Applied Intelligence (2024) 54:5278–5305
https://doi.org/10.1007/s10489-024-05455-5

Revisiting clustering for efficient unsupervised dialogue structure induction

Maarten De Raedt[1,2] · Fréderic Godin[1] · Chris Develder[2] · Thomas Demeester[2]

Accepted: 7 April 2024 / Published online: 18 April 2024

Abstract

In the development of a task-oriented dialogue system, defining the dialogue structure is a time-consuming task. Hence, several works have looked into automatically inferring it from data, e.g., actual conversations between a customer and a support agent. To recover such dialogue structure, recent methods based on discrete variational models learn to jointly encode and cluster utterances in dialogue states, but (i) represent utterances by only considering preceding dialogue context, and (ii) are slow to train since they are optimized with a compute-expensive *decoding* objective. We revisit and improve upon an existing efficient pipeline approach, commonly adopted as a baseline, that first encodes utterances and then clusters them with k-means to induce the dialogue structure. However, the existing approach represents utterances as bag-of-words or skip-thought vectors, which have been shown to perform poorly in semantic similarity tasks, and without considering dialogue context. We therefore first investigate the use of more powerful transformer-based encoders for encoding utterances. Next, we propose ELLoDAR, a method for learning representations that capture both preceding and subsequent dialogue context, inspired by word-to-vec training strategies. ELLoDAR is efficient since representations are learned directly in the encoding space by finetuning just a *single* linear layer on top of a *frozen* sentence encoder with a *vector-to-vector* regression training objective. Extensive experiments on representative datasets for dialogue structure induction (SimDial, Schema Guided Dialogues, DSTC2, and CamRest676) demonstrate that in terms of effectiveness to induce the correct dialogue structure, (i) clustering utterances represented by transformed-based encoders improves recent joint models by 13%–32% on standard cluster metrics, and (ii) clustering ELLoDAR's representations yields additional improvements ranging from +20% to +26%, with speedups of $\times$ **10–10^4** compared to the recent joint models.

Keywords Information extraction · Dialogue structure induction · Efficient NLP · Sentence representation learning · Text clustering

1 Introduction

Recently, systems for conversational modeling have enjoyed a lot of attention, including intent classification [1–3], dialogue state tracking [4–6], and slot filling [3, 7, 8]. Such systems have been used to automate customer service in the insurance and public transportation industries [9], to provide information in healthcare [10], and to perform tasks such as legal case retrieval [11] and conversational recommendation [12].

Designing conversational agents, however, requires readily available annotated data. While companies often have access to an abundance of unlabeled dialogues, such as those exchanged between their customers and support agents, annotating them to develop conversational agents remains costly. Consequently, dialogue structure induction (DSI), aims to unsupervisedly recover the latent conversational structure from a set of task-oriented user-agent dialogues. Figure 1 shows an example of such a graph in which the nodes represent the distinct user and agent (system) dialogue

✉ Maarten De Raedt
maarten.deraedt@ugent.be

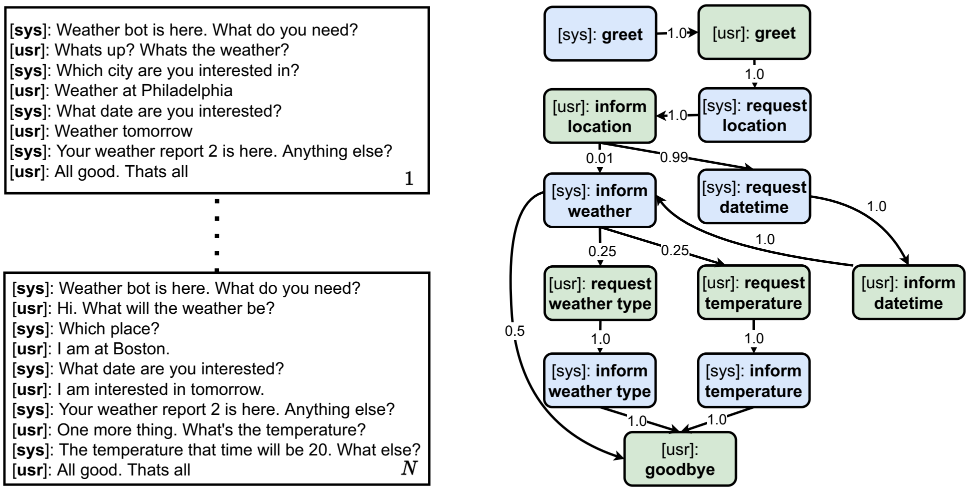
Fréderic Godin
frederic.godin@sinch.com

Chris Develder
chris.develder@ugent.be

Thomas Demeester
thomas.demeester@ugent.be

[1] Sinch Chatlayer, Ankerrui 9, Antwerpen 2000, Belgium

[2] IDLab, Ghent University - imec, Technologiepark 126, Zwijnaarde B-9052, Belgium



[sys]: Weather bot is here. What do you need?
[usr]: Whats up? Whats the weather?
[sys]: Which city are you interested in?
[usr]: Weather at Philadelphia
[sys]: What date are you interested?
[usr]: Weather tomorrow
[sys]: Your weather report 2 is here. Anything else?
[usr]: All good. Thats all
1

[sys]: Weather bot is here. What do you need?
[usr]: Hi. What will the weather be?
[sys]: Which place?
[usr]: I am at Boston.
[sys]: What date are you interested?
[usr]: I am interested in tomorrow.
[sys]: Your weather report 2 is here. Anything else?
[usr]: One more thing. What's the temperature?
[sys]: The temperature that time will be 20. What else?
[usr]: All good. Thats all
N

Fig. 1 *Dialogue Structure Induction* (DSI): the structure (*right*) is induced from the set of *N* user-system dialogues (*left*), with the distinct user and system states as nodes, and transition probabilities as edges. Illustration *based* on SimDial [27]

states, and the possible transitions and corresponding probabilities between successive states are denoted by the edges. A dialogue structure can compactly summarize an entire collection of dialogues, providing companies with relevant insights about their customers and agents and they thus give a solid starting point for designing conversational models.

In unsupervised DSI, utterances with similar conversational goals are first clustered into the same dialogue state, and the structure (transition probabilities from one state to another) can then be recovered either directly from the model's weights or by counting the number of transitions between successive states. Earlier work extended hidden Markov models to infer conversational graphs [13–15]. More recently, neural end-to-end models, e.g., DVRNN [16] and SVRNN [17], jointly learn to encode utterances and assign them to dialogue states. Yet, such neural models, (i) represent utterances by only considering the preceding dialogue context, (ii) require GPUs and tend to be slow at inducing dialogue structures as they are trained on a computationally expensive next turn *decoding* objective. Since DSI models embed the number of dialogue states in their architecture, they must be re-trained every time that number changes. It is thus important to induce dialogue structures efficiently, since in practice, users may need to experiment with different numbers of states to recover the optimal structure.

To address the weaknesses above, our work revisits and further builds upon the method of Gunasekara et al. [18, 19] that comprises two efficient steps in which utterances are first encoded into vectors and subsequently clustered. However, Gunasekara et al. [18, 19] represent utterances as bag-of-words or skip-thought [20] vectors, which have been shown to perform poorly in semantic similarity tasks [21, 22], and without considering dialogue context. In this work, we first demonstrate that encoding utterances by powerful transformer-based sentence encoders instead, already leads to improvements over recent joint models, in terms of both cluster metrics and being orders of magnitude faster at inducing the dialogue structure.

Next, we propose a highly efficient strategy to embed both preceding and *subsequent* dialogue context into utterance vector representations, called ELLoDAR (for "Efficiently Learnt Locally Dialogue Aware Representations"), which further boosts performance in terms of cluster metrics. We cluster ELLoDAR's representations to induce dialogue structure, and refer to the complete procedure as CELLoDAR. Regarding the aforementioned limitations of existing works, CELLoDAR (i) uses both preceding and subsequent context, (ii) can be trained on CPU within seconds, and thus (iii) makes determining the number of dialogue states up to four orders of magnitude faster than recent joint models.

To obtain dialogue-aware embeddings before clustering, ELLoDAR draws inspiration from the CBOW and skip-gram (so-called 'word-to-vec') models for learning word embeddings [23]: utterances with similar context windows, and context windows enclosing similar utterances, are represented closer to each other in the embedding space. ELLoDAR is efficient (trains within seconds on CPU) as it learns a *linear* transformation with a *vector-to-vector* regression training objective in the encoding space of a *frozen* pretrained encoder by exploiting a local, yet bidirectional, context window. By casting representation learning as vector-to-vector regression, ELLoDAR avoids the computational overhead incurred by decoding objectives, such as those used for training the joint DVRNN and SVRNN models.

Extensive experiments on 10 task-oriented domains spanning across the DSTC2 [24], CamRest676 [25, 26], SimDial [27] and Schema Guided dialogue [5] datasets, show that CELLoDAR, yields absolute improvements over recently proposed joint methods of 7%–74% in standard cluster metrics while being 10 to 10^4 times faster.

1.1 Research objective and contributions

Our objective is not to outperform existing approaches by merely developing increasingly more complex models. Rather, our goal is to attain state-of-the-art performance while being highly efficient compute-wise, thereby making it feasible to induce dialogue structures in practice. More specifically, we aim to obtain a model that (i) outperforms the more complex joint models, i.e., DVRNN [16] and SVRNN [17], as measured by standard cluster metrics, and (ii) is sufficiently lightweight for inducing dialogue structures on accessible and cheap computing resources such as CPUs (rather than requiring GPUs like such joint models).

We summarize our contributions as follows:

(1) We revisit the cluster baseline proposed in [18, 19], and demonstrate that clustering utterances encoded by transformer-based sentence encoders [22, 28] rather than by bag-of-words or skip-thoughts vectors, already outperforms the recent joint models for DSI [16, 17] in terms of inducing the correct dialogue structure, while being orders of magnitude faster.

(2) We contribute ELLoDAR, a highly efficient utterance representation learning approach that exploits local dialogue context to train linear transformations in the encoding space of a *frozen* sentence encoder using a vector-to-vector regression training objective. Clustering the ELLoDAR representations (referred to as CELLoDAR) is shown to outperform — by a large margin — the joint DVRNN and SVRNN models [16, 17] (while being orders of magnitude faster) as well as the improved transformer-based cluster baselines, on representative DSI datasets.

(3) Since there exists no common benchmark for DSI, we release[1] our modified datasets, evaluation, and models, which we hope will spur future research in the unexplored DSI task.

2 Related work

We summarize previous research on the relatively unexplored task of unsupervised dialogue structure induction. There are many variations to this task, including both supervised and unsupervised statistical methods that learn structures based on *dialogue acts*, as discussed in Section 2.1. However, we specifically focus on unsupervised dialogue structure induction for *task-oriented* dialogues, for which Section 2.2 reviews recent joint models based on neural and variational approaches, and compares them to our proposed approach. In addition, we discuss methods for structure learning based on unsupervised *slot extraction* in Section 2.3, which is a related but distinct task. Finally, Section 2.4 outlines the various applications for which dialogue structures have been used.

2.1 Unsupervised dialogue act modeling

Early work focused on structure modeling of dialogues based on categorizing utterances into high-level dialogue acts (e.g., question, statement, request, and acknowledgment) and then learning the structure (transitions) among these acts (states). In [29], utterances are manually annotated with dialogue acts, and the general discourse structure is then inferred using stochastic grammars. Since labeling dialogue text thus requires expensive annotation, focus shifted to *unsupervised* dialogue act learning. Crook et al. [30] use Dirichlet Process Mixtures to cluster utterances into dialogue acts, but their approach does not model structural information that captures transitions between different acts. Therefore, to both model acts and learning the structure among them, Ritter et al. [14] combine hidden Markov and topic models to identify general discourse structure (i.e., dialogue acts) and dialogue-specific topics in non-task-oriented conversations. Joty et al. [31] further improve the approach of [14] by expanding the set of sentence features used to estimate the hidden Markov model's act emission distribution to include the speaker, relative position, and sentence length in addition to unigrams. Similarly, the method of [32] uses hidden Markov models to model structural dependencies between dialogue acts, but instead estimate the act emission probabilities using Gaussian mixtures, enabling the use of real-valued sentence embeddings such as bag-of-words GloVe vectors to represent utterances, as opposed to discrete features [14, 31] such as, e.g., unigrams, and utterance length.

2.2 Unsupervised task-oriented dialogue structure induction

Rather than the aforementioned works on identifying high-level dialogue acts, another line of work focuses on the modeling of dialogue structures in *task-oriented* domains, with the aim of categorizing utterances into more fine-grained, task-specific intents. Early approaches, such as those of [13, 15], adopt hidden Markov models (HMMs) to cluster text spans in task-oriented dialogues into states and learn the dependencies between them. Zhai et al. [15] follow a similar approach to the above cited [14], but consider task-oriented dialogues, assuming that utterance words are generated from a mixture of topic models shared across all states rather than having a single model per state.

To better capture the highly non-linear dynamics in dialogues [33], recent solutions have shifted away from simple

[1] https://github.com/maarten-deraedt/efficient-unsupervised-dialogue-structure-induction

HMMs towards neural end-to-end models that jointly learn to encode and cluster utterances to induce task-oriented dialogue structures. Shi et al. [16] propose the use of Discrete Variational Recurrent Neural Networks (DVRNNs) to assign turns to discrete latent states, decoding the current turn from its predicted state and the *preceding* turns. Qiu et al. [17] extend DVRNNs to SVRNNs by adding structured attention [34] over its hidden states, enforcing a structural inductive bias that is more aligned with DSI. In [35], a modification of the DVRNN model that separates user and system utterances instead of treating them jointly is proposed, leading to more accurate assignment of system actions to states. However, the approach of [35] relies on weak supervision from database queries performed by a human at some point in the dialogue, whereas the unsupervised DVRNN and SVRNN models do not require such (weak) supervision. Rather than inducing dialogue structures in task-oriented domains, Xu et al. [36] induce them in an *open-domain* setting, using a combination of discrete variational models with graph neural networks to hierarchically discover different domains and then learning the structure within each domain. To obtain more easily interpretable structures, Sun et al. [37] propose an Edge-Enhanced Graph Auto-Encoder that induces *deterministic* dialogue structures.

Our work focuses on *unsupervised* induction of *non-deterministic* dialogue structures in *task-oriented* domains, given that transitions between dialogue states are inherently probabilistic. We thus focus on the same task as the DVRNN [16] and SVRNN [17] models that jointly learn to encode and cluster utterances. However, both those models (i) only consider preceding dialogue context and, because they are based on Variational Auto-Encoders optimized with a next turn decoding objective, they (ii) are slow to train, and (iii) are susceptible to the *posterior collapse* [38–40]. Posterior collapse occurs when the model relies solely on the decoder's auto-regressive properties to reconstruct inputs, thus bypassing the latent states altogether, which may result in utterances with the distinct conversational goal being erroneously assigned to the same state.

To address these limitations (i)–(iii), our work builds on the method of [18, 19] that comprises two efficient steps: utterances are first (1) encoded as vectors and then (2) clustered into dialogue states (e.g., using k-means). Clustering assigns utterances to states based on vector similarities rather than on an indirect *decoding objective*. However, the methods used in [18, 19] for representing utterances as vectors, such as bag-of-words and skip-thought vectors, are sub-optimal for semantic similarity tasks [21, 22]. Furthermore, since these bag-of-words or skip-thought vectors are not fine-tuned on task-specific dialogues, the approach of [18, 19] does not utilize dialogue context. Here, we first experiment with using more powerful transformer-based encoders like SBERT [22] and TOD-BERT [28] that are better suited for semantic similarity tasks. Then, we propose ELLoDAR as a method for obtaining task-specific contextual utterance representations by building upon an already pretrained transformer encoder, which is kept frozen, and subsequently learning a linear transformation on top of it with a *vector-to-vector* regression objective, using both preceding and subsequent context.

2.3 Unsupervised dialogue slot extraction

Similar to our current work, the methods discussed in Sections 2.1 and 2.2 induce dialogue structures by mapping *utterances* to states. In the related but different *slot-based* dialogue structure induction task, *words* or *subphrases* rather than *utterances* are mapped to states in task-oriented domains. To this end, Hudeček et al. [41] use weak supervision from rule-based parsers to identify potential slot candidates, which are then clustered into task-specific slots. Qiu et al. [42] employ transfer learning instead, using supervision from domains with available slot annotations to first train a model that detects slot boundaries. The obtained slot boundary detection model is then applied to unseen domains to identify slot candidates, which are subsequently clustered into states. Vukovic et al. [43] extend the transfer learning method of [42] by starting from the same slot boundary detection model, but using topological data analysis methods to increase the recall of the candidate slot extraction step. Rather than extracting slots through weak-supervision or transfer learning, the method of [44] extracts slots completely unsupervised by using self-supervised language models trained on the task-specific dialogues and unsupervised parsers to identify slot candidates, after which these are similarly clustered to obtain slot states.

2.4 Applications of dialogue structures

While in our current paper, we solely focus on structure induction as an information extraction task, the inferred dialogue structure may be further used for other applications. In particular, it can be used for (i) accelerating dialogue *policy learning* [16, 45, 46], (ii) more controllable and coherent dialogue agents, in open domain [36, 47] and domain-specific settings [48], (iii) response generation in multi-party dialogues [49], (iv) low-resource dialogue state tracking [37], and (v) zero-shot policy learning generalizing beyond a single domain [50].

