Dialog Intent Induction with Deep Multi-View Clustering

Hugh Perkins and **Yi Yang**
ASAPP Inc.
New York, NY 10007
{hp+yyang}@asapp.com

Abstract

We introduce the dialog intent induction task and present a novel deep multi-view clustering approach to tackle the problem. Dialog intent induction aims at discovering user intents from user query utterances in human-human conversations such as dialogs between customer support agents and customers.[1] Motivated by the intuition that a dialog intent is not only expressed in the user query utterance but also captured in the rest of the dialog, we split a conversation into two independent views and exploit multi-view clustering techniques for inducing the dialog intent. In particular, we propose alternating-view k-means (AV-KMEANS) for joint multi-view representation learning and clustering analysis. The key innovation is that the instance-view representations are updated iteratively by predicting the cluster assignment obtained from the alternative view, so that the multi-view representations of the instances lead to similar cluster assignments. Experiments on two public datasets show that AV-KMEANS can induce better dialog intent clusters than state-of-the-art unsupervised representation learning methods and standard multi-view clustering approaches.[2]

Customer 1: *A wireless charging case is fancy and all but can we get a "find my airpod" feature going?*
Agent 1: *If you have lost your AirPods, Find My iPhone can help you locate them.*

Customer 2: *hey man I lost and miss my airpods plz help me!*
Agent 2: *Hi there! With iOS 10.3 or later, Find My iPhone can help you locate missing AirPods.*

Figure 1: Two dialogs with the `FindAirPods` user intent. The user query utterances of the two dialogs are lexically and syntactically dissimilar, while the rests of the dialogs are similar.

1 Introduction

Goal-oriented dialog systems assist users to accomplish well-defined tasks with clear intents within a limited number of dialog turns. They have been adopted in a wide range of applications, including booking flights and restaurants (Hemphill et al., 1990; Williams, 2012), providing tourist information (Kim et al., 2016), aiding in the customer support domain, and powering intelligent virtual assistants such as Apple Siri, Amazon Alexa, or Google Assistant. The first step towards building such systems is to determine the target tasks and construct corresponding ontologies to define the constrained set of dialog states and actions (Henderson et al., 2014b; Mrkšić et al., 2015).

Existing work assumes the target tasks are given and excludes dialog intent discovery from the dialog system design pipeline. Because of this, most of the works focus on few simple dialog intents and fail to explore the realistic complexity of user intent space (Williams et al., 2013; Budzianowski et al., 2018). The assumption puts a great limitation on adapting goal-oriented dialog systems to important but complex domains like customer support and healthcare where having a complete view of user intents is impossible. For example, as shown in Fig. 1, it is non-trivial to predict user intents for troubleshooting a newly released product in advance. To address this problem, we propose to employ data-driven approaches to automatically discover user intents in dialogs from human-human conversations. Follow-up analysis can then be performed to identify the most valuable dialog intents and design dialog systems to automate the conversations accordingly.

Similar to previous work on user question/query intent induction (Sadikov et al., 2010; Haponchyk

[1] We focus on inducing abstract intents like `BookFlight` and ignore detailed arguments such as *departure date* and *destination*.

[2] When ready, the data and code will be published at `https://github.com/asappresearch/dialog-intent-induction`.

*Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing
and the 9th International Joint Conference on Natural Language Processing*, pages 4016–4025,
Hong Kong, China, November 3–7, 2019. ©2019 Association for Computational Linguistics

et al., 2018), we can induce dialog intents by clustering user query utterances[3] in human-human conversations. The key is to learn discriminative query utterance representations in the user intent semantic space. Unsupervised learning of such representations is challenging due to the semantic shift across different domains (Nida, 2015). We propose to overcome this difficulty by leveraging the rest of a conversation in addition to the user query utterance as a weak supervision signal. Consider the two dialogs presented in Fig. 1 where both of the users are looking for how to find their AirPods. Although the user query utterances vary in the choice of lexical items and syntactic structures, the human agents follow the same workflow to assist the users, resulting in similar conversation structures.[4]

