34 ± 0.230**	69.32 ± 0.431	**77.14 ± 0.503**	**73.27 ± 0.269**
					Recall@5	**75.09 ± 0.474**	**92.04 ± 0.588**	**83.56 ± 0.215**	**72.57 ± 0.207**	**76.91 ± 0.156**	**74.74 ± 0.131**	80.02 ± 0.486	**86.48 ± 0.250**	**83.27 ± 0.125**
					Recall@10	**89.94 ± 0.042**	**97.28 ± 0.203**	**93.61 ± 0.122**	**86.62 ± 0.263**	**88.01 ± 0.313**	**87.32 ± 0.094**	91.92 ± 0.165	**95.07 ± 0.308**	93.50 ± 0.086
			400	60	Recall@1	23.27 ± 0.905	42.86 ± 0.877	33.07 ± 0.872	**37.82 ± 1.283**	38.59 ± 0.588	38.20 ± 0.867	**38.57 ± 0.343**	44.50 ± 0.307	41.57 ± 0.186
					Recall@3	47.45 ± 1.198	73.33 ± 1.297	60.39 ± 1.028	**56.14 ± 0.933**	58.93 ± 0.397	57.53 ± 0.610	62.37 ± 0.322	67.51 ± 0.317	65.00 ± 0.226
					Recall@5	**59.42 ± 1.283**	83.83 ± 0.808	71.63 ± 0.982	**66.23 ± 1.501**	67.42 ± 0.430	**66.83 ± 0.599**	72.99 ± 0.415	76.48 ± 0.428	74.77 ± 0.310
					Recall@10	**74.13 ± 0.886**	92.33 ± 0.256	83.23 ± 0.450	**78.27 ± 2.177**	78.69 ± 0.419	**78.48 ± 1.023**	86.21 ± 0.270	87.22 ± 0.274	86.72 ± 0.230
			800	120	Recall@1	**15.38 ± 0.677**	24.74 ± 0.251	20.06 ± 0.266	34.99 ± 0.499	**33.45 ± 0.830**	34.22 ± 0.526	32.38 ± 0.235	39.45 ± 0.365	35.93 ± 0.298
					Recall@3	**31.48 ± 0.461**	50.00 ± 0.806	**40.74 ± 0.577**	**53.18 ± 0.296**	52.33 ± 0.808	**52.76 ± 0.543**	53.66 ± 0.378	61.28 ± 0.349	57.54 ± 0.329
					Recall@5	**42.58 ± 0.030**	64.44 ± 0.980	**53.51 ± 0.492**	**59.86 ± 0.304**	59.97 ± 0.670	**59.92 ± 0.484**	63.81 ± 0.378	69.78 ± 0.282	66.86 ± 0.325
					Recall@10	**59.21 ± 1.137**	82.62 ± 0.932	**70.91 ± 1.021**	70.19 ± 0.292	70.00 ± 0.637	70.10 ± 0.423	76.27 ± 0.404	79.91 ± 0.181	78.11 ± 0.265
FastGTN	82M + 1.9M	0.49	200	30	Recall@1	**35.16 ± 1.317**	**57.61 ± 0.944**	**46.39 ± 0.349**	**41.07 ± 0.995**	**43.20 ± 0.560**	**42.13 ± 0.219**	**45.33 ± 0.248**	**55.07 ± 0.304**	**50.29 ± 0.201**
					Recall@3	**62.87 ± 0.384**	85.39 ± 0.176	**74.13 ± 0.239**	**61.78 ± 3.246**	**67.35 ± 1.161**	**64.57 ± 1.701**	**70.53 ± 0.594**	**77.07 ± 0.523**	**73.85 ± 0.364**
					Recall@5	**75.17 ± 0.625**	**92.11 ± 0.453**	**83.64 ± 0.371**	**71.38 ± 3.411**	**76.95 ± 1.666**	**74.17 ± 2.010**	**81.41 ± 0.564**	**86.03 ± 0.839**	**83.74 ± 0.605**
					Recall@10	89.01 ± 0.221	97.05 ± 0.338	93.03 ± 0.094	**86.25 ± 3.279**	**88.41 ± 1.255**	**87.33 ± 1.573**	**92.84 ± 0.254**	**95.05 ± 0.380**	**93.95 ± 0.267**