3 Methodology

In Section 3.1, the DSI task is formalized. We specifically focus on recovering dialogue structures from *task-oriented* dialogues (Section 2.2), in which there are typically two parties who exchange utterances consecutively [5, 24–27, 51]. We will refer to the two parties in the dialogues as 'users' and 'systems' respectively, with the 'system' utterances generated by, e.g., a support agent in response to requests from a client ('user'). We describe the cluster-based approach of [18, 19] in Section 3.2, followed by our proposed ELLoDAR strategy to obtain utterance representations in Section 3.3.

3.1 Task formulation

We are given a set $\mathcal{D}$ containing N dialogues between users and system. Each dialogue $d \in \mathcal{D}$ is a sequence of n utterances, alternating between user utterances x^{U} and system utterances x^{S} (or vice versa): $\left[x_1^{\mathrm{U}}, x_2^{\mathrm{S}}, x_3^{\mathrm{U}}, \ldots, x_n^{\mathrm{S}}\right]$. Unsupervised dialogue structure induction aims to infer from $\mathcal{D}$ the conversational graph (V, E) with vertices V and edges E. To this end, those utterances that have a common conversational goal ('intent') are mapped onto a common dialogue state $v \in V$ across the corpus. User utterances x^{U} are mapped to a user dialogue state $v \in V^{\mathrm{U}}$, and system utterances x^{S} onto a system dialogue state $v \in V^{\mathrm{S}}$, whereby $V = V^{\mathrm{U}} \cup V^{\mathrm{S}}$ and $V^{\mathrm{U}} \cap V^{\mathrm{S}} = \emptyset$. Assigning utterances to the correct state depends on the conversational context such that two utterances with the same wording but in a different dialogue may refer to different dialogue states. The edges $e_{ij} \in E$ represent the probability $p_{i,j}$ of transitioning from state v_i to v_j when following the conversation. Given the alternating user and system utterances in a dialogue, it is assumed that state transitions happen from a user to a system state or vice versa: $\forall (v_i, v_j) \in V^{\mathrm{U}} \times V^{\mathrm{U}} : p_{v_i,v_j} = 0$ (similar for V^{S}).

3.2 Cluster-based dialogue structure induction

We consider the cluster-based method of [18, 19], frequently adopted as a baseline for DSI, which encodes utterances as vectors and then clusters them into the $|V^{\mathrm{U}}|$ user and $|V^{\mathrm{S}}|$ system states. The transition probabilities $p_{i,j}$ between states $v_i, v_j \in V$ are computed by counting the number of utterances in v_i for which the utterance that follows is in v_j and then normalize by dividing by the total number of utterances in v_i:

$$p_{i,j} = \frac{\#(v_i \rightarrow v_j)}{\# v_i}$$

Works that compare against this cluster-based method (i) use sub-optimal embeddings and (ii) do not use dialogue context. In particular, only the current utterance is encoded as a bag-of-words using GloVe [32, 52], word2vec [16, 23] or BERT [17, 37, 53]. Yet, such methods have been shown to produce sentence embeddings of low quality [21, 22]. Thus, we propose ELLoDAR, to efficiently learn locally dialogue-aware representations, by using (i) more powerful transformer-based sentence encoders such as SBERT [22] and TOD-BERT [28], and (ii) the local context window (i.e., preceding and next utterances) around the current utterance.

3.3 Efficiently learning locally dialogue-aware representations

ELLoDAR increases training efficiency by using the previous and next utterances as only context (yet considers both directions) based on the observation that utterances in task-oriented dialogues surrounded by similar context windows often have the same conversational goals. Additionally, ELLoDAR does not train an encoder from scratch as that would require significant computational efforts, and we envision a competitive but computationally efficient method. Rather, ELLoDAR exploits the rich semantics captured in the embeddings produced by pretrained transformer-based sentence encoders.

3.3.1 Model description

ELLoDAR combines two distinct strategies. In each strategy, a linear transformation is learned to transform an utterance x, as first encoded by a *frozen* pretrained sentence encoder $\phi(x)$, to a context-aware representation $f(\phi(x))$. We train different such transformations respectively for user and system representations (f^{U} resp. f^{S}). The first strategy is designed to learn representations that are similar for utterances that (can) appear in the same context of preceding and following utterances. In practice, we only consider *adjacent* utterances as the context window, and the linear maps are learned by *extrapolating* the considered utterance x's representation $\phi(x)$ onto those of the adjacent utterances.

More formally, the representation $f^*_{\mathrm{EXT},i} \in \mathbb{R}^{2h}$ for utterance x_i is obtained from the pretrained encoder representation $\phi(x_i) \in \mathbb{R}^h$ (with the superscript $* \in \{\mathrm{U}, \mathrm{S}\}$ indicating the system or user), as

$$f^*_{\mathrm{EXT},i} \triangleq f^*_{\mathrm{EXT}}(\phi(x_i)) = W^*_{\mathrm{EXT}}\, \phi(x_i) + b^*_{\mathrm{EXT}}$$

The parameters $W^*_{\mathrm{EXT}} \in \mathbb{R}^{2h \times h}$ and $b^*_{\mathrm{EXT}} \in \mathbb{R}^{2h}$ are trained by minimizing a vector similarity loss $\mathcal{L}^*_{\mathrm{EXT},i}$, i.e., ordinary least squares (OLS):

$$\mathcal{L}^*_{\mathrm{EXT},i} = \mathrm{OLS}\left(f^*_{\mathrm{EXT},i},\ \phi(x_{i-1}) \oplus \phi(x_{i+1})\right)$$

with $\oplus$ denoting concatenation. This is illustrated by the right-hand part of Fig. 2.

The second strategy *interpolates* the current user (system) embedding from the adjacent system (user) context embeddings, reflecting the assumption that context windows enclosing similar utterances should be represented close to

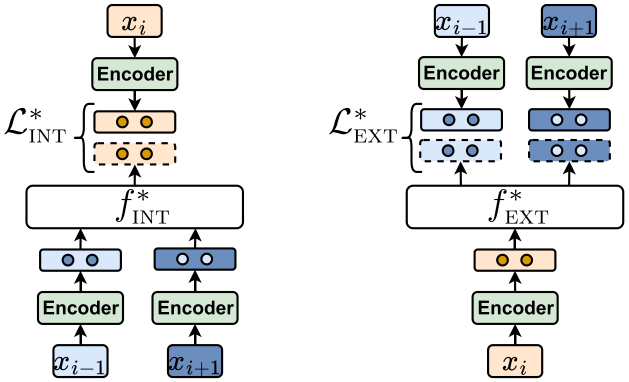


Fig. 2 *Training strategies* of ELLoDAR, where x_i are dialogue utterances, and Encoder is a pretrained sentence encoder. *Left*: f^*_{INT}, which predicts the embedding for the current utterance from local context embedding (i.e., the preceding and following utterances in a dialogue). *Right*: f^*_{EXT}, which predicts the context embedding from the current one

each other in the utterance representation space. The corresponding representation $f^*_{\text{INT},i} \in \mathbb{R}^h$ for utterance x_i is constructed from the pretrained encoder representations $\phi(x_{i-1})$ and $\phi(x_{i+1})$ of its adjacent[2] utterances as

$$f^*_{\text{INT},i} \triangleq f^*_{\text{INT}}\big(\phi(x_i)\big) = W^*_{\text{INT}}\,\big(\phi(x_{i-1}) \oplus \phi(x_{i+1})\big) + b^*_{\text{INT}}$$

with $W^*_{\text{INT}} \in \mathbb{R}^{h\times 2h}$ and $b^*_{\text{INT}} \in \mathbb{R}^h$. The corresponding loss is given by:

$$\mathcal{L}^*_{\text{INT},i} = \text{OLS}\left(f^*_{\text{INT},i}\ ,\ \phi(x_i)\right)$$

A visual summary is given on the left part of Fig. 2. During training, the introduced loss terms are calculated and minimized over all utterances across all dialogues. After training, CELLoDAR clusters the user utterances x^{U}_i represented as $f^{\text{U}}_{\text{EXT},i}$, $f^{\text{U}}_{\text{INT},i}$ or $f^{\text{U}}_{\text{EXT},i} \oplus f^{\text{U}}_{\text{INT},i}$ (similarly for the system utterances).

3.3.2 Background

ELLoDAR draws inspiration from the CBOW and skip-gram models [23] for learning word vectors, and especially $f^*_{\text{EXT},i}$ bears similarities to the skip-thought model [20] for learning *general purpose* sentence embeddings. However, skip-thought employs two separate decoders to generate the preceding and following sentences, which (i) necessitates substantial computational efforts, (ii) produces sub-optimal sentence embeddings [21, 22] and (iii) requires hyperparameter tuning. In contrast, to specifically obtain *dialogue aware* representations for clustering, ELLoDAR exploits pretrained sentence encoders (i) by efficiently learning linear transformations entirely in their encoding space (on CPU) with a vector-to-vector optimization objective, and thus (ii) *directly* optimizes the embeddings to capture the dialogue context necessary for DSI (rather than using an *indirect* decoding objective), and (iii) does not require hyperparameter tuning.

Other methods like DialoGPT [54], PLATO [55], and TOD-BERT [28], pretrain encoders on task-oriented dialogues to produce utterance representations that can be used in various downstream tasks, including clustering. It is worth noting that ELLoDAR differs in that it does not (pre)train an encoder from scratch, but rather works *complementary* and out-of-the-box with such already pretrained encoders: ELLoDAR adapts their representations to the task-specific dialogues by learning a linear transformation on top of them to specifically improve *cluster* performance. Therefore, ELLoDAR's efficient linear vector-to-vector regression is possible because pre-trained encoders, which already have undergone substantial computational efforts, enable this capability.

4 Experimental setup

We describe the datasets, and how they were adapted for DSI, in Section 4.1. In Section 4.2, we motivate our choices of three different types of pretrained sentence encoders that were used to train ELLoDAR, and discuss the recent joint models and cluster baselines to which ELLoDAR is compared in Section 4.3. We provide training details in Section 4.4, and extensively describe the evaluation methodology in Section 4.5.

4.1 Datasets

We follow prior works in unsupervised DSI [16, 17, 37] and conduct experiments on task-oriented dialogues that span 10 domains across four commonly used conversational datasets: DSTC2 [24], CamRest676 [25, 26], SimDial [27] and The Schema Guided dialogues (SGD) [5]. Our experiments comprise a broader range of datasets compared to prior works: the DVRNN model of [16] was benchmarked on SimDial and CamRest676, the SVRNN model of [17] solely on SimDial, and the model of [37] on SGD, CamRest676, and DSTC2. Our experiments cover all four datasets, thus making it the overall most comprehensive benchmark to date, to the best of our knowledge. SimDial contains synthetic dialogues that were generated using a pre-defined probabilistic grammar. The DTSC2 and SGD datasets consists of human-machine dialogues, whereas the human-human dialogues in CamRest676 were obtained with the Wizard-of-Oz methodology [56].

[2] Note that these adjacent utterances will be of the other type: *user* utterances are enclosed between two *system* utterances, and *system* utterances are enclosed between two *user* utterances.

In the aforementioned datasets, utterances are annotated with intents, acts and slots. We discard slot values and only consider their types since we map utterances, rather than slots, to dialogue states and because a single type can have potentially many values which would make the number of dialogue states intractable. Moreover, utterances may have multiple annotations, in which case we combine them. For example, the utterance "*I want to find a comedy movie. Search for movies now showing in Oakland*" with as intent *find-movies*, as act *inform*, and as slot types *genre* and *location*, becomes *[find-movies, inform.genre, inform.location]* (ignoring the respective values *comedy* and *Oakland*). Thus, we obtain exactly one label for each utterance, allowing us to compare the induced dialogue states against the gold utterance labels with external cluster metrics, as will be discussed in Section 4.5. The gold number of $|V^{\mathrm{U}}|$ user and $|V^{\mathrm{S}}|$ system dialogue states are respectively set to the number of unique user and system utterance labels. The statistics of the various domains and datasets are shown in Table 1 and samples of dialogues are given in the Tables 14–16. We release our modified datasets such that they can be adopted in future works.

Table 1 Dataset statistics

Dataset	N	$\|\mathcal{X}\|$	$\|V^{\mathrm{U}}\|$	$\|V^{\mathrm{S}}\|$	$\|V^{\mathrm{TURN}}\|$
SimDial					
- Weather	2,000	20,080	6	7	11
- Bus	2,000	24,158	7	8	13
- Restaurant	2,000	22,030	7	8	19
- Movies	2,000	26,156	8	9	21
SGD					
- Events	572	10,250	100	46	220
- Homes	268	4,128	71	24	113
- Music	331	4,566	28	10	43
- Movies	292	4,130	72	16	113
CamRest676	676	2,401	78	16	–
DSTC2	1,612	21,772	95	49	756

N denotes the total number of dialogues, $|\mathcal{X}|$ as the total number of utterances; and $|V^{\mathrm{U}}|$, $|V^{\mathrm{S}}|$ and $|V^{\mathrm{TURN}}|$ respectively indicating the number of unique user, system and *turn* (Section 4.5) states

4.2 Pretrained sentence encoders

As discussed in Section 3.3.2, rather than pretraining an encoder from scratch, ELLoDAR uses such already pretrained encoders out-of-the-box to produce utterance representations specifically for clustering. Since ELLoDAR is thus agnostic to the sentence encoder, it can in principle be used with any such encoder ϕ. For our experiments, we used three different types of models described below.

- **MiniLM-L6** [22]:[3] a *general* purpose sentence encoder that produces 384-dimensional vectors, offering a good trade-off between encoding speed and quality.
- **TOD-BERT-jnt** [28]: a BERT$_{\mathrm{BASE}}$ model, yielding 768-dimensional embeddings, and pretrained with a next-sentence prediction and contrastive objective, on 9 task-oriented datasets that include CamRest676 and all domains in SGD. It was pretrained to encode utterances within *dialogues*, so that these encodings could be used in a variety of task-oriented downstream tasks. Note that while we chose TOD-BERT, other choices of task-oriented encoders such as, e.g., DialoGPT [54] and PLATO [55] are also possible.
- **GloVe** [52]: utterances are represented as *bag-of-words*, i.e., their word-averaged GloVe embeddings. It is used as an ablation for the DVRNN and SVRNN models (see below) whose sentence encoders are initialized with GloVe, and as a baseline for the *sentence* encoders MiniLM and TOD-BERT.

4.3 Baselines

We aim to induce non-deterministic dialogue structures in task-oriented domains, as mentioned in Section 2.2. This same task is also considered by the joint DVRNN and SVRNN models, hence we use different configurations of these models as baselines for our CELLoDAR approach. In addition, we compare CELLoDAR to the cluster baselines of [18, 19] based on the used sentence encoders without ELLoDAR training. Specifically, the baselines we will benchmark our own approaches against are:

- **DVRNN** [16]: a discrete and recurrent extension of the Variational Auto-Encoder that learns to reconstruct the current turn from its discrete latent states and the preceding dialogue context. *Turns* are clustered into the discrete states.
- **SVRNN** [17]: shares the same architecture as DVRNN but extends it with a structured attention mechanism over its hidden states.
- **Cluster baselines**: utterances are clustered by using as input features their context window embeddings, represented as the concatenation of the embeddings of the utterances in the window. The utterance embeddings are obtained using the encoders of Section 4.2 and we consider as context windows (i) only the current utterance (indicated as C), as in prior works [16–19], (ii) the previous and current utterances, (PC), (iii) the full context window of previous, current, and next utterances (PCN).

[3] https://www.sbert.net/docs/pretrained_models.html

In contrast to existing works [16, 17] that compare only with cluster baseline (i), we additionally benchmark against (ii) and (iii), which serve as stronger baselines as they use additional dialogue context, similar to our CELLODAR approach.

4.4 Training details

As explained in Section 3.3, we use ordinary least squares to estimate the weights of the linear regression functions of ELLODAR (to obtain the representations $f^{\mathrm{U}}_{\mathrm{EXT},}, f^{\mathrm{U}}_{\mathrm{INT},}, f^{\mathrm{S}}_{\mathrm{EXT},}$ and $f^{\mathrm{S}}_{\mathrm{INT},}$) and thus do not require hyperparameter tuning. For both CELLODAR and the cluster baselines, we use k-means to separately cluster the user utterances x^{U} and the system utterances x^{S} into respectively the gold number of user and system dialogue states, $|V^{\mathrm{U}}|$ and $|V^{\mathrm{S}}|$. We use 10 random seeds to initialize k-means and report the average scores over these 10 runs for CamRest676 and DSTC2. For SimDial, the results presented in the main body are further averaged over the 4 domains *Weather*, *Bus*, *Restaurant*, and *Movies*. Similarly, for SGD, we further average over the 4 domains *Events*, *Homes*, *Music*, and *Movies*. Scores for the individual domains, as well as mean (± standard deviation) over the domains are given in Appendix A.