We present a deep multi-view clustering approach, alternating-view k-means (AV-KMEANS), to leverage the weak supervision for the semantic clustering problem. In this respect, we partition a dialog into two independent views: the user query utterance and the rest of the conversation. AV-KMEANS uses different neural encoders to embed the inputs corresponding to the two views and to encourage the representations learned by the encoders to yield similar cluster assignments. Specifically, we alternatingly perform k-means-style updates to compute the cluster assignment on one view and then train the encoder of the other view by predicting the assignment using a metric learning algorithm (Snell et al., 2017). Our method diverges from previous work on multi-view clustering (Bickel and Scheffer, 2004; Chaudhuri et al., 2009; Kumar et al., 2011), as it is able to learn robust representations via neural networks that are in clustering-analysis-friendly geometric spaces. Experimental results on a dialog intent induction dataset and a question intent clustering dataset show that AV-KMEANS significantly outperforms multi-view clustering algorithms without joint representation learning by 6–20% absolute F1 scores. It also gives rise to better F1 scores than quick thoughts (Logeswaran and Lee, 2018), a state-of-the-art unsupervised representation learning method.

Our contributions are summarized as follows:

- We introduce the dialog intent induction task and present a multi-view clustering formulation to solve the problem.
- We propose a novel deep multi-view clustering approach that jointly learns cluster-discriminative representations and cluster assignments.
- We derive and annotate a dialog intent induction dataset obtained from a public Twitter corpus and process a duplicate question detection dataset into a question intent clustering dataset.
- The presented algorithm, AV-KMEANS, significantly outperforms previous state-of-the-art multi-view clustering algorithms as well as two unsupervised representation learning methods on the two datasets.

2 Deep Multi-View Clustering

In this section, we present a novel method for joint multi-view representation learning and clustering analysis. We consider the case of two independent views, in which the first view corresponds to the user query utterance (query view) and the second one corresponds to the rest of the conversation (content view).

Formally, given a set of n instances $\{x_i\}$, we assume that each data point x_i can be naturally partitioned into two independent views $x_i^{(1)}$ and $x_i^{(2)}$. We further use two neural network encoders f_{ϕ_1} and f_{ϕ_2} to transform the two views into vector representations $\mathbf{x}_i^{(1)}, \mathbf{x}_i^{(2)} \in \mathbb{R}^D$. We are interested in grouping the data points into K clusters using the multi-view feature representations. In particular, the neural encoders corresponding to the two views are jointly optimized so that they would commit to similar cluster assignments for the same instances.

In this work, we implement the query-view encoder f_{ϕ_1} with a bi-directional LSTM (BiLSTM) network (Hochreiter and Schmidhuber, 1997) and the content-view encoder f_{ϕ_2} with a hierarchical BiLSTM model that consists of a utterance-level BiLSTM encoder and a content-level BiLSTM encoder. The concatenations of the hidden representations from the last time steps are adopted as the query or content embeddings.

[3] We treat the initial user utterances of the dialogs as user query utterances.

[4] Note this is not always the case. For the same dialog intent, the agent treatments may differ depending on the user profiles. The user may also change intent in the middle of a conversation. Thus, the supervision is often very noisy.

2.1 Alternating-view k-means clustering

In this work, we propose alternating-view k-means (AV-KMEANS) clustering, a novel method for deep multi-view clustering that iteratively updates neural encoders corresponding to the two views by encouraging them to yield similar cluster assignments for the same instances. In each semi-iteration, we perform k-means-style updates to compute a cluster assignment and centroids on feature representations corresponding to one view, and then project the cluster assignment to the other view where the assignment is used to train the view encoder in a supervised learning fashion.

Algorithm 1: alternating-view k-means

```
Input     : two-view inputs $\{(x_i^{(1)}, x_i^{(2)})\}$;
            numbers of iterations $T$, $M$;
            number of clusters $K$
Output    : final cluster assignment $\{z_i^{(1)}\}$
Parameter: encoders $f_{\phi_1}$ and $f_{\phi_2}$
Initialize $f_{\phi_1}$ and $f_{\phi_2}$ (§ 2.3)
$\{z_i^{(1)}\} \leftarrow$ K-MEANS$(\{f_{\phi_1}(x_i^{(1)})\}, K)$
for $t = 1, \cdots, T$ do
    // project cluster assignment from
       view 1 to view 2
    Update $f_{\phi_2}$ with pseudo training instances
      $\{(x_i^{(2)}, z_i^{(1)})\}$ (§ 2.2)
    Encode view-2 inputs: $\{\mathbf{x}_i^{(2)} \leftarrow f_{\phi_2}(x_i^{(2)})\}$
    $\{z_i^{(2)}\} \leftarrow$ K-MEANS$(\{\mathbf{x}_i^{(2)}\}, K, M, \{z_i^{(1)}\})$

    // project cluster assignment from
       view 2 to view 1
    Update $f_{\phi_1}$ with pseudo training instances
      $\{(x_i^{(1)}, z_i^{(2)})\}$ (§ 2.2)
    Encode view-1 inputs: $\{\mathbf{x}_i^{(1)} \leftarrow f_{\phi_1}(x_i^{(1)})\}$
    $\{z_i^{(1)}\} \leftarrow$ K-MEANS$(\{\mathbf{x}_i^{(1)}\}, K, M, \{z_i^{(2)}\})$
end
```