			400	60	Recall@1	**26.47 ± 0.782**	**47.25 ± 2.594**	**36.86 ± 1.103**	37.06 ± 1.282	38.65 ± 0.781	37.86 ± 0.989	**38.78 ± 0.828**	**45.61 ± 0.626**	**42.22 ± 0.335**
					Recall@3	47.88 ± 0.294	**76.48 ± 0.527**	**62.18 ± 0.350**	52.15 ± 1.400	60.21 ± 0.459	56.18 ± 0.792	**63.17 ± 0.366**	**68.59 ± 0.553**	**65.95 ± 0.341**
					Recall@5	58.73 ± 0.529	**85.86 ± 0.453**	72.29 ± 0.308	60.53 ± 1.321	68.95 ± 0.677	64.74 ± 0.548	**74.09 ± 0.647**	**77.78 ± 0.221**	**75.98 ± 0.387**
					Recall@10	**74.28 ± 1.041**	**94.17 ± 0.210**	**84.23 ± 0.435**	**73.82 ± 3.712**	79.23 ± 0.417	**76.52 ± 1.844**	**87.30 ± 0.175**	**88.44 ± 0.043**	**87.88 ± 0.103**
			800	120	Recall@1	**14.89 ± 0.514**	**27.94 ± 1.087**	**21.41 ± 0.504**	35.15 ± 0.487	**34.29 ± 0.319**	34.72 ± 0.293	32.32 ± 0.590	**40.51 ± 0.440**	**36.43 ± 0.444**
					Recall@3	**30.77 ± 1.093**	**53.30 ± 1.497**	**42.04 ± 0.276**	**52.00 ± 2.358**	**54.08 ± 0.224**	**53.04 ± 1.070**	**54.77 ± 0.494**	**62.29 ± 0.348**	**58.59 ± 0.406**
					Recall@5	**41.62 ± 1.024**	**67.33 ± 1.797**	**54.47 ± 0.702**	**58.89 ± 2.474**	**62.49 ± 0.457**	**60.69 ± 1.328**	**65.09 ± 0.692**	**71.54 ± 0.342**	**68.37 ± 0.457**
					Recall@10	**58.33 ± 0.976**	**84.94 ± 0.788**	**71.63 ± 0.171**	70.73 ± 0.667	**72.77 ± 0.121**	**71.75 ± 0.386**	**77.90 ± 0.264**	**82.73 ± 0.762**	**80.34 ± 0.425**
Encoder	82M	0.50	200	30	Recall@1	12.81 ± 1.085	35.03 ± 1.449	23.92 ± 0.806	19.09 ± 1.049	34.12 ± 2.074	26.60 ± 0.641	25.34 ± 0.486	27.09 ± 0.058	26.22 ± 0.272
					Recall@3	29.16 ± 0.588	65.97 ± 1.024	47.57 ± 0.219	33.09 ± 1.226	56.25 ± 0.604	44.67 ± 0.334	43.10 ± 0.895	46.60 ± 0.260	44.85 ± 0.321
					Recall@5	39.52 ± 0.553	78.10 ± 0.808	58.81 ± 0.405	42.40 ± 1.104	65.85 ± 0.147	54.13 ± 0.574	52.59 ± 0.716	56.24 ± 0.552	54.41 ± 0.089
					Recall@10	57.26 ± 2.562	90.24 ± 0.331	73.75 ± 1.164	61.47 ± 3.147	80.21 ± 0.358	70.84 ± 1.542	63.28 ± 0.774	66.26 ± 0.811	64.77 ± 0.088
			400	60	Recall@1	8.21 ± 0.385	23.84 ± 2.136	16.02 ± 0.900	17.47 ± 0.585	29.48 ± 0.421	23.47 ± 0.499	17.60 ± 0.244	20.78 ± 0.098	19.19 ± 0.077
					Recall@3	20.18 ± 0.372	49.38 ± 3.682	34.78 ± 1.666	29.04 ± 0.519	47.93 ± 0.711	38.49 ± 0.176	33.47 ± 0.202	38.09 ± 0.257	35.78 ± 0.227
					Recall@5	28.58 ± 0.280	62.78 ± 1.623	45.68 ± 0.674	35.73 ± 0.372	56.03 ± 1.055	45.88 ± 0.390	41.50 ± 0.211	46.98 ± 0.265	44.24 ± 0.231