4.5 Evaluation

Shi et al. [16] opted for a qualitative evaluation in which humans rated induced conversational graphs. Qiu et al. [17] presented two automatic metrics to quantitatively assess the quality of such graphs. They introduced Structure Euclidean Distance and Structure Cross-Entropy, which both estimate a probabilistic mapping between the induced and the gold states. However, the authors later deemed them unstable[4] because of their high variance and recommended instead to employ external cluster metrics for evaluating induced conversational graphs based on *slot* clusters [42].

In Section 4.1, we described how to obtain labels for *utterance*-based DSI, enabling us to also adopt such metrics, and more in particular: (i) the adjusted rand index (ARI) [57], (ii) the adjusted mutual information (AMI) [58] and (iii) the Fowlkes-Mallows score (FM) [59]. ARI and AMI extend respectively the rand index [60] and the mutual information to adjust for chance: random clusters obtain a score of 0.0 whereas perfect ones obtain 1.0. The rand index measures, out of all pairs of samples, the percentage of correct ones. A pair is correct when either (i) both samples have the same gold label and they are assigned to the *same* cluster, or (ii) both samples have a different gold label and they are mapped to a *different* cluster Mutual information, on the other hand, relates to purity and assigns a high score to clusters if the majority of their samples have the same label.

DVRNN and SVRNN cluster *turns* $(x_i^{\mathrm{U}}, x_{i+1}^{\mathrm{S}})$ of consecutive (user, system) utterances into *turn* states $v^{\mathrm{TURN}} \in V^{\mathrm{TURN}} \subseteq V^{\mathrm{U}} \times V^{\mathrm{S}}$ (or vice versa). The number of turn states $|V^{\mathrm{TURN}}|$ corresponds to the unique number of turn labels, i.e., the combination of labels of the turn's utterances. Turn clustering becomes challenging when states contain few utterances because the turn states will become even sparser, e.g., in Table 1 with for DSTC2: $|V^{\mathrm{TURN}}| = 756 \gg |V^{\mathrm{U}}| + |V^{\mathrm{S}}|$. To allow for a fair comparison with DVRNN and SVRNN, we report turn state cluster results on SGD and SimDial, for CELLODAR and the cluster baselines. These are automatically inferred by combining the separately induced cluster identifiers of the system and user utterances that comprise a turn.[5] In addition, we report *utterance*-based results for CELLODAR and the cluster baselines on CamRest676 and DSTC2. Note that CamRest676 lacks annotations for many system utterances, and the gold *turn-based* states for DSTC2 become very sparse ($|V^{\mathrm{TURN}}| = 756$). Therefore, obtaining turn-based results for CamRest676 and DSTC2 appeared not feasible, preventing the comparison of our models with DVRRN and SVRNN on these datasets (see Table 3).

5 Results

As the joint models DVRRN and SVRNN are initialized with GloVe embeddings, we first report results for the cluster baselines and CELLODAR also based on GloVe, and thereby eliminating the advantage that could be attributed to the use of pretrained transformers in our most competitive models. Table 2 shows that the cluster baselines outperform the joint models in almost all cases. Only for SimDial does the bag-of-words model of [18, 19] (GloVe$_{\mathrm{C}}$) perform worse in terms of ARI and FM. Most notably, the strongest baseline (PCN) surpasses SVRNN on SimDial (SGD) by +49.4 (+10) percentage points in ARI, +34.9 (+27.3) in AMI, and +60.8 (+7.6) in FM. Moreover, the best CELLODAR model consistently outperforms the best cluster baseline, with further improvements on SimDial (SGD) of +23.3 (+3.8) in ARI, +13.1 (+1.2) in AMI, and +19.3 (+4.1) in FM. The key observations from this GloVe-based comparison are: (i) all cluster baselines, except the bag-of-words model of [18, 19] (GloVe$_{\mathrm{C}}$), outperform the joint models, and (ii) the best

[4] https://github.com/Liang-Qiu/SVRNN-dialogues

[5] As a reference, the corresponding *utterance*-based evaluation of the cluster baselines and CELLODAR on SimDial and SGD are given in Appendix A.

Table 2 Main results with GloVe

Model	SimDial ARI	SimDial AMI	SimDial FM	SGD ARI	SGD AMI	SGD FM
Joint models						
DVRNN [16]	40.5	61.3	50.9	14.3	25.3	22.7
SVRNN [17]	25.7	50.9	37.9	16.3	28.4	24.2
Cluster baselines						
- GloVe$_{\text{C}}$ [18, 19]	48.0	71.5	57.7	14.2	42.4	19.8
- GloVe$_{\text{PC}}$	65.8	81.6	72.2	22.2	52.7	28.0
- GloVe$_{\text{PCN}}$	75.1	85.8	79.4	26.3	55.7	31.8
CELLoDAR						
- GloVe$_{\text{INT}}$	98.1	98.5	98.4	25.4	50.9	31.4
- GloVe$_{\text{EXT}}$	90.9	94.1	92.6	26.4	52.5	33.0
- GloVe$_{\text{INT+EXT}}$	**98.4**	**98.9**	**98.7**	**30.1**	**56.9**	**35.9**

Turn-based evaluation. SimDial (SGD) results are averaged over: Weather, Bus, Restaurant and Movies (Events, Homes, Music and Movies). The **best** model is typeset in bold and the runner-up is underlined

CELLoDAR-model consistently outperforms the best cluster baseline.

Table 3 shows that these observations also hold for the sentence encoders MiniLM and TOD-BERT, with CELLoDAR and the cluster baselines outperforming their counterparts based on the bag-of-words GloVe encoder. Note that the models of [18, 19] (subscripted by C) with MiniLM and TOD-BERT consistently outperform the joint models, which was not always the case for GloVe.

Table 4 reports the training times of CELLoDAR, and the joint DVRNN and SVRNN models. First, we discuss the computational resources required to train each model. DVRNN and SVRNN are built on the same code base.[6] We adopt the hyperparameters from [17] with: (i) dropout set to 0.5, (ii) Adam as optimizer, (iii) a learning rate of 0.001, and (iv) 60 epochs. DVRNN is trained on a single GTX 1080 Ti GPU, using 40 dialogues per batch for all SimDial and all SGD domains. SVRNN uses a single Tesla V100 with batch size 40 for all SimDial domains and size 10 for all SGD domains (we could not fit more in memory). In contrast, CELLoDAR uses a single 2.6 GHz Intel Core i7 to first learn its representations with ordinary least squares, and then cluster them with k-means [61], with 1,000 as the maximum iterations, and its k centroids initialized by k++means [62]. On a Tesla V100 GPU, MiniLM and TOD-BERT have encodings speeds[7] of respectively 14,200 and 2,800 utterances/second. Encoding the largest considered dataset then takes respectively 1.84 and 9.34 seconds for MiniLM and TOD-BERT. When adding 9.34 seconds for the worst case encoding speed to the average of 15.2 seconds to both learn and cluster representations, our slowest model, TOD-BERT$_{\text{INT+EXT}}$, achieves a speedup of 89× compared to DVRNN and 4,909× compared to SVRNN, as shown in Table 4. Encoding sentences with MiniLM rather than with TOD-BERT, results in further speedups, making it 279× and 15,894× faster than DVRNN and SVRNN, respectively.

6 Discussion

In Section 6.1, we compare the recent joint models to the cluster-based methods, i.e., cluster baselines and CELLoDAR. Next, we compare the performance between ELLoDAR's two encoding strategies INT and EXT in Section 6.2. The effect of including local context by vector concatenation on the cluster baselines' performance is analyzed in Section 6.3. We discuss the impact of using a *bag-of-words*, *general* purpose, or *task-oriented* sentence encoder on the cluster performance in Section 6.4. Then, in Section 6.5, we vary the gold number of dialogue states used as input for the clustering algorithm, to analyze CELLoDAR's effectiveness if that gold number of states is unknown. We compare in Section 6.6 the training time performance of the joint models to that of CELLoDAR, and conclude in Section 6.9 by discussing the limitations of this work.

6.1 Joint methods versus cluster-based approaches

Tables 2 and 3 show that the joint methods are outperformed consistently by the cluster baselines and CELLoDAR. Also evidenced by the low AMI, ARI, and FM scores, we observed that the joint models frequently clustered utterances with different ground-truth labels into the same state. As the joint models are based on variational auto-encoders, optimized with a next turn decoding objective, we hypothesize that their poor performance is caused by the *posterior collapse* [38–40]. The latter occurs when the model solely relies on the decoder's auto-regressive properties rather than on the latent states to decode the next turn. That is, even if the joint models ignore the latent states entirely, they may still attain a small decoding loss. Hence explaining why utterances with different ground-truth states are often incorrectly assigned to the same state. The cluster baselines and CELLoDAR, on the other hand, do not rely on such decoding objectives, but instead induce dialogue states with k-means and thus directly exploit similarities between vector representations of utterances.

[6] https://github.com/Liang-Qiu/SVRNN-dialogues

[7] https://www.sbert.net/docs/pretrained_models.html. Since we did not benchmark TOD-BERT, we report the speed of another BERT$_{\text{BASE}}$ encoder: msmarco-bert-base.

Table 3 Main results with sentence encoders

	SimDial			SGD			CamRest676			DSTC2		
Model	ARI	AMI	FM	ARI	AMI	FM	ARI	AMI	FM	ARI	AMI	FM
Joint models												
DVRNN [16]	40.5	61.3	50.9	14.3	25.3	22.7	–	–	–	–	–	–
SVRNN [17]	25.7	50.9	37.9	16.3	28.4	24.2	–	–	–	–	–	–
Cluster baselines												
- MiniLM-L6$_{\text{C}}$ [18, 19]	64.0	80.1	69.2	23.3	53.6	30.2	15.7	39.1	26.0	56.9	66.0	62.7
- MiniLM-L6$_{\text{PC}}$	75.0	87.0	78.2	31.5	63.6	37.3	14.6	34.9	24.9	37.5	57.1	44.8
- MiniLM-L6$_{\text{PCN}}$	85.3	90.0	87.2	34.3	64.2	40.2	14.2	36.7	24.4	24.5	51.0	32.2
CELLoDAR												
- MiniLM-L6$_{\text{INT}}$	**99.9**	**99.8**	**99.9**	37.7	64.7	42.9	16.3	40.9	26.7	26.2	42.2	32.3
- MiniLM-L6$_{\text{EXT}}$	99.5	99.6	99.5	39.0	63.3	45.3	**19.9**	38.1	30.2	**65.2**	**73.9**	**69.3**
- MiniLM-L6$_{\text{INT+EXT}}$	99.6	99.7	99.6	**40.4**	**66.2**	**46.0**	**19.9**	**43.0**	**30.5**	48.0	67.1	54.7
Cluster baselines												
- TOD-BERT$_{\text{C}}$ [18, 19]	73.4	84.9	78.7	29.4	56.9	36.6	13.6	30.7	24.0	56.9	66.5	62.6
- TOD-BERT$_{\text{PC}}$	79.0	88.2	83.1	32.9	62.7	38.8	9.8	26.6	19.6	36.2	56.5	43.3
- TOD-BERT$_{\text{PCN}}$	88.2	93.2	90.4	40.1	65.8	45.0	11.6	30.5	21.6	21.8	49.1	29.5
CELLoDAR												
- TOD-BERT$_{\text{INT}}$	**99.8**	**99.7**	**99.8**	39.7	63.2	44.7	13.3	30.5	23.3	26.7	45.0	33.0
- TOD-BERT$_{\text{EXT}}$	97.8	98.6	98.2	48.1	66.4	53.1	13.2	31.0	23.7	**62.9**	**70.9**	**67.6**
- TOD-BERT$_{\text{INT+EXT}}$	97.9	98.7	98.3	**49.8**	**68.6**	**54.2**	**14.6**	**34.4**	**25.2**	48.6	65.1	55.3

Turn-based evaluation for SimDial and SGD, *utterance*-based evaluation for CamRest676 and DSTC2. SimDial (SGD) results are averaged over: Weather, Bus, Restaurant and Movies (Events, Homes, Music and Movies). The **best** model is typeset in bold and the runner-up is underlined

Moreover, the results in Section 5 demonstrate that the best CELLoDAR models consistently outperform the best cluster baselines. Unlike ELLoDAR, the cluster baselines do not *learn* to incorporate local dialogue context into utterance representations, instead they simply concatenate representations. This indicates that *learning* how to include local context into representations is beneficial for DSI, and that ELLoDAR's learning schemes are successful at doing so. We consider ELLoDAR the main technical contribution of this work.

Table 4 Training times

				Epoch Time (s)			Cluster Time (s)			Total (s)
Model	CPU	$\lvert\theta\rvert$	N_{E}	Avg.	Min.	Max.	Avg.	Min.	Max.	Avg.
Joint models										
DVRNN [16]	✗	6.60M	60	35.3	18.2	77.6	–	–	–	2,118
SVRNN [17]	✗	7.23M	60	2,008	555	5,158	–	–	–	120,480
CELLoDAR										
- MiniLM-L6$_{\text{INT}}$	✓	590K	1	1.28	0.59	2.33	1.50	0.75	3.79	2.78
- MiniLM-L6$_{\text{EXT}}$	✓	591K	1	0.59	0.26	1.10	4.05	2.00	11.24	4.65
- MiniLM-L6$_{\text{INT+EXT}}$	✓	1.18M	1	1.80	0.85	3.11	3.94	1.63	9.50	5.74
- TOD-BERT$_{\text{INT}}$	✓	2.36M	1	5.16	3.56	6.34	2.79	1.39	8.29	7.95
- TOD-BERT$_{\text{EXT}}$	✓	2.36M	1	2.03	1.00	3.23	8.56	3.84	25.32	10.59
- TOD-BERT$_{\text{INT+EXT}}$	✓	4.72M	1	7.19	4.57	9.64	8.01	3.06	21.57	15.20

The number of learnable parameters, $\lvert\theta\rvert$, is averaged over all SGD and SimDial domains, and N_{E} as the number of epochs. The total runtime is the sum of the average epoch and average cluster times. All runtimes are in seconds. For each model the average, minimum and maximum epoch (cluster) times are calculated over all SGD and SimDial domains

6.2 Comparing ellodar's encoding schemes

We note that INT and EXT encodings of the utterance take different views: while EXT aims to reconstruct a representation of the context from an utterance $\phi(x)$ itself, INT rather aims to reconstruct the utterance representation $\phi(x)$. Given this complementary mechanism, we a priori expect their combination (INT+EXT) to perform best, while superiority of one over the other cannot be intuitively anticipated. The results in Tables 2 and 3 reveal that for SimDial all 3 encodings perform nearly perfectly,[8] which prevents us from distinguishing their performance. Still, for both SGD and CamRest676, INT+EXT performs notably better than INT, and slightly better than EXT, thus confirming our a priori expectation. Somewhat surprisingly, on DSTC2 INT+EXT clearly performs worse than EXT. We can however attribute this to the fact that DSTC2 comprises human-to-chatbot dialogues where the bot frequently misinterprets the user, thus leading to contexts that are sometimes disconnected from an enclosed utterance: as a result (erroneous) context information from INT is not as useful, as also reflected in low INT scores.

6.3 Impact of local context on the performance of the cluster baselines

We investigate the effect on structure quality of using the two straightforward vector concatenation approaches for incorporating preceding (PC), and both preceding and subsequent (PCN) context. This contrasts with the model of [18, 19] that uses no context and which was later adopted as a baseline in [16, 17]. Intuitively, we expect the cluster baselines that leverage full context (PCN) to perform better than those using only the preceding (PC) or no context at all (C). Tables 2 and 3 reveal that on SimDial and SGD, the cluster metrics indeed consistently improve as the context window expands: C<PC<PCN. Conversely, the results for CamRest676 and DSTC2 get worse as the context window grows larger. The CamRest676 results indicate that naively including context does not always improve structure quality, emphasizing the benefits of using more advances strategies like EXT and INT+EXT. On DSTC2, the difference is even more apparent, with the cluster baselines of C clearly outperforming those of PCN, which is consistent with the previously discussed results of INT and EXT and thus attributed to the erroneous context from the human-to-chatbot dialogues. Still, we recommend adopting PC and PCN as baselines, since they significantly improve the average structure quality on all 4 SimDial and SGD domains.

6.4 Impact of the sentence encoder on the structure quality

First, as Tables 2 and 3 show, both the cluster baselines and the CELLODAR models based on bag-of-words representations (GloVe) perform consistently worse than their counterparts based on powerful sentence encoders (MiniLM and TOD-BERT), supporting our claim that transformer-based encoders are better for DSI.

Second, we investigate whether TOD-BERT, specifically trained to encode utterances in dialogues, outperforms the general purpose encoder MiniLM. The results in Table 3 are mixed. Since TOD-BERT was trained on all 16 SGD domains, including the 4 that we consider, we indeed find that on SGD, TOD-BERT models consistently outperform those based on MiniLM, notably for INT+EXT (improvements of +9.4, +2.4, and +12.4 for the ARI, AMI, and FM metrics respectively). Since TOD-BERT is also trained on all CamRest676 dialogues, it is surprising that MiniLM outperforms it. We hypothesize that this is due to the fact that the CamRest dialogues (i) comprise only 0.67% of the total dialogues used to train TOD-BERT (rather than the SGD dialogues accounting for 22.66%), and (ii) are dissimilar to those of SGD, such that little transfer occurs. Furthermore, the results on SimDial and DSTC2 (which were not used to train TOD-BERT) vary, with TOD-BERT outperforming MiniLM for some models but not for others, making it difficult to draw conclusions about the transferability to unseen domains.