The full training algorithm is presented in Alg. 1, where K-MEANS$(\{\mathbf{x}_i\}, K, M, \{z'_i\})$ is a function that runs k-means clustering on inputs $\{\mathbf{x}_i\}$. K is the number of clusters. M and $\{z'_i\}$ are optional arguments that represent the number of k-means iterations and the initial cluster assignment. The function returns cluster assignment $\{z_i\}$. A visual demonstration of one semi-iteration of AV-KMEANS is also available in Fig. 2.

In particular, we initialize the encoders randomly or by using pretrained encoders (§ 2.3). Then, we can obtain the initial cluster assignment by performing k-means clustering on vector representations encoded by f_{ϕ_1}. During each AV-KMEANS iteration, we first project cluster assignment from view 1 to view 2 and update the neural encoder for view 2 by formulating a supervised learning problem (§ 2.2). Then we perform M vanilla k-means steps to adjust the cluster assignment in view 2 based on the updated encoder. We repeat the procedure for view 2 in the same iteration. Note that in each semi-iteration, the initial centroids corresponding to a view are calculated based on the cluster assignment obtained from the other view. The algorithm runs a total number of T iterations.

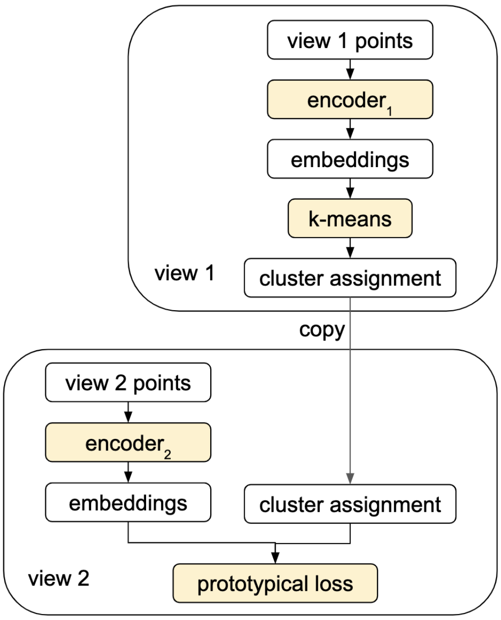

Figure 2: A depiction of a semi-iteration of the alternating-view k-means algorithm. k-means clustering and prototypical classification are performed for view 1 and view 2 respectively. The view 1 encoder is frozen and the view 2 encoder is updated in this semi-iteration.

2.2 Prototypical episode training

In each AV-KMEANS iteration, we need to solve two supervised classification problems using the pseudo training datasets $\{(x_i^{(2)}, z_i^{(1)})\}$ and $\{(x_i^{(1)}, z_i^{(2)})\}$ respectively. A simple way to do so is putting a softmax classification layer on top of each encoder network. However, we find that it is beneficial to directly perform classification in the k-means clustering space. To this end, we adopt prototypical networks (Snell et al., 2017), a metric learning approach, to solely rely on the encoders to form the classifiers instead of introducing additional classification layers.

Given input data $\{(x_i, z_i)\}$ and a neural network encoder f_ϕ, prototypical networks compute a D-dimensional representation $\mathbf{c}_k$, or prototype, of each class by averaging the vectors of the embedded support points belonging to its class:

$$\mathbf{c}_k = \frac{1}{|S_k|} \sum_{(x_i, z_i) \in S_k} f_\phi(x_i) \qquad (1)$$

here we drop the view superscripts for simplicity. Conceptually, the prototypes $\{\mathbf{c}_k\}$ are similar to the centroids in the k-means algorithm, except that a prototype is computed on a subset of the instances of a class (the support set) while a centroid is computed based on all instances of a class.