					Recall@10	41.73 ± 0.728	77.61 ± 0.685	59.67 ± 0.370	47.78 ± 1.994	68.13 ± 0.739	57.96 ± 1.186	52.82 ± 0.298	57.66 ± 0.193	55.24 ± 0.233
			800	120	Recall@1	4.87 ± 0.198	14.31 ± 0.792	9.59 ± 0.335	15.80 ± 0.587	24.45 ± 0.326	20.12 ± 0.243	12.78 ± 0.051	15.59 ± 0.053	14.18 ± 0.016
					Recall@3	11.66 ± 0.106	30.44 ± 0.069	21.05 ± 0.036	26.60 ± 0.422	41.13 ± 0.504	33.86 ± 0.431	25.30 ± 0.077	28.76 ± 0.082	27.03 ± 0.045
					Recall@5	17.16 ± 0.304	41.48 ± 0.686	29.32 ± 0.378	31.80 ± 0.400	48.65 ± 0.229	40.22 ± 0.085	31.93 ± 0.220	35.57 ± 0.130	33.75 ± 0.157
					Recall@10	27.55 ± 0.699	59.19 ± 1.416	43.37 ± 0.905	40.02 ± 0.235	58.41 ± 0.678	49.22 ± 0.287	41.26 ± 0.223	44.94 ± 0.275	43.10 ± 0.240
ConveRT	46M	0.36	200	30	Recall@1	10.15 ± 0.377	25.14 ± 0.810	17.65 ± 0.591	18.04 ± 0.886	31.92 ± 0.057	24.98 ± 0.464	22.68 ± 0.286	33.84 ± 0.219	28.26 ± 0.034
					Recall@3	25.75 ± 0.240	54.44 ± 1.246	40.09 ± 0.716	33.15 ± 0.988	57.11 ± 0.820	45.13 ± 0.846	43.55 ± 0.669	57.85 ± 0.820	50.70 ± 0.352
					Recall@5	37.19 ± 1.667	67.88 ± 1.075	52.53 ± 1.370	42.88 ± 0.904	69.67 ± 0.131	56.27 ± 0.511	54.15 ± 0.307	68.25 ± 1.180	61.20 ± 0.507
					Recall@10	58.16 ± 0.837	84.31 ± 0.648	71.23 ± 0.565	58.47 ± 0.608	83.79 ± 0.291	71.13 ± 0.172	69.69 ± 0.170	79.26 ± 0.910	74.48 ± 0.450
			400	60	Recall@1	5.59 ± 0.126	18.63 ± 0.993	12.11 ± 0.472	17.10 ± 0.520	27.27 ± 0.725	22.19 ± 0.240	16.93 ± 0.426	23.08 ± 0.378	20.01 ± 0.236
					Recall@3	15.99 ± 0.087	41.46 ± 1.874	28.73 ± 0.976	29.60 ± 0.472	49.03 ± 0.659	39.31 ± 0.250	33.77 ± 0.510	42.13 ± 0.713	37.95 ± 0.475
					Recall@5	24.17 ± 0.461	54.45 ± 0.700	39.31 ± 0.576	36.92 ± 0.579	60.13 ± 1.242	48.52 ± 0.682	43.71 ± 0.512	52.55 ± 0.745	48.13 ± 0.488
					Recall@10	38.65 ± 0.254	70.45 ± 0.437	54.55 ± 0.111	49.12 ± 0.946	73.68 ± 0.857	61.40 ± 0.644	57.08 ± 0.839	65.90 ± 0.706	61.49 ± 0.558
			800	120	Recall@1	3.18 ± 0.222	10.16 ± 0.294	6.67 ± 0.214	14.97 ± 0.181	23.97 ± 0.777	19.47 ± 0.455	12.74 ± 0.110	18.15 ± 0.225	15.45 ± 0.167
					Recall@3	8.82 ± 0.419	23.89 ± 0.701	16.36 ± 0.265	25.90 ± 0.111	42.67 ± 0.800	34.29 ± 0.397	26.28 ± 0.290	35.31 ± 0.204	30.79 ± 0.209
					Recall@5	14.26 ± 0.581	33.92 ± 1.125	24.09 ± 0.393	31.85 ± 0.110	52.09 ± 1.034	41.97 ± 0.570	33.85 ± 0.395	44.28 ± 0.149	39.07 ± 0.261