In summary, the preliminary evidence on SGD suggests that it may be beneficial to pretrain sentence encoders *specifically* on the dialogues from which the structure is induced. The advantages of transferring to dialogues from unseen datasets (SimDial, DSTC2), however, remain unclear.

6.5 Overestimating the number of dialogue states

We assumed the gold number of the $|V^{\mathrm{U}}|$ user and $|V^{\mathrm{S}}|$ system states to be known and used them to initialize k-means. In practice, $|V^{\mathrm{U}}|$ and $|V^{\mathrm{S}}|$ can be estimated by inspecting a subset of dialogues, but determining them exactly, however, is challenging. To this end, we investigate the effect of *overestimating* the number of states by initializing k-means with twice the gold number of user and system states: $k = 2 \cdot |V^{\mathrm{U}}|$ and $k = 2 \cdot |V^{\mathrm{S}}|$.

We present MiniLM results for the best cluster baseline (PCN) and the best CELLODAR model (INT+EXT), both with the *overestimated* number of clusters, and compare them to their

[8] CELLODAR seems to almost perfectly reconstruct the underlying synthetic structure of SimDial, rendering the dataset less useful for future research on DSI.

counterparts, as well as the DVRNN and SVRNN models with the *gold* number of states.

First, Table 5 shows that the *overestimated* cluster baseline and INT+EXT still outperform DVRNN and SVRNN in all metrics and on all datasets, with notable improvements for MiniLM$_{\text{PCN}}$ (MiniLM$_{\text{INT+EXT}}$) in AMI: +14.9 (+16.3) on SimDial, and +27.4 (+29.4) on SGD.

Second, when comparing the *overestimated* models to their counterparts initialized with the *gold* number, we find that the *overestimated* models (i) drop in ARI and FM, and (ii) drop in AMI but MiniLM$_{\text{PCN}}$ (MiniLM$_{\text{INT+EXT}}$) still attain relatively high values of 76.2 (77.6) on SimDial and 52.7 (54.7) on SGD. Since the number of clusters increased twofold, utterances of the same gold state can be partitioned further into different clusters. Therefore, the decrease in ARI and FM is expected since these metrics penalize utterances of the same gold state if they are mapped to different clusters. AMI, on the other hand, measures cluster purity, with a high score indicating that most utterances in a cluster belong to the same gold state.

Thus, even when the number of clusters is *overestimated* by a factor of two, the cluster baseline and CELLoDAR induce relatively pure clusters, with the latter outperforming the former, and both still considerably better than the DVRNN and SVRNN with the *gold* number of states.

6.6 Training time performance

In Section 5, we reported that our slowest CELLoDAR model achieved a speedup of 89× over DVRNN and 4,909× over SVRNN. This efficiency gap can be attributed to the fact that joint models are optimized with stochastic gradient descent (SGD), whereas CELLoDAR is trained with more efficient learning schemes. Training neural networks with SGD requires multiple epochs of forward and backward passes through all training samples before converging to a local minimum, and thus, as per [16, 17], we used 60 epochs to train the joint models. Although CELLoDAR relies on neural networks (MiniLM and TOD-BERT) to obtain sentence representations, encoding all training samples requires just a single forward pass. Similarly, ELLoDAR's linear transformations are cast as vector-to-vector regression and thus can be learned with ordinary least squares in a single pass. As [61]'s algorithm for k-means has efficient implementations [63], clustering the ELLoDAR representations is fast.

The training time difference between CELLoDAR based on TOD-BERT and on MiniLM is twofold. With 3M parameters compared to 110M, MiniLM encodes sentences much faster than TOD-BERT. Additionally, MiniLM produces 384-dimensional vectors, while TOD-BERT produces vectors with twice the number of dimensions (i.e., 768). As k-means runtime depends on the number of input features, clustering MiniLM's representations is thus faster than clustering TOD-BERT's.

Table 5 Overestimating the number of dialogue states

	SimDial			SGD		
Model	ARI	AMI	FM	ARI	AMI	FM
Gold number						
- DVRNN [16]	40.5	61.3	50.9	14.3	25.3	22.7
- SVRNN [17]	25.7	50.9	37.9	16.3	28.4	24.2
- MiniLM-L6$_{\text{PCN}}$	85.3	90.0	87.2	34.3	64.2	40.2
- MiniLM-L6$_{\text{INT+EXT}}$	99.6	99.7	99.6	40.4	66.2	46.0
Overestimated						
- MiniLM-L6$_{\text{PCN}}$	46.2	76.2	58.3	18.9	52.7	28.6
- MiniLM-L6$_{\text{INT+EXT}}$	61.8	77.6	70.3	22.7	54.7	32.7

Turn-based evaluation for SimDial and SGD. SimDial (SGD) results are averaged over: Weather, Bus, Restaurant and Movies (Events, Homes, Music and Movies)

6.7 Ablation study

We provide ablations to assess the impact of ELLoDAR's different components. First, we examine if training ELLoDAR with bidirectional context, i.e., both preceding and subsequent dialogue, improves structure quality compared to training ELLoDAR with only preceding *or* subsequent context. Second, since ELLoDAR uses a local context window (the preceding and subsequent utterance) for efficient representation learning, we explore whether training on larger context windows is useful.

Impact of bidirectional context on structure quality To assess the impact of bidirectional context on cluster performance, we compare ELLoDAR's strategies: INT, EXT, and INT+EXT, trained with only the preceding (P) or next (N) utterance as context, against ELLoDAR's standard bidirectional (PN) context. For INT, we transform the preceding (respectively next) utterance representation $\phi(x_{i-1})$ (respectively, $\phi(x_{i+1})$) into the representation of the current utterance $\phi(x_i)$. The training scheme and loss for 'interpolating' from the preceding utterance are:

$$f^*_{\text{INT,P},i} \triangleq f^*_{\text{INT,P}}\big(\phi(x_i)\big) = W^*_{\text{INT,P}}\,\phi(x_{i-1}) + b^*_{\text{INT,P}},$$

and loss $\mathcal{L}^*_{\text{INT,P},i} = \text{OLS}\left(f^*_{\text{INT,P},i}, \phi(x_i)\right)$.
Similarly, for EXT with the preceding (next) utterance as context, we extrapolate from x_i to x_{i-1} (x_{i+1}) using:

$$f^*_{\text{EXT,P},i} \triangleq f^*_{\text{EXT,P}}\big(\phi(x_i)\big) = W^*_{\text{EXT,P}}\,\phi(x_i) + b^*_{\text{EXT,P}},$$

and loss $\mathcal{L}^*_{\mathrm{EXT,P},i} = \mathrm{OLS}\left(f^*_{\mathrm{EXT,P},i}, \phi(x_{i-1})\right)$.

Note: the representations for INT+EXT with the preceding utterance as only context is given by the concatenation $f^*_{\mathrm{EXT,P}}\left(\phi(x_i)\right) \oplus f^*_{\mathrm{INT,P}}\left(\phi(x_i)\right)$ (and similarly for the next utterance as context).

Except for EXT and DSTC2, the results presented in Table 6 clearly underscore the importance of using bidirectional context for learning representations to induce dialogue structures: across all datasets and strategies, using both preceding and subsequent dialogue context (PN) consistently yields higher structure quality compared to using either preceding (P) or subsequent (N) context alone.

Impact of dialogue context width on structure quality In the previous paragraph, we highlighted the importance of training ELLoDAR with the bidirectional rather than either solely with the preceding or subsequent dialogue context. However, it is worth noting that ELLoDAR uses only the local dialogue context, comprising the preceding and subsequent utterances to efficiently learn representations. Here, we investigate whether using larger (*bidirectional*) dialogue contexts can yield improved ELLoDAR representations. To explore this, we compare the performance of ELLoDAR's strategies INT, EXT, and INT+EXT trained on larger dialogue contexts against training ELLoDAR with the default local context window. We experiment with two context windows increasingly larger than ELLoDAR's default local dialogue context window of just 1 preceding and next utterance (PN):

(1) The dialogue context consisting of the concatenation of representations of the 2 preceding and 2 subsequent utterances (shortly written as P_2N_2), for which we provide the training and loss below for the INT strategy.

$$\begin{aligned} f^*_{\mathrm{INT,P_2N_2},i} &\triangleq f^*_{\mathrm{INT,P_2N_2}}\left(\phi\left(x_i\right)\right) \\ &= W^*_{\mathrm{INT,P_2N_2}}\left(\phi(x_{i-2})\oplus\phi(x_{i-1})\oplus\phi(x_{i+1}) \right. \\ &\quad \left. \oplus\phi(x_{i+2})\right) + b^*_{\mathrm{INT,P_2N_2}}, \end{aligned}$$

with as loss $\mathcal{L}^*_{\mathrm{INT,P_2N_2},i} = \mathrm{OLS}\left(f^*_{\mathrm{INT,P_2N_2},i}, \phi(x_i)\right)$.

(2) The dialogue context consisting of the concatenation of the average of all preceding and the average of all subsequent utterance representations (P_*N_*). Note that here we take the mean instead of concatenating all preceding utterances and the mean of all subsequent utterance representations to avoid high-dimensional representations that may prevent efficient clustering. We provide the training and loss below for the INT strategy.

$$f^*_{\mathrm{INT,P_*N_*},i} \triangleq f^*_{\mathrm{INT,P_*N_*}}\left(\phi(x_i)\right) =$$

$$W^*_{\mathrm{INT,P_*N_*}}\left(\underset{j=0,\ldots,i-1}{\mathrm{average}}\ \phi(x_j)\oplus \underset{k=i+1,\ldots,N-1}{\mathrm{average}}\ \phi(x_k)\right)+b^*_{\mathrm{INT,P_*N_*}},$$

with as loss $\mathcal{L}^*_{\mathrm{INT,P_*N_*},i} = \mathrm{OLS}\left(f^*_{\mathrm{INT,P_*N_*},i}, \phi(x_i)\right)$.

Table 7 reveals that, for SimDial, there is minimal difference in cluster performance among various dialogue context sizes. However, across all other datasets (excluding DSTC2 and the INT(+EXT) strategy), the results indicate that the overall best structure quality is achieved when ELLoDAR is trained with the local context window of just a single preceding and next utterance (PN). It consistently outperforms

Table 6 Impact of bidirectional context on structure quality

	SimDial			SGD			CamRest676			DSTC2		
Model	ARI	AMI	FM	ARI	AMI	FM	ARI	AMI	FM	ARI	AMI	FM
MiniLM-L6$_{\mathrm{INT}}$												
- P	93.2	89.1	94.1	20.5	48.6	25.3	13.3	31.3	24.8	**28.4**	31.7	**32.9**
- N	88.9	88.9	90.4	25.2	48.0	42.9	13.7	34.0	24.3	25.1	32.5	30.3
- PN **(ours)**	**99.9**	**99.8**	**99.9**	**37.7**	**64.7**	**42.9**	**16.3**	**40.9**	**26.7**	26.2	**42.2**	32.3
MiniLM-L6$_{\mathrm{EXT}}$												
- P	94.7	95.9	95.4	25.8	52.9	33.8	11.6	24.0	21.2	61.0	68.1	65.3
- N	94.4	92.2	95.1	27.0	50.8	33.0	16.8	35.0	27.0	60.5	65.8	64.7
- PN **(ours)**	**99.5**	**99.6**	**99.5**	**39.0**	**63.3**	**45.3**	**19.9**	**38.1**	**30.2**	**65.2**	**73.9**	**69.3**
MiniLM-L6$_{\mathrm{INT+EXT}}$												
- P	95.2	92.7	95.8	24.0	53.4	30.5	13.8	30.2	23.4	37.7	53.7	43.6
- N	93.7	91.8	94.6	29.4	53.6	34.4	15.5	37.2	25.8	36.7	54.1	42.9
- PN **(ours)**	**99.6**	**99.7**	**99.6**	**40.4**	**66.2**	**46.0**	**19.9**	**43.0**	**30.5**	**48.0**	**67.1**	**54.7**

Turn-based evaluation for SimDial and SGD, *utterance*-based evaluation for CamRest676 and DSTC2. SimDial (respectively SGD) results are averaged over: Weather, Bus, Restaurant and Movies (respectively Events, Homes, Music and Movies)

Table 7 Impact of dialogue context width on structure quality

Model	SimDial			SGD			CamRest676			DSTC2		
	ARI	AMI	FM	ARI	AMI	FM	ARI	AMI	FM	ARI	AMI	FM
MiniLM-L6$_{\text{INT}}$												
- P$_*$N$_*$	**100**	**99.9**	**100**	24.7	51.8	30.6	15.4	40.0	25.7	20.3	39.1	27.0
- P$_2$N$_2$	98.7	98.6	98.9	29.2	59.4	36.1	16.3	40.7	26.6	**32.4**	44.3	**38.9**
- PN **(ours)**	99.9	99.8	99.9	**37.7**	**64.7**	**42.9**	**16.3**	**40.9**	**26.7**	26.2	**42.2**	32.3
MiniLM-L6$_{\text{EXT}}$												
- P$_*$N$_*$	94.2	92.2	95.0	18.5	42.3	25.0	11.8	25.5	21.8	58.8	63.2	63.3
- P$_2$N$_2$	97.2	97.3	97.6	**39.2**	62.3	**45.6**	19.1	35.7	29.3	60.9	69.2	65.6
- PN **(ours)**	**99.5**	**99.6**	**99.5**	39.0	**63.3**	45.3	**19.9**	**38.1**	**30.2**	**65.2**	**73.9**	**69.3**
MiniLM-L6$_{\text{INT+EXT}}$												
- P$_*$N$_*$	**99.9**	**99.8**	**99.6**	26.6	55.7	33.1	16.8	40.4	27.2	27.0	48.7	34.2
- P$_2$N$_2$	98.5	98.3	98.7	35.6	63.9	42.0	18.3	40.5	28.9	**52.5**	**66.2**	**58.7**
- PN **(ours)**	99.6	99.7	**99.6**	**40.4**	**66.2**	**46.0**	**19.9**	**43.0**	**30.5**	48.0	67.1	54.7

MiniLM-L6$_{\text{PCN}}$. *Turn*-based evaluation for SimDial and SGD, *utterance*-based evaluation for CamRest676 and DSTC2. SimDial (SGD) results are averaged over: Weather, Bus, Restaurant and Movies (Events, Homes, Music and Movies)

ELLoDAR trained with the full context window P$_*$N$_*$, and is either better or on par with ELLoDAR trained on P$_*$N$_*$ as context. This observation is further supported by the fact that the larger the context window, the poorer the cluster performance: the cluster performance for P$_*$N$_*$ is inferior to that of P$_2$N$_2$, with the latter slightly underperforming compared to the local dialogue context window PN. These results confirm that using only the local dialogue context for learning ELLoDAR's representations is a good choice. However, it is worth noting that the capacity of ELLoDAR's linear vector-to-vector regression is limited. As a result, ELLoDAR may be too constrained to effectively exploit the subtle signals in larger dialogue context. Nevertheless, the observation that ELLoDAR can effectively exploit signals in the local dialogue context alone suggests that there is sufficient signal in this local context to induce dialogue structures. This is particularly noteworthy when compared to more complex variational-based models such as DVRNN and SVRNN, which rely on the entire preceding dialogue context, yet struggle to induce representative dialogue structures.

6.8 Qualitative analysis

While ELLoDAR's representations can be efficiently learned, its efficiency primarily stems from its linear vector-to-vector regression objective. Yet, linear transformations may be too restrictive to handle complex edge cases, as there is a trade-off between efficiency and the complexity of cases ELLoDAR can model. Hence, to better understand these limitations, we conduct a qualitative analysis of common failure modes of ELLoDAR. We begin by identifying three failure modes, i.e., instances where ELLoDAR's utterance representations are incorrectly assigned to clusters, and provide examples of each. Next, we present the distribution of these three failure modes by manually categorizing a randomly selected subset of utterances that were erroneously assigned to clusters induced by CELLoDAR into these failure modes.

Identification of common failure modes To better understand ELLoDAR's shortcomings, we reveal and analyze common failure modes where utterances are incorrectly assigned to clusters due to ELLoDAR's learning approach, particularly due to its reliance on local context.