Given a sampled query data point x, prototypical networks produce a distribution over classes based on a softmax over distances to the prototypes in the embedding space:

$$p(y = k|x) = \frac{\exp(-d(f_\phi(x), \mathbf{c}_k))}{\sum_{k'} \exp(-d(f_\phi(x), \mathbf{c}_{k'}))}, \qquad (2)$$

where the distance function is the squared Euclidean distance $d(\mathbf{x}, \mathbf{x}') = ||\mathbf{x} - \mathbf{x}'||^2$.

The model minimizes the negative log-likelihood of the data: $L(\phi) = -\log p(y = k|x)$. Training episodes are formed by randomly selecting a subset of classes from the training set, then choosing a subset of examples within each class to act as the support set and a subset of the remainder to serve as query points. We refer to the original paper (Snell et al., 2017) for more detailed description of the model.

2.3 Parameter initialization

Although AV-KMEANS can effectively work with random parameter initializations, we do expect that it will benefit from initializations obtained from pretrained models with some well-studied unsupervised learning objectives. We present two methods to initialize the utterance encoders for both the query and content views. The first approach is based on recurrent autoencoders. We embed an utterance using a BiLSTM encoder. The utterance embedding is then concatenated with every word vector corresponding to the decoder inputs that are fed into a uni-directional LSTM decoder. We use the neural encoder trained with the autoencoding objective to initialize the two utterance encoders in AV-KMEANS.

Recurrent autoencoders independently reconstruct an input utterance without capturing semantic dependencies across consecutive utterances. We consider a second initialization method, quick thoughts (Logeswaran and Lee, 2018), that addresses the problem by predicting a context utterance from a set of candidates given a target utterance. Here, the target utterances are sampled randomly from the corpus, and the context utterances are sampled from within each pair of adjacent utterances. We use two separate BiLSTM encoders to encode utterances, which are named as the target encoder f and the context encoder g. To score the compatibility of a target utterance $\mathbf{s}$ and a candidate context utterance $\mathbf{t}$, we simply use the inner product of the two utterance vectors $f(\mathbf{s})^\top \cdot g(\mathbf{t})$. The training objective maximizes the log-likelihood of the context utterance given the target utterance and the candidate utterance set. After pretraining, we adopt the target encoder to initialize the two utterance encoders in AV-KMEANS.

3 Data

As discussed in the introduction, existing goal-oriented dialog datasets mostly concern predefined dialog intents in some narrow domains such as restaurant or travel booking (Henderson et al., 2014a; Budzianowski et al., 2018; Serban et al., 2018). To carry out this study, we adopt a more challenging corpus that consists of human-human conversations for customer service and manually annotate the user intents of a small number of dialogs. We also build a question intent clustering dataset to assess the generalization ability of the proposed method on the related problem.

3.1 Twitter airline customer support

We consider the customer support on Twitter corpus released by Kaggle,[5] which contains more than three million tweets and replies in the customer support domain. The tweets constitute conversations between customer support agents of some big companies and their customers. As the conversations regard a variety of dynamic topics, they serve as an ideal testbed for the dialog intent induction task. In the customer service domain, different industries generally address unrelated topics and concerns. We focus on dialogs in the airline industry,[6] as they represent the largest number of conversations in the corpus. We name

[5] https://www.kaggle.com/thoughtvector/customer-support-on-twitter

[6] We combined conversations involved the following Twitter handles: @Delta, @British_Airways, @SouthwestAir, and @AmericanAir.

Dialog intent	# Dialogs	Query utterance example
`Baggage`	40	hi, do suit bags count as a personal items besides carry on baggage?
`BookFlight`	27	trying all day to book an international flight, only getting error msg.
`ChangeFlight`	16	can i request to change my flight from lax to msy on 10/15?
`CheckIn`	21	hy how can i have some help... having some problems with a check in
`CustomerService`	19	2 hour wait time to talk to a customer service agent?!?
`FlightDelay`	85	delay... detroit < orlando
`FlightEntertainment`	40	@airline is killing it with these inflight movie options
`FlightFacility`	32	just flew @airline economy... best main cabin seat ive ever sat in.
`FlightStaff`	30	great crew on las vegas to baltimore tonight.
`Other`	116	hi, i have a small question!
`RequestFeature`	10	when are you going to update your app for iphone x?
`Reward`	17	need to extend travel funds that expire tomorrow!
`TerminalFacility`	13	thx for the new digital signs at dallas lovefield. well done!!
`TerminalOperation`	34	would be nice if you actually announced delays