					Recall@10	24.57 ± 1.037	49.53 ± 0.641	37.05 ± 0.212	41.63 ± 0.507	64.48 ± 0.776	53.06 ± 0.451	45.28 ± 0.440	56.32 ± 0.143	50.80 ± 0.292
ConveRT MAP	46M + 2M	0.78	200	30	Recall@1	20.23 ± 1.506	43.35 ± 1.198	31.79 ± 1.251	30.34 ± 0.894	32.60 ± 1.291	31.47 ± 0.741	33.45 ± 0.856	41.42 ± 1.430	37.44 ± 0.887
					Recall@3	46.31 ± 1.720	76.19 ± 1.257	61.25 ± 1.496	51.81 ± 1.345	56.00 ± 1.828	53.90 ± 0.674	60.13 ± 1.431	68.85 ± 1.586	64.49 ± 1.485
					Recall@5	59.38 ± 1.720	88.06 ± 0.903	73.72 ± 1.129	62.12 ± 1.478	68.79 ± 1.407	65.46 ± 0.747	72.76 ± 0.134	81.92 ± 0.561	77.34 ± 0.215
					Recall@10	81.06 ± 2.524	96.29 ± 0.959	88.67 ± 1.636	78.70 ± 1.620	83.05 ± 1.426	80.87 ± 1.297	86.36 ± 0.303	91.08 ± 0.040	88.72 ± 0.13
			400	60	Recall@1	15.42 ± 1.109	26.30 ± 1.121	20.86 ± 1.014	15.36 ± 0.950	22.72 ± 1.109	19.04 ± 0.647	26.77 ± 0.368	31.92 ± 0.285	29.35 ± 0.284
					Recall@3	34.96 ± 1.127	58.43 ± 1.108	46.69 ± 0.927	40.88 ± 2.110	44.01 ± 1.679	42.45 ± 1.967	51.30 ± 0.903	55.44 ± 0.464	53.37 ± 0.595
					Recall@5	45.20 ± 1.406	72.78 ± 1.171	58.99 ± 1.593	52.80 ± 1.309	54.15 ± 1.572	53.47 ± 1.343	62.48 ± 0.664	66.89 ± 1.362	64.68 ± 0.736
					Recall@10	61.97 ± 1.676	88.25 ± 1.822	75.11 ± 1.743	68.21 ± 1.150	69.21 ± 1.230	68.71 ± 1.044	77.60 ± 0.517	80.10 ± 1.426	78.85 ± 0.492
			800	120	Recall@1	6.36 ± 0.780	12.08 ± 0.227	9.22 ± 0.503	21.40 ± 0.468	15.37 ± 1.025	18.39 ± 0.284	20.61 ± 1.165	23.86 ± 0.101	22.23 ± 0.633
					Recall@3	16.80 ± 0.896	31.66 ± 0.391	24.23 ± 0.548	32.94 ± 0.502	33.17 ± 1.216	33.06 ± 0.358	39.19 ± 1.549	46.51 ± 0.430	42.85 ± 0.990
					Recall@5	23.36 ± 1.242	44.92 ± 0.197	35.64 ± 0.603	40.03 ± 0.331	42.67 ± 1.422	41.35 ± 0.591	49.93 ± 1.419	56.96 ± 0.856	53.44 ± 1.138
					Recall@10	42.29 ± 1.925	70.35 ± 0.609	56.32 ± 0.675	56.13 ± 0.391	56.75 ± 2.076	56.35 ± 1.221	64.73 ± 0.214	71.12 ± 2.216	67.93 ± 1.215

Table 7: The experimental results of the various intent prediction approaches on the closed domain datasets. The training time of the models was counted from the start of training until the Early Stopping. The all metric is the average of the user metric and the dialogue system metric. To ensure stability of results, all approaches were trained on 3 different sets of clusters and the resulting metrics were averaged.