For this, we conduct a qualitative analysis of CELLoDAR-induced clusters based on human annotation. By manually categorizing incorrectly assigned utterances, along with their respective previous and subsequent utterances, we can reveal the most visible failure modes inherent to ELLoDAR's learning scheme. First, to identify incorrectly assigned utterances, we use the following heuristic: for each CELLoDAR-induced cluster $\mathcal{C}$, we assign a gold label to $\mathcal{C}$ which is the most prevalent gold label $y_{\mathcal{C},\text{GOLD}}$ among all utterances in $\mathcal{C}$. An utterance x_i is then erroneously assigned to $\mathcal{C}$ if its gold label y_i differs from the most frequently occurring gold label in $\mathcal{C}$, i.e., $y_i \neq y_{\mathcal{C},\text{GOLD}}$. Second, to categorize misassigned utterances into failure modes intrinsic to ELLoDAR's learning scheme, we manually compare each misassigned utterance and its local context window with those of correctly assigned utterances within the same cluster. We consider the following failure modes:

(1) P✗C✓N✗: the misassigned utterance x_i shares semantics with correctly assigned utterances $x_{j\neq i}$ in cluster $\mathcal{C}$. However, the preceding utterance x_{i-1} and subsequent utterance x_{i+1} differ from the dialogue context of correctly assigned utterances, i.e., $x_{i+1} \not\approx x_{j+1}$ and $x_{i-1} \not\approx x_{j-1}$. The example in the upper row of Table 8, illustrates this, where the misassigned utterance *"Yes, that sounds great"* is equivalent to the correctly assigned *"Sounds great"*. Yet, their dialogue states (y_j*: affirm* for correctly assigned x_j; y_i*: select* for misassigned x_i) differ due to variations in the semantics of both preceding and subsequent utterances. This mode is intrinsic to ELLoDAR, where: (i) for $f^*_{\text{INT},i}$, two distinct context windows transform into the same utterance representation, i.e., $\phi($*"Sounds great"*$) \approx \phi($*"Yes, that sounds great"*$)$, and (ii) for $f^*_{\text{EXT},i}$, the same input utterance representation $\phi($*"Sounds great"*$) \approx \phi($*"Yes, that sounds great"*$)$ may transform based into the context representation most frequently associated with this input (e.g., that of the correctly assigned utterances).

(2) P✓C✗N✓: the misassigned utterance x_i lacks shared semantics with correctly assigned utterances $x_{j\neq i}$ in $\mathcal{C}$. However, both the preceding and subsequent utterances share semantics among misassigned and correctly assigned utterances, i.e., $x_{i+1} \approx x_{j+1}$ and $x_{i-1} \approx x_{j-1}$. In the middle part of Table 8, x_i is more specific than x_j as it not only informs about the number of beds but also mentions allowing pets. Note that the reverse, where x_j is more specific than x_i, can also occur. This mode is intrinsic to ELLoDAR, where: (i) for $f^*_{\text{INT},i}$, two equivalent context representations may transform into the utterance representation most frequently surrounded by that context, i.e., $\phi(x_j)$, and (ii) for $f^*_{\text{EXT},i}$, semantically different input utterances transform into the same context representation.

(3) P✗C✗N✓: the only shared semantics among the correctly assigned utterances $x_{j\neq i}$ and misassigned utterance x_i are those of the subsequent utterances, i.e., $x_{i+1} \approx x_{j+1}$. Illustrated in the bottom part of Table 8, akin to the example for P✓C✗N✓, the semantics of x_i and x_j are similar, but x_j is more specific as it also requests the event name aside from the city. This mode is intrinsic to ELLoDAR for similar reasons as the P✓C✗N✓-mode, with the difference that the subsequent utterance, whose semantics are shared among x_i and x_j, has a more larger effect on the final representation compared to the previous utterance that does not share the same semantics.

Table 8 Illustration of common failure modes in ELLoDAR

Mode	Misassigned x_i	Reference correctly assigned x_j
P✗C✓N✗	$\mathbf{x_{i-1}}$: No pets are not allowed.	$\mathbf{x_{j-1}}$: Please confirm your visit. March 13th at Meadowood Apartments.
	$\mathbf{x_i}$: **Yes that one sounds good.**	$\mathbf{x_j}$: **Sounds great.**
	(y_i:select)	*(y_j:affirm)*
	$\mathbf{x_{i+1}}$: Do you want to schedule a visit?	$\mathbf{x_{j+1}}$: Your visit is now scheduled.
P✓C✗N✓	$\mathbf{x_{i-1}}$: How many bedrooms are you looking for in the apartment?	$\mathbf{x_{j-1}}$: What amount of bedrooms do you want in your apartment?
	$\mathbf{x_i}$: **I would like three bedrooms. I also need pets to be allowed.**	$\mathbf{x_j}$: **The apartment should have 2 bedrooms.**
	(y_i:inform.numbeds.petsallowed)	*(y_j:inform.numbeds)*
	$\mathbf{x_{i+1}}$: Which area are you looking in?	$\mathbf{x_{j+1}}$: Where do you want to look for property?
P✗C✗N✓	$\mathbf{x_{i-1}}$: I've heard that NYCFC vs Union will be a great one.	$\mathbf{x_{j-1}}$: I would like dates for certain events.
	$\mathbf{x_i}$: **What city should I check for the event?**	$\mathbf{x_j}$: **Is there a specific city and event that you are interested in today?**
	(y_i:request.city)	*(y_j: request.city.eventname)*
	$\mathbf{x_{i+1}}$: I want to find events in NYC. Something like Vertical Horizon which is supposed to be good.	$\mathbf{x_{j+1}}$: Please look for Spose in the LA area.

Each failure type is exemplified by a pair of utterances, x_j and x_i, assigned to the same cluster and accompanied by their respective gold states, y_j and y_i. **Left**: a sampled erroneously assigned utterance x_i to cluster $\mathcal{C}$, i.e., with $y_i \neq y_{\mathcal{C},\text{GOLD}}$. **Right**: a correctly assigned utterance x_j to cluster $\mathcal{C}$, serving as a representative instance of cluster $\mathcal{C}$, i.e., with $y_j = y_{\mathcal{C},\text{GOLD}}$. Underlined excerpts highlight the (contextual) distinctions causing the misassignment of x_i to $\mathcal{C}$

Note that our list of three failure modes is non-exhaustive, i.e., the following failure modes may also occur: $\mathrm{P}_{\text{✗}}\mathrm{C}_{\checkmark}\mathrm{N}_{\checkmark}$, $\mathrm{P}_{\checkmark}\mathrm{C}_{\text{✗}}\mathrm{N}_{\text{✗}}$, and $\mathrm{P}_{\checkmark}\mathrm{C}_{\checkmark}\mathrm{N}_{\text{✗}}$. However, as we found these failure modes not to occur in our randomly selected subset of 120 erroneously assigned utterances (as described below), we opted not to include them here. Aside from the presented failure modes inherent to ELLoDAR, there are also failure modes not related to ELLoDAR but inherent to k-means clustering itself, such as, e.g., outliers. For the instances where utterances cannot be categorized into one of the three presented failure modes, we include an extra “other” mode.

The distribution of common failure modes To better understand the frequency with which each of the three identified failure modes occur, we randomly sampled and manually annotated 20 (10 user and 10 system) misassigned utterances of CELLoDAR- induced clusters for each SGD domain (Events, Homes, Music, and Movies), CamRest676, and DSTC2 for a total of 120 utterances. Note that SimDial is excluded from this analysis, as CELLoDAR almost perfectly recovers its underlying gold structures.

Table 9 illustrates the distribution of error modes in the SGD, CamRest676, and DSTC2 datasets. Overall, $\mathrm{P}_{\checkmark}\mathrm{C}_{\text{✗}}\mathrm{N}_{\checkmark}$ is the most frequently occurring failure mode, with other types of errors occurring less frequently. The results for $\mathrm{P}_{\checkmark}\mathrm{C}_{\text{✗}}\mathrm{N}_{\checkmark}$ suggest that ELLoDAR faces difficulties handling the edge case in which the surrounding local dialogue context is shared between two utterances that are semantically different (i.e., with different underlying gold dialogue states). ELLoDAR cannot resolve this failure case well due to its sole reliance on local context and linear transformations. Therefore, future work could explore trading off efficiency by increasing complexity, e.g., by using non-linear transformations and/or more effectively exploiting subtler cues in larger dialogue contexts.

Table 9 The distribution of common failure modes

Mode	SGD	CamRest676	DSTC2	Average
$\mathrm{P}_{\text{✗}}\mathrm{C}_{\checkmark}\mathrm{N}_{\text{✗}}$	5%	0%	10%	5%
$\mathrm{P}_{\checkmark}\mathrm{C}_{\text{✗}}\mathrm{N}_{\checkmark}$	65%	40%	75%	60%
$\mathrm{P}_{\text{✗}}\mathrm{C}_{\text{✗}}\mathrm{N}_{\checkmark}$	16.25%	15%	10%	13.75%
Other	13.75%	45%	5%	21.25%

Percentage with which each type of failure mode occurs in $\text{CELLoDAR}_{\text{INT+EXT}}$ induced clusters based on MiniLM-L6. SGD results are averaged among the Events, Homes, Music and Movies domains

6.9 Limitations

Application domain First, ELLoDAR is designed specifically for clustering dialogue utterances, using the context of both preceding and subsequent utterances to produce contextual representations by fine-tuning a pretrained encoder. This means it cannot be used for task-oriented downstream tasks that only utilize preceding dialogue, such as intent classification and response generation.

Second, our work focuses on inducing dialogue structures at the utterance level (assigning utterances to states) and thus cannot be straightforwardly applied to the task of recovering dialogue structures based on slot type induction (assigning words or subphrases to states) as in [42].

Third, our specific focus was on extracting dialogue structures from *task-oriented* dialogues. Task-oriented dialogues typically involve two parties who exchange utterances consecutively. Therefore, we did not conduct experiments where we recover structures from dialogues with multiple consecutive user or system utterances, nor on multi-party dialogues (where more than two actors can appear in a single dialogue).

Finally, our work focuses on inducing dialogue structures from text only. However, in order to better recover structures, an interesting and unexplored direction for future work would be to consider a multi-modal setting where dialogues are augmented with other modalities, such as images.

Reliance on the ground truth number of dialogue states The main presented results rely on initializing the number of clusters of all considered models with the ground truth number of dialogue states. In practice, however, the ground truth number of states is unknown and thus would need to be estimated by inspecting a subset of the available dialogues. To assess the impact of not correctly setting the ground truth number of states, Section 6.5 analyzes the impact of overestimating the number of ground truth states by a factor of two, demonstrating that our proposed methods induce relatively pure clusters and still outperform both joint methods. An interesting direction for future work would thus be to investigate cluster algorithms that do not require the number of dialogue states as input such as, e.g., DBSCAN [64], Mean shift [65], and Affinity propagation [66].

Dialogue context representation strategy To include local context, we clustered the concatenation of the considered utterance’s representation and its adjacent utterances’ representations, rather than leveraging more advanced techniques that integrate different views of data such as, e.g., Multi-View k-means [67]. We leave the latter for future work.

Training time performance analysis The training time performance discussion Section 6.6 involved comparing the training times of state-of-the-art joint models to those of our approaches. Because training time is affected by factors like implementation, batch size, etc. the reported times should be interpreted as an indication rather than as exact numbers.

Pre-training sentence encoders Because our work focuses on computational efficiency, we did not further experiment with specifically pretraining sentence encoders on each distinct domain or dataset. However, the preliminary results of TOD-BERT on SGD, discussed in Section 6.4, suggest that such specific pretraining might be beneficial. It is further worth noting that the effectiveness and efficiency of ELLoDAR's linear-to-linear regression is in part attributed to ELLoDAR building upon an out-of-the-box pre-trained sentence encoder that already has undergone substantial computation efforts: training such encoders from scratch is computationally expensive to obtain.

Generalizability to additional human-human dialogues While compared to prior DSI works, our experiments cover a broader range of datasets, i.e., 4 commonly used conversational datasets (DSTC2, CamRest676, SimDial, and SGD), it is worth noting that SimDial comprises synthetic dialogues, SGD and DSTC2 human-machine dialogues, and CamRest676 human-human dialogues. As such, there remains uncertainty about the generalizability of CELLoDAR to human-human dialogues other than CamRest676. Unfortunately, due to the lack of utterance-level annotated conversational datasets (as opposed to slot extraction datasets, e.g., MultiWOZ [51]), we were unable to cover additional datasets, and defer exploring this to future work.

7 Implications of the presented research results

The findings in this work have implications on the relatively underexplored DSI domain. Our main goal was to design an efficient DSI model, which we argued to be essential in practical settings, e.g., when users need to run the DSI model multiple times with different numbers of dialogue states to recover the optimal structure. By revisiting and further developing the cluster-based method of [18, 19], we demonstrated that simple DSI models can be orders of magnitude faster yet still outperform more complex existing models. Therefore, we want to emphasize that pragmatic architectural choices, which may not necessarily follow the trend of aiming for performance gains through more complex/advanced (neural) models, may lead to both efficiency and performance improvements over more complex models. We hope that this will encourage the community to pursue model efficiency as an important design aspect, besides model effectiveness.

Second, as no publicly available framework for benchmarking DSI models currently exists, we release our modified datasets and evaluation to accelerate future DSI research, for which we hope that our simple CELLoDAR approach will serve as a strong baseline.

8 Conclusions

Unlike recently proposed DSI models that jointly learn to encode and cluster utterances, we revisited an efficient cluster-based approach that proceeds in two steps. It first encodes utterances as vectors, after which it clusters the obtained representations to induce the dialogue structure in the second step. However, the previously proposed cluster-based approach encodes utterances as bag-of-words or skip-thought vectors without using dialogue context. Hence, we proposed to adopt more powerful transformer-based sentence encoders and contributed ELLoDAR, a highly efficient approach for learning dialogue aware representations. ELLoDAR trains linear transformations with a vector-to-vector regression objective in the encoding space of a frozen sentence encoder using a local context window. Extensive experiments on representative DSI datasets show that: (i) the cluster-based approach outperforms the recent joint models when using transformer-based encoders to represent utterances, (ii) clustering ELLoDAR's representations further improves performance consistently, while being orders of magnitude faster than the joint models. We release our datasets (which are variants of commonly adopted DSI datasets), evaluation, and models as a common benchmark for DSI, which is currently missing.

Appendix A: Additional results

We present the full *turn*-based results on the individual domains of SimDial and SGD in Tables 10–11, and the full *utterance*-based results of SimDial and SGD are shown in Tables 12–13. Both the full *turn* and *utterance*-based results follow the same trends as reported in Section 5: (i) almost all cluster baselines outperform the joint models, (ii) the best CELLoDAR-model consistently outperforms the best cluster baseline, and (iii) the results for dedicated sentence encoders (MiniLM and TOD-BERT) are consistently higher than those of GloVe. With as exception the *Events* domain of SGD for which the cluster baselines are often outperformed by the joints models. Additionally, on SimDial and SGD, some of the GloVe$_C$ baselines perform worse than the joint models.

Table 10 Full results on SimDial

SimDial	Weather			Bus			Restaurant			Movies			Average		
Model	ARI	AMI	FM	ARI	AMI	FM	ARI	AMI	FM	ARI	AMI	FM	ARI	AMI	FM
Joint models															
DVRNN [16]	55.3	70.8	64.5	25.5	50.7	43.7	52.6	64.9	58.8	28.5	58.8	36.7	40.5	61.3	50.9
SVRNN [17]	21.1	43.3	32.7	39.4	65.5	53.7	17.2	40.3	31.3	25.1	54.6	33.9	25.7	50.9	37.9
Cluster baselines															
- GloVe$_{\text{C}}$ [18, 19]	47.3	72.5	58.8	41.9	67.2	52.4	55.5	72.1	64.0	47.1	74.3	55.8	48.0$_{\pm 3.7}$	71.5$_{\pm 2.4}$	57.7$_{\pm 3.3}$
- GloVe$_{\text{PC}}$	67.9	82.7	75.0	64.3	78.7	71.4	63.8	81.2	70.1	67.0	83.6	72.4	65.8$_{\pm 3.7}$	81.6$_{\pm 1.8}$	72.2$_{\pm 3.1}$
- GloVe$_{\text{PCN}}$	64.1	80.2	71.1	87.8	91.0	90.0	72.3	84.6	77.0	76.2	87.4	79.7	75.1$_{\pm 3.7}$	85.8$_{\pm 1.7}$	79.4$_{\pm 3.1}$
CELLoDAR															
- GloVe$_{\text{INT}}$	99.6	99.5	99.7	**99.0**	**98.3**	**99.2**	99.6	99.4	99.6	94.1	96.9	95.1	98.1$_{\pm 0.0}$	98.5$_{\pm 0.0}$	98.4$_{\pm 0.0}$
- GloVe$_{\text{EXT}}$	92.0	95.5	93.7	77.8	84.1	81.8	**100**	99.9	**100**	93.8	96.8	94.8	90.9$_{\pm 0.0}$	94.1$_{\pm 0.0}$	92.6$_{\pm 0.0}$
- GloVe$_{\text{INT+EXT}}$	**100**	**99.9**	**100**	93.8	95.9	94.9	**100**	**100**	**100**	**100**	**99.9**	**100**	**98.4**$_{\pm 0.0}$	**98.9**$_{\pm 0.0}$	**98.7**$_{\pm 0.0}$
Cluster baselines															
- MiniLM-L6$_{\text{C}}$ [18, 19]	67.4	79.6	72.5	64.3	82.4	70.3	64.5	79.1	69.8	59.6	79.2	64.0	64.0$_{\pm 2.4}$	80.1$_{\pm 1.8}$	69.2$_{\pm 2.2}$
- MiniLM-L6$_{\text{PC}}$	75.2	86.9	78.8	75.7	87.1	78.8	72.4	86.6	75.9	76.6	87.5	79.2	75.0$_{\pm 3.6}$	87.0$_{\pm 1.4}$	78.2$_{\pm 3.2}$
- MiniLM-L6$_{\text{PCN}}$	87.5	90.0	89.4	90.0	92.6	91.4	86.1	90.5	87.9	77.6	86.7	80.2	85.3$_{\pm 2.5}$	90.0$_{\pm 1.3}$	87.2$_{\pm 2.2}$
CELLoDAR															
- MiniLM-L6$_{\text{INT}}$	99.9	99.8	99.9	**99.9**	**99.7**	**99.9**	99.8	99.8	99.8	**100**	99.8	**100**	**99.9**$_{\pm 0.0}$	**99.8**$_{\pm 0.0}$	**99.9**$_{\pm 0.0}$
- MiniLM-L6$_{\text{EXT}}$	**100**	**100**	**100**	97.8	98.5	98.1	**100**	**100**	**100**	**100**	**100**	**100**	99.5$_{\pm 0.0}$	99.6$_{\pm 0.0}$	99.5$_{\pm 0.0}$
- MiniLM-L6$_{\text{INT+EXT}}$	**100**	**100**	**100**	98.3	98.8	98.5	**100**	**100**	**100**	**100**	**100**	**100**	99.6$_{\pm 0.0}$	99.7$_{\pm 0.0}$	99.6$_{\pm 0.0}$
Cluster baselines															
- TOD-BERT$_{\text{C}}$ [18, 19]	68.6	82.2	75.4	62.9	77.9	70.2	77.2	88.3	81.7	84.8	91.2	87.4	73.4$_{\pm 3.4}$	84.9$_{\pm 2.1}$	78.7$_{\pm 2.8}$
- TOD-BERT$_{\text{PC}}$	81.1	88.7	85.1	73.7	85.4	78.7	82.0	90.1	85.6	79.4	88.5	83.0	79.0$_{\pm 2.3}$	88.2$_{\pm 1.3}$	83.1$_{\pm 1.9}$
- TOD-BERT$_{\text{PCN}}$	82.1	89.3	85.9	85.4	91.3	88.0	95.3	97.5	96.2	89.9	94.7	91.5	88.2$_{\pm 2.6}$	93.2$_{\pm 1.3}$	90.4$_{\pm 2.2}$
CELLoDAR															
- TOD-BERT$_{\text{INT}}$	99.9	99.9	**100**	99.7	99.4	99.8	99.8	99.7	99.8	99.8	99.7	99.8	**99.8**$_{\pm 0.0}$	**99.7**$_{\pm 0.0}$	**99.8**$_{\pm 0.0}$
- TOD-BERT$_{\text{EXT}}$	**100**	**100**	**100**	91.2	94.6	92.8	**100**	**100**	**100**	**100**	**100**	**100**	97.8$_{\pm 0.0}$	98.6$_{\pm 0.0}$	98.2$_{\pm 0.0}$
- TOD-BERT$_{\text{INT+EXT}}$	**100**	**100**	**100**	91.7	95.0	93.2	**100**	**100**	**100**	**100**	**100**	**100**	97.9$_{\pm 0.0}$	98.7$_{\pm 0.0}$	98.3$_{\pm 0.0}$