Table 1: Statistics of the labeled Twitter airline customer support (TwACS) dataset and the corresponding user query utterance examples. The Twitter handles of the airlines are replaced by @airline.

the resulting dataset the *Twitter airline customer support (TwACS)* corpus. We rejected any conversation that redirects the customer to a URL or another communication channel, e.g., direct messages. We ended up with a dataset of $43,072$ dialogs. The total numbers of dialog turns and tokens are $63,147$ and $2,717,295$ respectively.

After investigating 500 randomly sampled conversations from TwACS, we established an annotation task with 14 dialog intents and hired two annotators to label the sampled dialogs based on the user query utterances. The Cohen's kappa coefficient was 0.75, indicating a substantial agreement between the annotators. The disagreed items were resolved by a third annotator. To our knowledge, this is the first dialog intent induction dataset. The data statistics and user query utterance examples corresponding to different dialog intents are presented in Table 1.

3.2 AskUbuntu

AskUbuntu is a dataset collected and processed by Shah et al. (2018) for the duplicate question detection task. The dataset consists of technical support questions posted by users on AskUbuntu website with annotations indicating that two questions are semantically equivalent. For instance,

q_1 : *how to install ubuntu w/o removing windows*
q_2 : *installing ubuntu over windows 8.1*

are duplicate and they can be resolved with similar answers. A total number of $257,173$ questions are included in the dataset and $27,289$ pairs of questions are labeled as duplicate ones. In addition, we obtain the top rated answer for each question from the AskUbuntu website dump.[7]

In this work, we reprocess the data and build a question intent clustering dataset using an automatic procedure. Following Haponchyk et al. (2018), we transform the duplicate question annotations into the question intent cluster annotations with a simple heuristic: for each question pair q_1, q_2 annotated as a duplicate, we assigned q_1 and q_2 to the same cluster. As a result, the question intent clusters correspond to the connected components in the duplicate question graph. There are $7,654$ such connected components. However, most of the clusters are very small: 91.7% of the clusters contain only 2–5 questions. Therefore, we experiment with the largest 20 clusters that contain $4,692$ questions in this work. The sizes of the largest and the smallest clusters considered in this study are $1,364$ and 71 respectively.

4 Experiments

In this section, we evaluate AV-KMEANS on the TwACS and AskUbuntu datasets as described in § 3. We compare AV-KMEANS with competitive systems for representation learning or multi-view clustering and present our main findings in § 4.2. In addition, we examine the output clusters obtained from AV-KMEANS on the TwACS dataset to perform a thoughtful error analysis.

4.1 Experimental settings

We train the models on all the instances of a dataset and evaluate on the labeled instances. We employ the publicly available 300-dimensional

[7] `https://archive.org/details/stackexchange`

Clustering algorithm	Pretraining method	TwACS				AskUbuntu			
		Prec	Rec	F1	ACC	Prec	Rec	F1	ACC
Baseline systems									
k-means	PCA	28.1	28.3	28.2	19.8	35.1	27.8	31.0	22.0
	autoencoders	34.4	25.9	29.5	23.2	27.3	20.1	23.1	14.6
	quick thoughts	46.7	38.3	42.1	35.4	42.9	39.4	41.1	33.1
MVSC	PCA	32.4	24.2	27.8	22.6	40.4	27.7	32.9	25.5
	autoencoders	36.1	27.7	31.3	24.9	36.7	22.8	28.2	21.4
	quick thoughts	45.8	35.4	40.0	32.6	35.1	23.4	28.1	22.3
Our approach									
Av-Kmeans	no pretraining	37.5	33.6	35.4	29.5	52.0	51.9	51.9	**44.0**
	autoencoders	44.4	34.6	38.9	31.6	50.6	46.1	48.2	39.7
	quick thoughts	**48.9**	**43.8**	**46.2**	**39.9**	**53.8**	**52.7**	**53.3**	41.1

Table 2: Evaluation results on the TwACS and AskUbuntu datasets for different systems. MVSC is short for the multi-view spectral clustering algorithm proposed by Kanaan-Izquierdo et al. (2018). The pretrained representations are fixed during k-means and MVSC clustering and they are fine-tuned during Av-Kmeans clustering. The best results are in **bold**.

GloVe vectors (Pennington et al., 2014) pretrained with 840 billion tokens to initialize the word embeddings for all the models.

Competitive systems We consider state-of-the-art methods for representation learning and/or multi-view clustering as our baseline systems. We formulate the dialog induction task as an unsupervised clustering task and include two popular clustering algorithms *k-means* and *spectral clustering*. *multi-view spectral clustering (MVSC)* (Kanaan-Izquierdo et al., 2018) is a competitive standard multi-view clustering approach.[8] In particular, we carry out clustering using the query-view and content-view representations learned by the representation learning methods (k-means only requires query-view representations). In the case where a content-view input corresponds to multiple utterances, we take the average of the utterance vectors as the content-view output representation for autoencoders and quick thoughts.

Av-Kmeans is a joint representation learning and multiview clustering method. Therefore, we compare with SOTA representation learning methods *autoencoders*, and *quick thoughts* (Logeswaran and Lee, 2018). Quick thoughts is a strong representation learning baseline that is adopted in BERT (Devlin et al., 2019). We also include *principal component analysis (PCA)*, a classic representation learning and dimensionality reduction method, since bag-of-words representations are too expensive to work with for clustering analysis.

We compare three variants of Av-Kmeans that differ in the pretraining strategies. In addition to the Av-Kmeans systems pretrained with autoencoders and quick thoughts, we also consider a system whose encoder parameters are randomly initialized (*no pretraining*).

Metrics We compare the competitive approaches on a number of standard evaluation measures for clustering analysis. Following prior work (Kumar et al., 2011; Haponchyk et al., 2018; Xie et al., 2016), we set the number of clusters to the number of ground truth categories and report precision, recall, F1 score, and unsupervised clustering accuracy (ACC). To compute precision or recall, we assign each predicted cluster to the most frequent gold cluster or assign each gold cluster to the most frequent predicted cluster respectively. The F1 score is the harmonic average of the precision and recall. ACC uses a one-to-one assignment between the gold standard clusters and the predicted clusters. The assignment can be efficiently computed by the Hungarian algorithm (Kuhn, 1955).

Parameter tuning We empirically set both the dimension of the LSTM hidden state and the number of principal components in PCA to 300. The number of Av-Kmeans iterations T and the number of k-means steps in a Av-Kmeans semi-iteration M are set to 50 and 10 respectively, as we find that more iterations lead to similar cluster assignments. We adopt the same set of hyperparameter values as used by Snell et al. (2017)

[8] We use the scikit-learn k-means implementation and the MVSC implementation available at: `https://pypi.org/project/multiview/`.

for training the prototypical networks. Specifically, we fix the number of query examples and the number of support examples to 15 and 5. The networks are trained for 100 episodes per AV-KMEANS semi-iteration. The number of sampled classes per episode is chosen to be 10, as it has to be smaller than the number of ground truth clusters. Adam (Kingma and Ba, 2015) is utilized to optimize the models and the initial learning rate is 0.001. During autoencoders or quick thoughts pretraining, we check the performance on the development set after each epoch to perform early stopping, where we randomly sample 10% unlabeled instances as the development data.

4.2 Results

Our main empirical findings are presented in Table 2, in which we compare AV-KMEANS with standard single-view and multi-view clustering algorithms. We also evaluate classic and neural approaches for representation learning, where the pretrained representations are fixed during k-means and MVSC clustering and they are fine-tuned during AV-KMEANS clustering. We analyze the empirical results in details in the following paragraphs.

Utilizing multi-view information Among all the systems, k-means clustering on representations trained with PCA or autoencoders only employs single-view information encoded in user query utterances. They clearly underperform the rest of the systems that leverage multi-view information of the entire conversations. Quick thoughts infuses the multi-view knowledge through the learning of the query-view vectors that are aware of the content-view semantics. In contrast, multi-view spectral clustering can work with representations that are separately learned for the individual views and the multi-view information is aggregated using the common eigenvectors of the data similarity Laplacian matrices. As shown, k-means clustering on quick thoughts vectors gives superior results than MVSC pretrained with PCA or autoencoders by more than 10% F1 or ACC, which indicates that multi-view representation learning is effective for problems beyond simple supervised learning tasks. Combining representation learning and multi-view clustering in a static way seems to be less ideal—MVSC performs worse than k-means using the quick thoughts vectors as clustering inputs. Multi-view representation learning breaks the independent-view assumption that is critical for classic multi-view clustering algorithms.