Turn-based evaluation. The **best** model is typeset in bold and the runner-up is underlined

Table 11 Full results on SGD with averaged standard deviations

The Schema Guided Dialogues Dataset	Events			Homes			Music			Movies			Average		
Model	ARI	AMI	FM	ARI	AMI	FM	ARI	AMI	FM	ARI	AMI	FM	ARI	AMI	FM
Joint models															
DVRNN [16]	20.1	25.3	25.3	9.4	23.5	20.4	17.1	29.8	28.0	10.5	22.6	17.1	14.3	25.3	22.7
SVRNN [17]	16.2	24.8	19.2	8.5	28.5	21.4	28.1	38.8	35.5	12.5	21.3	20.7	16.3	28.4	24.2
Cluster baselines															
- GloVe$_{C}$ [18, 19]	7.7	37.3	14.5	17.6	45.4	23.5	19.6	48.8	24.3	11.8	38.4	16.8	$14.2_{\pm 0.9}$	$42.4_{\pm 0.8}$	$19.8_{\pm 1.1}$
- GloVe$_{PC}$	14.7	48.2	21.1	26.7	55.1	32.2	30.2	58.9	34.9	17.1	48.5	23.7	$22.2_{\pm 1.0}$	$52.7_{\pm 0.7}$	$28.0_{\pm 1.1}$
- GloVe$_{PCN}$	$\underline{17.8}$	$\underline{50.8}$	24.4	$\underline{27.6}$	$\underline{55.9}$	$\underline{33.2}$	34.2	$\underline{62.1}$	38.8	25.6	**54.2**	31.0	$26.3_{\pm 2.1}$	$\underline{55.7}_{\pm 1.0}$	$31.8_{\pm 2.2}$
CELLODAR															
- GloVe$_{INT}$	14.1	45.7	22.4	24.2	50.4	29.7	37.0	56.9	42.4	26.3	50.5	31.2	$25.4_{\pm 1.6}$	$50.9_{\pm 0.6}$	$31.4_{\pm 1.6}$
- GloVe$_{EXT}$	16.4	47.9	$\underline{25.4}$	25.7	52.6	32.0	$\underline{38.1}$	58.5	$\underline{43.0}$	25.5	50.9	**31.6**	$\underline{26.4}_{\pm 1.4}$	$52.5_{\pm 0.5}$	$\underline{33.0}_{\pm 1.4}$
- GloVe$_{INT+EXT}$	**23.5**	**55.8**	**31.3**	**28.9**	**56.5**	**34.3**	**41.8**	**62.2**	**46.5**	**26.1**	$\underline{53.0}$	$\underline{31.3}$	$\mathbf{30.1}_{\pm 1.4}$	$\mathbf{56.9}_{\pm 0.6}$	$\mathbf{35.9}_{\pm 1.4}$
Cluster baselines															
- MiniLM-L6$_{C}$ [18, 19]	12.5	45.3	21.5	24.8	55.2	31.1	35.2	62.4	40.0	20.4	51.6	28.1	$23.3_{\pm 1.6}$	$53.6_{\pm 1.1}$	$30.2_{\pm 1.8}$
- MiniLM-L6$_{PC}$	19.6	57.6	27.7	36.4	$\underline{66.7}$	41.1	44.6	72.6	48.4	25.5	57.4	32.0	$31.5_{\pm 1.9}$	$63.6_{\pm 0.8}$	$37.3_{\pm 1.7}$
- MiniLM-L6$_{PCN}$	21.1	57.6	29.0	37.5	65.3	42.8	47.7	**72.9**	52.2	31.0	61.2	36.7	$34.3_{\pm 2.2}$	$64.2_{\pm 1.0}$	$40.2_{\pm 2.3}$
CELLODAR															
- MiniLM-L6$_{INT}$	**26.9**	**60.9**	**35.4**	36.2	64.5	41.3	51.5	72.3	55.3	36.2	61.0	39.8	$37.7_{\pm 2.3}$	$\underline{64.7}_{\pm 0.8}$	$42.9_{\pm 2.1}$
- MiniLM-L6$_{EXT}$	20.2	54.5	30.3	$\underline{40.0}$	65.2	$\underline{46.0}$	**55.3**	70.0	**59.3**	**40.5**	$\underline{63.5}$	**45.5**	$\underline{39.0}_{\pm 2.3}$	$63.3_{\pm 0.8}$	$\underline{45.3}_{\pm 2.1}$
- MiniLM-L6$_{INT+EXT}$	$\underline{25.8}$	$\underline{60.5}$	**35.4**	**41.9**	**67.7**	**46.9**	$\underline{53.6}$	$\underline{72.7}$	$\underline{57.7}$	$\underline{40.3}$	**64.0**	$\underline{44.5}$	$\mathbf{40.4}_{\pm 2.6}$	$\mathbf{66.2}_{\pm 0.7}$	$\mathbf{46.0}_{\pm 2.5}$
Cluster baselines															
- TODBERT-jnt$_{C}$ [18, 19]	22.8	51.9	32.4	28.3	56.3	34.9	38.8	63.7	44.1	27.8	55.9	35.1	$29.4_{\pm 2.2}$	$56.9_{\pm 0.7}$	$36.6_{\pm 2.2}$
- TODBERT-jnt$_{PC}$	19.8	54.9	28.9	31.5	61.0	37.0	49.2	**73.8**	53.2	31.2	61.3	36.2	$32.9_{\pm 2.3}$	$62.7_{\pm 0.8}$	$38.8_{\pm 2.1}$
- TODBERT-jnt$_{PCN}$	28.2	59.6	35.2	39.7	65.6	43.4	$\underline{55.1}$	$\underline{72.9}$	$\underline{59.1}$	37.7	65.2	42.1	$40.1_{\pm 2.6}$	$65.8_{\pm 0.9}$	$45.0_{\pm 2.4}$
CELLODAR															
- TOD-BERT$_{INT}$	39.3	64.3	45.7	35.8	60.3	41.2	47.3	69.0	51.6	36.3	59.1	40.2	$39.7_{\pm 2.8}$	$63.2_{\pm 0.9}$	$44.7_{\pm 2.6}$
- TOD-BERT$_{EXT}$	$\underline{40.5}$	$\underline{63.9}$	$\underline{48.2}$	$\underline{49.6}$	$\underline{66.7}$	$\underline{54.3}$	54.8	68.2	58.8	**47.4**	**66.9**	**51.1**	$\underline{48.1}_{\pm 3.0}$	$\underline{66.4}_{\pm 0.7}$	$\underline{53.1}_{\pm 2.6}$
- TOD-BERT$_{INT+EXT}$	**45.5**	**66.9**	**51.8**	**52.2**	**70.0**	**55.9**	**58.0**	71.5	**61.6**	$\underline{43.5}$	$\underline{66.1}$	$\underline{47.5}$	$\mathbf{49.8}_{\pm 3.2}$	$\mathbf{68.6}_{\pm 0.9}$	$\mathbf{54.2}_{\pm 2.8}$

Turn-based evaluation. The **best** model is typeset in bold and the $\underline{\text{runner-up}}$ is underlined

Table 12 Full results on SimDial with averaged standard deviations

SimDial	Weather			Bus			Restaurant			Movie			Average		
Model	ARI	AMI	FM	ARI	AMI	FM	ARI	AMI	FM	ARI	AMI	FM	ARI	AMI	FM
Cluster baselines															
- GloVe$_{C}$ [18, 19]	53.7	71.3	63.0	49.6	65.6	58.1	55.9	72.5	64.0	47.7	69.3	57.0	$51.7_{\pm 4.9}$	$69.7_{\pm 3.3}$	$60.5_{\pm 3.7}$
- GloVe$_{PC}$	72.7	84.3	77.9	67.5	78.4	72.7	74.3	85.3	78.5	68.8	83.4	73.3	$70.8_{\pm 5.6}$	$82.9_{\pm 2.7}$	$75.6_{\pm 4.6}$
- GloVe$_{PCN}$	65.0	79.9	71.8	87.6	90.7	89.5	78.7	87.6	82.1	79.5	88.6	82.3	$77.7_{\pm 3.6}$	$86.7_{\pm 1.7}$	$81.4_{\pm 3.0}$
CELLoDAR															
- GloVe$_{INT}$	99.5	99.2	99.6	**98.6**	**98.1**	**98.8**	99.5	99.2	99.5	95.5	97.0	96.1	**98.3**$_{\pm 0.0}$	$98.4_{\pm 0.0}$	$98.5_{\pm 0.0}$
- GloVe$_{EXT}$	96.2	97.4	96.8	82.9	89.1	85.5	**100**	**100**	**100**	96.2	97.6	96.7	$93.8_{\pm 0.0}$	$96.0_{\pm 0.0}$	$94.8_{\pm 0.0}$
- GloVe$_{INT+EXT}$	**100**	**99.9**	**100**	93.1	95.7	94.3	**100**	**100**	**100**	**100**	99.9	**100**	$98.2_{\pm 0.0}$	**98.9**$_{\pm 0.0}$	**98.6**$_{\pm 0.0}$
Cluster baselines															
- MiniLM-L6$_{C}$ [18, 19]	60.4	76.7	69.0	68.9	80.3	73.8	61.2	77.7	67.7	68.4	83.5	73.1	$64.7_{\pm 4.0}$	$79.5_{\pm 2.2}$	$70.9_{\pm 3.1}$
- MiniLM-L6$_{PC}$	82.3	89.3	85.5	76.9	86.4	80.5	80.3	88.6	83.5	81.5	89.7	84.0	$80.2_{\pm 3.4}$	$88.5_{\pm 1.8}$	$83.4_{\pm 2.8}$
- MiniLM-L6$_{PCN}$	90.5	93.2	92.3	92.6	94.8	93.8	88.2	92.8	90.1	82.9	90.4	85.3	$88.6_{\pm 1.8}$	$92.8_{\pm 1.0}$	$90.4_{\pm 1.5}$
CELLoDAR															
- MiniLM-L6$_{INT}$	99.7	99.6	99.7	**99.6**	**99.5**	**99.7**	99.8	99.8	99.8	99.8	99.8	99.8	**99.7**$_{\pm 0.0}$	**99.7**$_{\pm 0.0}$	**99.8**$_{\pm 0.0}$
- MiniLM-L6$_{EXT}$	**100**	**100**	**100**	97.5	97.9	97.9	**100**	**100**	**100**	**100**	**100**	**100**	$99.4_{\pm 0.0}$	$99.5_{\pm 0.0}$	$99.5_{\pm 0.0}$
- MiniLM-L6$_{INT+EXT}$	**100**	**100**	**100**	98.0	98.3	98.3	**100**	**100**	**100**	**100**	**100**	**100**	$99.5_{\pm 0.0}$	$99.6_{\pm 0.0}$	$99.6_{\pm 0.0}$
Cluster baselines															
- TOD-BERT$_{C}$ [18, 19]	81.0	87.5	84.5	66.2	79.5	71.7	83.8	90.1	86.4	88.7	92.4	90.3	$79.9_{\pm 3.2}$	$87.4_{\pm 1.9}$	$83.2_{\pm 2.7}$
- TOD-BERT$_{PC}$	90.1	92.9	92.0	77.3	86.7	81.1	87.2	91.2	89.4	86.3	89.7	88.2	$85.2_{\pm 2.5}$	$90.1_{\pm 1.5}$	$87.7_{\pm 2.1}$
- TOD-BERT$_{PCN}$	91.1	93.8	92.8	92.7	94.8	93.9	96.4	97.7	97.0	92.7	95.5	93.7	$93.2_{\pm 1.3}$	$95.5_{\pm 0.8}$	$94.3_{\pm 1.1}$
CELLoDAR															
- TOD-BERT$_{INT}$	99.7	99.6	99.7	**99.2**	**99.1**	**99.3**	99.8	99.8	99.8	99.8	99.8	99.8	**99.6**$_{\pm 0.0}$	**99.6**$_{\pm 0.0}$	**99.7**$_{\pm 0.0}$
- TOD-BERT$_{EXT}$	**100**	**100**	**100**	94.3	95.8	95.2	**100**	**100**	**100**	**100**	**100**	**100**	$98.6_{\pm 0.0}$	$98.9_{\pm 0.0}$	$98.8_{\pm 0.0}$
- TOD-BERT$_{INT+EXT}$	**100**	**100**	**100**	94.8	96.2	95.6	**100**	**100**	**100**	**100**	**100**	**100**	$98.7_{\pm 0.0}$	$99.0_{\pm 0.0}$	$98.9_{\pm 0.0}$

Utterance-based evaluation. The **best** model is typeset in bold and the runner-up is underlined