Joint representation learning and clustering We now investigate whether joint representation learning and clustering can reconcile the conflict between cross-view representation learning and classic multi-view clustering. AV-KMEANS outperforms k-means and MVSC baselines by considerable margins. It achieves 46% and 53% F1 scores and 40% and 44% ACC scores on the TwACS and AskUbuntu datasets, which are 5–30 percent higher than competitive systems. Compared to alternative methods, AV-KMEANS is able to effectively seek clustering-friendly representations that also encourage similar cluster assignments for different views of the same instances. With the help of quick thoughts pretraining, AV-KMEANS improves upon the strongest baseline, k-means clustering on quick thoughts vectors, by 4.5% ACC on the TwACS dataset and 12.2% F1 on the AskUbuntu dataset.

Model pretraining for AV-KMEANS Evaluation results on AV-KMEANS with different parameter initialization strategies are available in Table 2. As suggested, pretraining neural encoders is important for obtaining competitive results on the TwACS dataset, while its impact on the AskUbuntu dataset is less pronounced. AskUbuntu is six times larger than TwACS and models trained on AskUbuntu are less sensitive to their parameter initializations. This observation is consistent with early research on unsupervised pretraining, where Schmidhuber et al. (2012) argue that unsupervised initialization/pretraining is not necessary if a large amount of training data is available. Between the two pretraining methods, quick thoughts is much more effective than autoencoders—it improves upon no pretraining and autoencoders by 10.4% and 8.3% ACC scores on the TwACS dataset.

4.3 Error analysis

Our best performed system still fails to hit 50% F1 or ACC score on the TwACS dataset. We examine the outputs of the quick thoughts pretrained AV-KMEANS on TwACS, focusing on investigating the most frequent errors made by the system. To this end, we compute the confusion matrix based on the one-to-one assignment between the gold clusters and the predicted clusters used by ACC.

Ground truth	Prediction	# Instances
Other	CustomerService	21
TerminalOp.	FlightDelay	10
FlightDelay	ChangeFlight	10
Other	FlightStaff	10
FlightEnter.	Other	8

Table 3: The top 5 most frequent errors made by the quick thoughts pretrained Av-Kmeans on the TwACS dataset. The one-to-one assignment between the gold clusters and the predicted clusters is computed by the Hungarian algorithm.

The top 5 most frequent errors are presented in Table 3. As shown, three of the five errors involve Other. Instances under the Other category correspond to miscellaneous dialog intents, thereby they are less likely to be grouped together based on the semantic meaning representations.

The other two frequent errors confuse FlightDelay with TerminalOperation and ChangeFlight respectively. Poor terminal operations often incur unexpected customer delays. Two example query utterances are shown as follows,

q_1 : *who's running operation at mia flight 1088 been waiting for a gate.*

q_2 : *have been sitting in the plane waiting for our gate for 25 minutes.*

Sometimes, a user may express more than one intents in a single query utterance. For example, in the following query utterance, the user complaints about the delay and requests for an alternative flight:

q : *why is ba flight 82 from abuja to london delayed almost 24 hours? and are you offering any alternatives?*

We leave multi-intent induction to future work.

5 Related Work

User intent clustering Automatic discovery of user intents by clustering user utterances is a critical task in understanding the dynamics of a domain with user generated content. Previous work focuses on grouping similar web queries or user questions together using supervised or unsupervised clustering techniques. Kathuria et al. (2010) perform simple k-means clustering on a variety of query traits to understand user intents. Cheung and Li (2012) present an unsupervised method for query intent clustering that produces a pattern consisting of a sequence of semantic concepts and/or lexical items for each intent. Jeon et al. (2005) use machine translation to estimate word translation probabilities and retrieve similar questions from question archives. A variation of k-means algorithm, MiXKmeans, is presented by Deepak (2016) to cluster threads that present on forums and Community Question Answering websites. Haponchyk et al. (2018) propose to cluster questions into intents using a supervised learning method that yields better semantic similarity modeling. Our work focuses on a related but different task that automatically induces user intents for building dialog systems. Two sources of information are naturally available for exploring our deep multi-view clustering approach.