Table 13 Full results on SGD with averaged standard deviations

The Schema Guided Dialogues Dataset															
	Events			Homes			Music			Movies			Average		
Model	ARI	AMI	FM	ARI	AMI	FM	ARI	AMI	FM	ARI	AMI	FM	ARI	AMI	FM
Cluster baselines															
- GloVe$_{C}$ [18, 19]	23.4	53.3	27.0	39.4	60.7	42.9	38.9	57.0	44.5	32.4	53.5	36.7	$33.5_{\pm 1.8}$	$56.1_{\pm 0.9}$	$37.8_{\pm 1.6}$
- GloVe$_{PC}$	24.9	56.8	28.9	38.2	63.7	42.1	41.9	62.2	47.3	35.8	59.1	40.1	$35.2_{\pm 1.6}$	$60.4_{\pm 0.9}$	$39.6_{\pm 1.5}$
- GloVe$_{PCN}$	28.9	60.2	33.2	41.3	$\underline{66.4}$	45.3	46.6	$\underline{67.7}$	51.7	39.7	$\underline{64.9}$	44.3	$39.1_{\pm 1.9}$	$\underline{64.8}_{\pm 1.0}$	$43.6_{\pm 1.8}$
CELLODAR															
- GloVe$_{INT}$	28.7	57.1	32.9	40.6	63.0	44.0	46.8	64.1	$\underline{51.8}$	43.4	61.7	47.2	$39.9_{\pm 1.7}$	$61.5_{\pm 0.6}$	$44.0_{\pm 1.5}$
- GloVe$_{EXT}$	$\underline{36.5}$	$\underline{61.4}$	$\underline{40.3}$	**44.2**	65.7	$\underline{47.5}$	$\underline{46.8}$	65.4	51.6	**47.3**	63.8	**50.9**	$\underline{43.7}_{\pm 1.5}$	$64.1_{\pm 0.7}$	$\underline{47.6}_{\pm 1.4}$
- GloVe$_{INT+EXT}$	**37.9**	**66.2**	**42.3**	**44.2**	**68.7**	**47.9**	**50.4**	**69.4**	**55.1**	$\underline{47.0}$	**66.8**	$\underline{50.7}$	**44.9**$_{\pm 1.7}$	**67.7**$_{\pm 0.7}$	**49.0**$_{\pm 1.5}$
Cluster baselines															
- MiniLM-L6$_{C}$ [18, 19]	31.6	60.7	36.2	46.6	69.2	50.0	52.7	71.2	57.1	43.5	65.4	47.9	$43.6_{\pm 2.4}$	$66.6_{\pm 1.0}$	$47.8_{\pm 2.2}$
- MiniLM-L6$_{PC}$	32.9	66.0	38.0	47.1	72.8	50.9	56.1	77.0	60.3	44.4	69.1	49.0	$45.1_{\pm 2.0}$	$71.2_{\pm 0.8}$	$49.5_{\pm 1.9}$
- MiniLM-L6$_{PCN}$	35.5	68.3	41.0	48.0	75.4	52.2	58.4	78.5	62.6	48.2	72.4	52.9	$47.5_{\pm 2.2}$	$73.6_{\pm 1.0}$	$52.2_{\pm 2.1}$
CELLODAR															
- MiniLM-L6$_{INT}$	40.2	$\underline{70.0}$	45.5	53.4	$\underline{76.6}$	56.6	$\underline{61.0}$	**79.2**	$\underline{65.0}$	53.8	73.0	56.9	$52.1_{\pm 2.1}$	$\underline{74.7}_{\pm 0.6}$	$56.0_{\pm 1.9}$
- MiniLM-L6$_{EXT}$	$\underline{41.3}$	68.3	$\underline{46.3}$	$\underline{55.4}$	$\underline{76.6}$	$\underline{58.7}$	59.6	76.5	63.7	**58.8**	$\underline{74.9}$	**62.4**	$\underline{53.8}_{\pm 1.7}$	$74.1_{\pm 0.7}$	$\underline{57.8}_{\pm 1.6}$
- MiniLM-L6$_{INT+EXT}$	**44.4**	**72.5**	**49.7**	**56.8**	**78.7**	**60.1**	**61.8**	**79.2**	**65.7**	$\underline{58.0}$	**75.1**	$\underline{61.5}$	**55.3**$_{\pm 2.3}$	**76.4**$_{\pm 0.9}$	**59.2**$_{\pm 2.1}$
Cluster baselines															
- TOD-BERT$_{C}$ [18, 19]	42.0	64.7	46.4	46.8	70.0	50.4	52.4	72.4	56.9	49.6	69.5	53.8	$47.7_{\pm 2.2}$	$69.2_{\pm 0.9}$	$51.9_{\pm 2.0}$
- TOD-BERT$_{PC}$	36.5	65.4	41.1	46.8	70.2	50.4	56.6	77.2	60.8	47.2	70.4	51.4	$46.8_{\pm 2.5}$	$70.8_{\pm 1.1}$	$50.9_{\pm 2.4}$
- TOD-BERT$_{PCN}$	39.2	69.4	44.2	51.2	74.8	54.9	$\underline{61.9}$	**79.8**	$\underline{65.9}$	52.1	75.0	56.4	$51.1_{\pm 2.5}$	$74.7_{\pm 0.8}$	$55.3_{\pm 2.3}$
CELLODAR															
- TOD-BERT$_{INT}$	50.7	72.1	54.8	50.0	72.6	53.4	58.2	76.9	62.3	54.2	72.5	57.4	$53.2_{\pm 2.4}$	$73.5_{\pm 0.7}$	$57.0_{\pm 2.1}$
- TOD-BERT$_{EXT}$	**53.9**	$\underline{73.4}$	**57.8**	$\underline{58.2}$	$\underline{75.9}$	$\underline{61.2}$	60.0	75.9	64.0	**61.0**	$\underline{76.8}$	**64.3**	$\underline{58.3}_{\pm 2.0}$	$\underline{75.5}_{\pm 0.7}$	$\underline{61.8}_{\pm 1.8}$
- TOD-BERT$_{INT+EXT}$	$\underline{53.4}$	**74.1**	$\underline{57.5}$	**58.6**	**77.9**	**61.7**	**62.7**	$\underline{78.6}$	**66.5**	$\underline{59.6}$	**77.0**	$\underline{63.0}$	**58.6**$_{\pm 1.8}$	**76.9**$_{\pm 0.7}$	**62.2**$_{\pm 1.6}$

♠: [18, 19]. *Utterance*-based evaluation. The **best** model is typeset in bold and the runner-up is underlined

Table 14 Randomly sampled dialogues from each SimDial domain with corresponding gold annotations

SimDial Utterances	Label
Weather	
`[SYS]`: Weather bot is here. What do you need?	[greet, request-need]
`[USR]`: Whats up? Whats the weather?	[greet, request-default]
`[SYS]`: Which city are you interested in?	[request-loc]
`[USR]`: Weather at Philadelphia	[inform-loc]
`[SYS]`: What date are you interested?	[request-datetime]
`[USR]`: Weather tomorrow	[inform-datetime]
`[SYS]`: Your weather report 2 is here. Anything else?	[inform-default, request-happy]
`[USR]`: All good. Thats all.	[goodbye, satisfy]
`[SYS]`: See you next time.	[goodbye]
Bus	
`[SYS]`: Ask me about bus information. How can I help?	[greet, request-need]
`[USR]`: Hello robot. Recommend me a bus to take.	[greet, request-default]
`[SYS]`: Where are you leaving from?	[request-fromloc]
`[USR]`: Leaving from Downtown.	[inform-fromloc]
`[SYS]`: Where do you want to take off?	[request-toloc]
`[USR]`: Destination is Lawrance.	[inform-toloc]
`[SYS]`: What time do you need the bus?	[request-datetime]
`[USR]`: I am leaving on 23.	[inform-datetime]
`[SYS]`: Bus 50 can take you there. What else can I do?	[inform-default, request-happy]
`[USR]`: I have more requests. How long do I need to wait?	[morerequest, request-arrivein]
`[SYS]`: The bus will arrive in 25 minutes. What else can I do?	[inform-arrivein, request-happy]
`[USR]`: I have all I need. See you.	[goodbye, satisfy]
`[SYS]`: Goodbye.	[goodbye]
Restaurant	
`[SYS]`: Welcome to restaurant recommendation system. How can I help?	[greet, request-need]
`[USR]`: Hello robot. I need a restaurant.	[greet, request-default]
`[SYS]`: Which place?	[request-loc]
`[USR]`: At Philadelphia.	[inform-loc]
`[SYS]`: What type of restaurant?	[request-foodpref]
`[USR]`: I like Indian food.	[inform-foodpref]
`[SYS]`: Restaurant 44 is a good choice. What else can I do?	[inform-default, request-happy]
`[USR]`: I have more requests. What kind of parking does it have?.	[morerequest, request-parking]
`[SYS]`: This place has street parking. What else can I do?	[inform-parking, request-happy]
`[USR]`: I have all I need. See you.	[goodbye, satisfy]
`[SYS]`: See you next time.	[goodbye]
Movie	
`[USR]`: Hello robot. Give me some good suggestions about movies.	[greet, request-default]
`[SYS]`: What genre do you like?	[request-genre]

Table 14 continued

SimDial Utterances	Label
[USR]: I love Drama ones.	[inform-genre]
[SYS]: Whats the time period?	[request-years]
[USR]: In 2010-present.	[inform-years]
[SYS]: Movie from what country?	[request-country]
[USR]: Germany.	[inform-country]
[SYS]: Movie 193 is a good choice. Are you happy about my answer?	[inform-default, request-happy]
[USR]: Not done yet. Which company?	[morerequest, request-company]
[SYS]: The movie is from MGM. What else can I do?	[inform-company, request-happy]
[USR]: One more thing. Whats the rating?	[morerequest, request-rating]
[SYS]: The rating is 1. Are you happy about my answer?	[inform-rating, request-happy]
[USR]: No more questions. See you.	[goodbye, satisfy]
[SYS]: Goodbye.	[goodbye]

Table 15 Randomly sampled dialogues from CamRest676 and DSTC2 datasets with corresponding gold annotations

Utterances	Label
CamRest676	
[USR]: I am looking for a moderately priced restaurant in the west part of town.	[inform.area, inform.pricerange]
[SYS]: Meghna is an Indian restaurant, it is in the moderate price range, and it is in the west part of town. Would you like the phone number?	[phone]
[USR]: yes, and the address and postcode please.	[request.address, request.phone, request.postcode]
[SYS]: The phone number for meghna is 01223 727410 and they are located at 205 Victoria Road Chesterton. The postcode is C.B 4, 3 L.F.	
[USR]: Perfect. Thanks for the information. Goodbye.	
[SYS]: Thank you for choosing our system. Good bye	
DSTC2	
[SYS]: Hello , welcome to the Cambridge restaurant system? You can ask for restaurants by area , price range or food type . How may I help you?	[welcomemsg]
[USR]: uh korean food	[informfood]
[SYS]: Im sorry but there is no restaurant serving creative food	[canthelpfood]
[USR]: korean food	[informfood]
[SYS]: Sorry would you like creative or korean food?	[selectfood]
[USR]: korean	[informfood]
[SYS]: little seoul serves korean food	[informfood, offername]

Note that there are missing annotations for CamRest676

Table 15 continued

Utterances	Label
[USR]: address	[requestaddr]
[SYS]: Sure , little seoul is on 108 Regent Street City Centre	[informaddr, offername]
[USR]: phone number	[requestphone]
[SYS]: The phone number of little seoul is 01223 308681.	[informphone, offername]
[USR]: thank you good bye	[bye, thankyou]

Table 16 Randomly sampled dialogues from the Schema Guided Dialogues with corresponding gold annotations

The Schema Guided Dialogues Utterances	Label
Music	
[USR]: Can you help? I am trying to locate some songs.	[informintent.[lookupmusic]]
[SYS]: Yes, be happy to. I have 10 songs to choose from. I recommend Take Me Away by Scotty Sire from the Ruin Your Party album.	[informcount.count, offer.album, offer.artist, offer.songname]
[USR]: That one sounds wonderful	[select.]
[SYS]: Shall I play it now for you?	[offerintent.[playmedia]]
[USR]: Yes, Id like to hear that one	[affirmintent.]
[SYS]: To confirm: youd like to hear Take Me Away on the TV?	[confirm.playbackdevice, confirm.songname]
[USR]: Id rather hear it on my kitchen please	[inform.playbackdevice, negate]
[SYS]: To confirm: You wish to hear Take Me Away on your kitchen speaker. Is this correct?	[confirm.playbackdevice, confirm.songname]
[USR]: Yes it is, thank you. Whats the genre of that song?	[affirm, request.genre]
[SYS]: It is a Pop song. Your selection has started.	[inform.genre, notifysuccess]
[USR]: Thank you for your help.	[thankyou]
[SYS]: Is there anything else I can do?	[reqmore]
[USR]: No that will be all for now.	[negate, thankyou]
[SYS]: Have a great day.	[goodbye]
Homes	
[USR]: I am looking for an apartment.	[informintent.[findapartment]]
[SYS]: In which area?	[request.area]
[USR]: In Gilroy area.	[inform.area]
[SYS]: With how many bed rooms?	[request.numberofbeds]
[USR]: With 1 bed room.	[inform.numberofbeds]
[SYS]: There are 4 matches. Mission Park at 766 1st Street has 1 bed room and 1 bath at a price of 2000 $/month.	[informcount.count, offer.address, offer.numberofbaths, offer.numberofbeds, offer.propertyname, offer.rent]
[USR]: Show me other apartments with 1 bath, please.	[inform.numberofbaths, requestalts]
[SYS]: Another place would be Park View Apartments at 181 Pierce Street that has 1 bed room and 1 bath as well, it costs 5050 $/month.	[offer.address, offer.numberofbaths, offer.numberofbeds, offer.propertyname, offer.rent]
[USR]: Is it furnished?	[request.furnished]
[SYS]: No, it hasnt furniture.	[inform.furnished]
[USR]: It sounds good, I would like to schedule a visit on the 14th.	[inform.visitdate, informintent.[schedulevisit], select.]

Table 16 continued

The Schema Guided Dialogues Utterances	Label
[SYS]: Please confirm these details: scheduling of a visit in Park View Apartments on March 14th.	[confirm.propertyname, confirm.visitdate]
[USR]: Yes, thank you, whats the phone number of the property?	[affirm, request.phonenumber]
[SYS]: Visit confirmed. Propertys phone number is 408-842-6341.	[inform.phonenumber, notifysuccess]
[USR]: Thank you. Thats all.	[goodbye, thankyou]
[SYS]: My pleasure. Have a nice day.	[goodbye]
Events	
[USR]: Im looking for events.	[informintent.[findevents]]
[SYS]: What type of events are you looking for? Music, sports, etc. What city are you looking in?	[request.city, request.eventtype]
[USR]: I want to find music events in NY next Friday.	[inform.city, inform.date, inform.eventtype]
[SYS]: There are 10 events that match your request. Amber Run Brooklyn plays at Warsaw.	[informcount.count, offer.eventname, offer.venue]
[USR]: Sounds good, I want to find more events.	[informintent.[geteventdates], select.]
[SYS]: I found 1 events next Friday. Amber Run Brooklyn is at Warsaw.	[informcount.count, offer.date, offer.eventname, offer.venue]
[USR]: When does it start?	[request.time]
[SYS]: It starts at 5 pm.	[inform.time]
[USR]: Where is it located?	[request.venueaddress]
[SYS]: It is located at 261 Driggs Avenue, Brooklyn.	[inform.venueaddress]
[USR]: Sounds good to me.	[select.]
[SYS]: Do you want to purchase tickets?	[offerintent.[buyeventtickets]]
[USR]: Yes, I want to buy tickets.	[affirmintent.]
[SYS]: How many tickets do you want to buy?	[request.numberoftickets]
[USR]: I need 1 ticket.	[inform.numberoftickets]
[SYS]: Please confirm: 1 ticket for Amber Run Brooklyn next Friday in New York.	[confirm.city, confirm.date, confirm.eventname, confirm.numberoftickets]
[USR]: No, I want to buy three tickets.	[inform.numberoftickets, negate]
[SYS]: Please confirm: 3 tickets for the event next Friday.	[confirm.date, confirm.numberoftickets]
[USR]: Yes, sounds good.	[affirm]
[SYS]: Your reservation was successfully made.	[notifysuccess]
[USR]: Thanks, a lot.	[thankyou]
[SYS]: Anything else I can help with?	[reqmore]
[USR]: No, thats all. Thanks.	[negate, thankyou]
[SYS]: Have a good day.	[goodbye]
Movie	
[USR]: I want to find a comedy movie. Search for movies now showing in Oakland.	[inform.genre, inform.location, informintent.[findmovies]]
[SYS]: Dr. Strangelove, Gloria Bell, and Little are playing. Do any of these interest you?	[offer.moviename]
[USR]: Lets try something else. I want a regular show. Something like a Biographical movie.	[inform.genre, inform.showtype, requestalts]
[SYS]: Heres 1. It is called Breakthrough.	[informcount.count, offer.moviename]
[USR]: Breakthrough is great.	[select.moviename]
[SYS]: Do you want to purchase tickets now?	[offerintent.[buymovietickets]]

Table 16 continued

The Schema Guided Dialogues Utterances	Label
[USR] : No, not right now. What time does it play? I also want to watch Regal Jack London next Friday.	[inform.showdate, inform.theatername, informintent.[gettimesformovie], negateintent.]
[SYS] : I have found 1 showtime. It is at 9:30 pm, playing at Regal Cinemas Jack London 9.	[informcount.count, offer.showtime, offer.theatername]
[USR] : What is the theater address?	[request.streetaddress]
[SYS] : The address is 100 Washington Street.	[inform.streetaddress]
[USR] : Great, thats all I need.	[goodbye, select.]
[SYS] : Have a great day	[goodbye]

Author Contributions Conceptualization: [Maarten De Raedt, Fréderic Godin, Chris Develder, Thomas Demeester]; Methodology: [Maarten De Raedt, Thomas Demeester]; Formal analysis and investigation: [Maarten De Raedt]; Data curation [Maarten De Raedt]; Writing - original draft preparation: [Maarten De Raedt]; Writing - review and editing: [Chris Develder, Thomas Demeester]; Funding acquisition: [Chris Develder, Thomas Demeester, Fréderic Godin]; Supervision: [Chris Develder, Thomas Demeester, Fréderic Godin].

Funding This work was funded by Flanders Innovation & Entrepreneurship (VLAIO), through Baekeland project-HBC.2019.2221 in collaboration with Sinch Chatlayer. This research also received funding from the Flemish Government under the “Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen” programme.

Availability of data and materials The original version of the datasets analysed during the current study are available: SimDial, SGD, CamRest676, and DSTC2. The adapted versions of these datasets, which were generated and analyzed during the current study, are available from the corresponding author on reasonable request, and will be made available upon acceptance while adhering to the previously discussed licenses.

Declarations

Competing interests The authors have no competing interests to declare that are relevant to the content of this article.

Ethical and informed consent for data used Due to the academic nature of the datasets used and their associated licenses, we are allowed to use them in this study. The SimDial dataset [27] is licensed under the Apache License 2.0, which permits modification and distribution with a copyright notice. The SGD dataset [5] is licensed under CC-BY-4.0, which allows modification and redistribution under the same license and with a copyright notice. CamRest676 [25, 26] has been widely adopted in previous studies and has no specific license. The DSTC2 [24] dataset is licensed under the GPL-3.0 License, which permits modification and redistribution under the conditions of the license and copyright notice, as well as stating changes and disclosing the source code.

References

1. Casanueva I, Temčinas T, Gerz D, Henderson M, Vulić I (2020) Efficient intent detection with dual sentence encoders. In: Proceedings of the 2nd workshop on natural language processing for conversational AI, pp 38–45. https://doi.org/10.18653/v1/2020.nlp4convai-1.5
2. Henderson M, Casanueva I, Mrkšić N, Su P-H, Wen T-H, Vulić I (2020) ConveRT: Efficient and accurate conversational representations from transformers. In: Findings of the Association for Computational Linguistics: EMNLP 2020, pp 2161–2174. https://doi.org/10.18653/v1/2020.findings-emnlp.196
3. Wu D, Ding L, Lu F, Xie J (2020) Slotrefine: A fast non-autoregressive model for joint intent detection and slot filling. In: Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP), pp 1932–1937. https://doi.org/10.18653/v1/2020.emnlp-main.152
4. Wu C-S, Madotto A, Hosseini-Asl E, Xiong C, Socher R, Fung P (2019) Transferable multi-domain state generator for task-oriented dialogue systems. In: Proceedings of the 57th annual meeting of the association for computational linguistics, pp 808–819. https://doi.org/10.18653/v1/P19-1078
5. Rastogi A, Zang X, Sunkara S, Gupta R, Khaitan P (2020) Towards scalable multi-domain conversational agents: The schema-guided dialogue dataset. In: Proceedings of the AAAI conference on artificial intelligence, vol 34, pp 8689–8696. https://doi.org/10.1609/aaai.v34i05.6394
6. Yang Y, Li Y, Quan X (2021) UBAR: Towards fully end-to-end task-oriented dialog system with gpt-2. In: Proceedings of the AAAI conference on artificial intelligence, vol 35, pp 14230–14238. https://doi.org/10.1609/aaai.v35i16.17674
7. Qin L, Che W, Li Y, Wen H, Liu T (2019) A stack-propagation framework with token-level intent detection for spoken language understanding. 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), pp 2078–2087. https://doi.org/10.18653/v1/D19-1214
8. Niu P, Chen Z, Song M (2019) A novel bi-directional interrelated model for joint intent detection and slot filling. In: Proceedings of the 57th annual meeting of the association for computational linguistics, pp 5467–5471. https://doi.org/10.18653/v1/P19-1544

9. Kim N, Hong S (2021) Automatic classification of citizen requests for transportation using deep learning: Case study from boston city. Inf Process Manag 58(1):102410. https://doi.org/10.1016/j.ipm.2020.102410
10. Polignano M, Lops P, de Gemmis M, Semeraro G (2023) HELENA: An intelligent digital assistant based on a lifelong health user model. Inf Process Manag 60(1):103124. https://doi.org/10.1016/j.ipm.2022.103124
11. Liu B, Wu Y, Zhang F, Liu Y, Wang Z, Li C, Zhang M, Ma S (2022) Query generation and buffer mechanism: Towards a better conversational agent for legal case retrieval. Inf Process Manag 59(5):103051. https://doi.org/10.1016/j.ipm.2022.103051
12. Li S, Xie R, Zhu Y, Zhuang F, Tang Z, Zhao WX, He Q (2022) Self-supervised learning for conversational recommendation. Inf Process Manag 59(6):103067. https://doi.org/10.1016/j.ipm.2022.103067
13. Chotimongkol A, Rudnicky A (2008) Acquiring domain-specific dialog information from task-oriented human-human interaction through an unsupervised learning. In: Proceedings of the 2008 conference on empirical methods in natural language processing, pp 955–964. https://doi.org/10.3115/1613715.1613839
14. Ritter A, Cherry C, Dolan WB (2010) Unsupervised modeling of twitter conversations. In: Human Language Technologies: The 2010 annual conference of the North American chapter of the association for computational linguistics, pp 172–180
15. Zhai K, Williams JD (2014) Discovering latent structure in task-oriented dialogues. In: Proceedings of the 52nd annual meeting of the association for computational linguistics (Vol 1: Long Papers), pp 36–46. https://doi.org/10.3115/v1/p14-1004
16. Shi W, Zhao T, Yu Z (2019) Unsupervised dialog structure learning. In: Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, vol 1 (Long and Short Papers), pp 1797–1807. https://doi.org/10.18653/v1/n19-1178
17. Qiu L, Zhao Y, Shi W, Liang Y, Shi F, Yuan T, Yu Z, Zhu S-c (2020) Structured attention for unsupervised dialogue structure induction. In: Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP), pp 1889–1899. https://doi.org/10.18653/v1/2020.emnlp-main.148
18. Gunasekara RC, Nahamoo D, Polymenakos LC, Ganhotra J, Fadnis KP (2017) Quantized-dialog language model for goal-oriented conversational systems. In: Dialog system technology challenges workshop, DSTC6
19. Gunasekara RC, Nahamoo D, Polymenakos LC, Ciaurri DE, Ganhotra J, Fadnis KP (2019) Quantized dialog - a general approach for conversational systems. Comput Speech Lang 54:17–30. https://doi.org/10.1016/j.csl.2018.06.003
20. Kiros R, Zhu Y, Salakhutdinov R, Zemel RS, Torralba A, Urtasun R, Fidler S (2015) Skip-thought vectors. In: Proceedings of the 28th international conference on neural information processing systems - volume 2. NIPS'15, pp 3294–3302
21. Conneau A, Kiela D, Schwenk H, Barrault L, Bordes A (2017) Supervised learning of universal sentence representations from natural language inference data. In: Proceedings of the 2017 conference on empirical methods in natural language processing, pp 670–680. https://doi.org/10.18653/v1/d17-1070
22. Reimers N, Gurevych I (2019) Sentence-BERT: Sentence embeddings using Siamese BERT-networks. 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), pp 3982–3992. https://doi.org/10.18653/v1/D19-1410
23. Mikolov T, Chen K, Corrado G, Dean J (2013) Efficient estimation of word representations in vector space. In: 1st international conference on learning representations, workshop track proceedings. arXiv:1301.3781
24. Henderson M, Thomson B, Williams JD (2014) The second dialog state tracking challenge. In: Proceedings of the 15th annual meeting of the special interest group on discourse and dialogue (SIGDIAL), pp 263–272. https://doi.org/10.3115/v1/w14-4337
25. Wen T-H, Gasic M, Mrkšić N, Barahona LMR, Su P-H, Ultes S, Vandyke D, Young S (2016) Conditional generation and snapshot learning in neural dialogue systems. In: Proceedings of the 2016 conference on empirical methods in natural language processing, pp 2153–2162. https://doi.org/10.18653/v1/d16-1233
26. Wen T-H, Vandyke D, Mrkšić N, Gasic M, Barahona LMR, Su P-H, Ultes S, Young S (2017) A network-based end-to-end trainable task-oriented dialogue system. In: Proceedings of the 15th conference of the European chapter of the association for computational linguistics: vol 1, long papers, pp 438–449. https://doi.org/10.18653/v1/e17-1042
27. Zhao T, Eskenazi M (2018) Zero-shot dialog generation with cross-domain latent actions. In: Proceedings of the 19th Annual SIGdial meeting on discourse and dialogue, pp 1–10. https://doi.org/10.18653/v1/w18-5001
28. Wu C-S, Hoi SC, Socher R, Xiong C (2020) TOD-BERT: Pre-trained natural language understanding for task-oriented dialogue. In: Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP), pp 917–929. https://doi.org/10.18653/v1/2020.emnlp-main.66
29. Jurafsky D (1997) Switchboard swbd-damsl shallow-discourse-function annotation coders manual. Institute of Cognitive Science Technical Report
30. Crook N, Granell R, Pulman S (2009) Unsupervised classification of dialogue acts using a dirichlet process mixture model. In: Proceedings of the SIGDIAL 2009 conference: The 10th annual meeting of the special interest group on discourse and dialogue. SIGDIAL '09, pp 341–348. https://doi.org/10.3115/1708376.1708427
31. Joty S, Carenini G, Lin C-Y (2011) Unsupervised modeling of dialog acts in asynchronous conversations. In: Proceedings of the twenty-second international joint conference on artificial intelligence - volume three. IJCAI'11, pp 1807–1813
32. Brychcín T, Král P (2017) Unsupervised dialogue act induction using Gaussian mixtures. In: Proceedings of the 15th conference of the European chapter of the association for computational linguistics: volume 2, short papers, pp 485–490. https://aclanthology.org/E17-2078
33. Chung J, Kastner K, Dinh L, Goel K, Courville A, Bengio Y (2015) A recurrent latent variable model for sequential data. In: Proceedings of the 28th international conference on neural information processing systems - volume 2. NIPS'15, pp 2980–2988
34. Kim Y, Denton C, Hoang L, Rush AM (2017) Structured attention networks. In: 5th International conference on learning representations, conference track proceedings
35. Hudeček V, Dušek O (2022) Learning interpretable latent dialogue actions with less supervision. In: Proceedings of the 2nd conference of the Asia-Pacific chapter of the association for computational linguistics and the 12th international joint conference on natural language processing (volume 1: long papers), pp 297–308. https://aclanthology.org/2022.aacl-main.24
36. Xu J, Lei Z, Wang H, Niu Z-Y, Wu H, Che W (2021) Discovering dialog structure graph for coherent dialog generation. In: Proceedings of the 59th annual meeting of the association for computational linguistics and the 11th international joint conference on natural language processing (vol 1: long papers), pp 1726–1739. https://doi.org/10.18653/v1/2021.acl-long.136
37. Sun Y, Shan Y, Tang C, Hu Y, Dai Y, Yu J, Sun J, Huang F, Si L (2021) Unsupervised learning of deterministic dialogue structure with edge-enhanced graph auto-encoder. In: Proceedings of the AAAI conference on artificial intelligence, vol 35, pp 13869–13877. https://doi.org/10.1609/aaai.v35i15.17634

38. Subramanian S, Mudumba SR, Sordoni A, Trischler A, Courville AC, Pal C (2018) Towards text generation with adversarially learned neural outlines. Advances in Neural Information Processing Systems 31
39. Lucas J, Tucker G, Grosse RB, Norouzi M (2019) Understanding posterior collapse in generative latent variable models. In: Deep generative models for highly structured data, ICLR workshop. https://openreview.net/forum?id=r1xaVLUYuE
40. Wang Y, Blei D, Cunningham JP (2021) Posterior collapse and latent variable non-identifiability. In: Advances in neural information processing systems vol 34, pp 5443–5455
41. Hudeček V, Dušek O, Yu Z (2021) Discovering dialogue slots with weak supervision. In: Proceedings of the 59th annual meeting of the association for computational linguistics and the 11th international joint conference on natural language processing (vol 1: long papers), pp 2430–2442. https://doi.org/10.18653/v1/2021.acl-long.189
42. Qiu L, Wu C-S, Liu W, Xiong C (2022) Structure extraction in task-oriented dialogues with slot clustering. Preprint arXiv:2203.00073
43. Vukovic R, Heck M, Ruppik B, van Niekerk C, Zibrowius M, Gasic M (2022) Dialogue term extraction using transfer learning and topological data analysis. In: Proceedings of the 23rd annual meeting of the special interest group on discourse and dialogue, pp 564–581. https://aclanthology.org/2022.sigdial-1.53
44. Yu D, Wang M, Cao Y, Shafran I, Shafey L, Soltau H (2022) Unsupervised slot schema induction for task-oriented dialog. In: Proceedings of the 2022 conference of the North American chapter of the association for computational linguistics: human language technologies, pp 1174–1193. https://doi.org/10.18653/v1/2022.naacl-main.86
45. Zhao T, Xie K, Eskenazi M (2019) Rethinking action spaces for reinforcement learning in end-to-end dialog agents with latent variable models. In: Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, vol 1 (long and short papers), pp 1208–1218. https://doi.org/10.18653/v1/n19-1123
46. Shi W, Qian K, Wang X, Yu Z (2019) How to build user simulators to train RL-based dialog systems. 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), pp 1990–2000. https://doi.org/10.18653/v1/D19-1206
47. Xu J, Wang H, Niu Z-Y, Wu H, Che W, Liu T (2020) Conversational graph grounded policy learning for open-domain conversation generation. In: Proceedings of the 58th annual meeting of the association for computational linguistics, pp 1835–1845. https://doi.org/10.18653/v1/2020.acl-main.166
48. Raghu D, Agarwal S, Joshi S, Mausam (2021) End-to-end learning of flowchart grounded task-oriented dialogs. In: Proceedings of the 2021 conference on empirical methods in natural language processing, pp 4348–4366. https://doi.org/10.18653/v1/2021.emnlp-main.357
49. Hu W, Chan Z, Liu B, Zhao D, Ma J, Yan R (2019) Gsn: A graph-structured network for multi-party dialogues. Preprint arXiv:1905.13637
50. Mehri S, Eskenazi M (2021) Schema-guided paradigm for zero-shot dialog. In: Proceedings of the 22nd annual meeting of the special interest group on discourse and dialogue, pp 499–508. https://doi.org/10.18653/v1/w18-5001
51. Zang X, Rastogi A, Sunkara S, Gupta R, Zhang J, Chen J (2020) MultiWOZ 2.2 : A dialogue dataset with additional annotation corrections and state tracking baselines. In: Wen T-H, Celikyilmaz A, Yu Z, Papangelis A, Eric M, Kumar A, Casanueva I, Shah R (eds.) Proceedings of the 2nd workshop on natural language processing for conversational AI, pp 109–117. Association for Computational Linguistics, Online. https://doi.org/10.18653/v1/2020.nlp4convai-1.13. https://aclanthology.org/2020.nlp4convai-1.13
52. Pennington J, Socher R, Manning C (2014) GloVe: Global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pp 1532–1543. https://doi.org/10.3115/v1/D14-1162
53. Devlin J, Chang M-W, Lee K, Toutanova K (2019) BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, vol 1 (long and short papers), pp 4171–4186. https://doi.org/10.18653/v1/N19-1423
54. Zhang Y, Sun S, Galley M, Chen Y-C, Brockett C, Gao X, Gao J, Liu J, Dolan B (2020) DIALOGPT : Large-scale generative pre-training for conversational response generation. In: Proceedings of the 58th annual meeting of the association for computational linguistics: system demonstrations, pp 270–278. https://doi.org/10.18653/v1/2020.acl-demos.30
55. Bao S, He H, Wang F, Wu H, Wang H (2020) PLATO: Pre-trained dialogue generation model with discrete latent variable. In: Proceedings of the 58th annual meeting of the association for computational linguistics, pp 85–96. https://doi.org/10.18653/v1/2020.acl-main.9
56. Kelley JF (1984) An iterative design methodology for user-friendly natural language office information applications. ACM Trans Inf Syst (TOIS) 2(1):26–41. https://doi.org/10.1145/357417.357420
57. Steinley D (2004) Properties of the hubert-arable adjusted rand index. Psychol Methods 9(3):386. https://doi.org/10.1037/1082-989x.9.3.386
58. Vinh NX, Epps J, Bailey J (2010) Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. J Mach Learn Res 11:2837–2854. https://doi.org/10.1145/1553374.1553511
59. Fowlkes EB, Mallows CL (1983) A method for comparing two hierarchical clusterings. J Am Stat Assoc 78(383):553–569. https://doi.org/10.1080/01621459.1983.10478008
60. Hubert L, Arabie P (1985) Comparing partitions. J Classif 2(1):193–218. https://doi.org/10.1007/bf01908075
61. Lloyd S (1982) Least squares quantization in pcm. IEEE Trans Inf Theory 28(2):129–137. https://doi.org/10.1109/tit.1982.1056489
62. Vassilvitskii S, Arthur D (2006) k-means++: The advantages of careful seeding. In: Proceedings of the eighteenth annual ACM-SIAM symposium on discrete algorithms, pp 1027–1035
63. Kanungo T, Mount DM, Netanyahu NS, Piatko CD, Silverman R, Wu AY (2002) An efficient k-means clustering algorithm: Analysis and implementation. IEEE Trans Pattern Anal Mach Intell 24(7):881–892. https://doi.org/10.1109/tpami.2002.1017616
64. Ester M, Kriegel H-P, Sander J, Xu X (1996) A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proceedings of the second international conference on knowledge discovery and data mining, pp 226–231
65. Comaniciu D, Meer P (2002) Mean shift: A robust approach toward feature space analysis. IEEE Trans Pattern Anal Mach Intell 24(5):603–619
66. Frey BJ, Dueck D (2007) Clustering by passing messages between data points. Science 315(5814):972–976. https://doi.org/10.1126/science.1136800
67. Cai X, Nie F, Huang H (2013) Multi-view k-means clustering on big data. In: Proceedings of the twenty-third international joint conference on artificial intelligence. IJCAI ’13, pp 2598–2604

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.