Multi-view clustering Multi-view clustering (MVC) aims at grouping similar subjects into the same cluster by combining the available multi-view feature information to search for consistent cluster assignments across different views (Chao et al., 2017). Generative MVC approaches assume that the data is drawn from a mixture model and the membership information can be inferred using the multi-view EM algorithm (Bickel and Scheffer, 2004). Most of the works on MVC employ discriminative approaches that directly optimize an objective function that involves pairwise similarities so that the average similarity within clusters can be minimized and the average similarity between clusters can be maximized. In particular, Chaudhuri et al. (2009) propose to exploit canonical correlation analysis to learn multi-view representations that are then used for downstream clustering. Multi-view spectral clustering (Kumar et al., 2011; Kanaan-Izquierdo et al., 2018) constructs a similarity matrix for each view and then iteratively updates a matrix using the eigenvectors of the similarity matrix computed on another view. Standard MVC algorithms expect multi-view feature inputs that are fixed during unsupervised clustering. Av-Kmeans works with raw multi-view text inputs and learns representations that are particularly suitable for clustering.

Joint representation learning and clustering Several recent works propose to jointly learn feature representations and clustering via neural networks. Xie et al. (2016) present the deep embedded clustering (DEC) method that learns a mapping from the data space to a lower-dimensional feature space where it iteratively optimizes a KL divergence based clustering objective. Deep clustering network (DCN) (Yang et al., 2016) is a joint dimensional reduction and k-means clustering framework, in which the dimensional reduction model is implemented with a deep neural network. These methods focus on the learning of single-view representations and the multi-view information is under-explored. Lin et al. (2018) present a joint framework for deep multi-view clustering (DMJC) that is the closest work to ours. However, DMJC only works with single-view inputs and the feature representations are learned using a multi-view fusion mechanism. In contrast, AV-KMEANS assumes that the inputs can be naturally partitioned into multiple views and carry out learning with the multi-view inputs directly.

6 Conclusion

We introduce the novel task of dialog intent induction that concerns automatic discovery of dialog intents from user query utterances in human-human conversations. The resulting dialog intents provide valuable insights in helping design goal-oriented dialog systems. We propose to leverage the dialog structure to divide a dialog into two independent views and then present AV-KMEANS, a deep multi-view clustering algorithm, to jointly perform multi-view representation learning and clustering on the views. We conduct extensive experiments on a Twitter conversation dataset and a question intent clustering dataset. The results demonstrate the superiority of AV-KMEANS over competitive representation learning and multi-view clustering baselines. In the future, we would like to abstract multi-view data from multi-lingual and multi-modal sources and investigate the effectiveness of AV-KMEANS on a wider range of tasks in the multi-lingual or multi-modal settings.

Acknowledgments

We thank Shawn Henry and Ethan Elenberg for their comments on an early draft of the paper. We also thank the conversational AI and the general research teams of ASAPP for their support throughout the project. We thank the EMNLP reviewers for their helpful feedback.

References

Steffen Bickel and Tobias Scheffer. 2004. Multi-view clustering. In *Proceedings of the IEEE International Conference on Data Mining (ICDM)*.

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gasic. 2018. Multiwoz-a large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. In *Proceedings of Empirical Methods for Natural Language Processing (EMNLP)*.

Guoqing Chao, Shiliang Sun, and Jinbo Bi. 2017. A survey on multi-view clustering. *arXiv preprint arXiv:1712.06246*.

Kamalika Chaudhuri, Sham M Kakade, Karen Livescu, and Karthik Sridharan. 2009. Multi-view clustering via canonical correlation analysis. In *Proceedings of the International Conference on Machine Learning (ICML)*.

Jackie Chi Kit Cheung and Xiao Li. 2012. Sequence clustering and labeling for unsupervised query intent discovery. In *Proceedings of the ACM international conference on Web search and data mining (WSDM)*.

Padmanabhan Deepak. 2016. Mixkmeans: Clustering question-answer archives. In *Proceedings of Empirical Methods for Natural Language Processing (EMNLP)*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

Iryna Haponchyk, Antonio Uva, Seunghak Yu, Olga Uryupina, and Alessandro Moschitti. 2018. Supervised clustering of questions into intents for dialog system applications. In *Proceedings of Empirical Methods for Natural Language Processing (EMNLP)*.

Charles T Hemphill, John J Godfrey, and George R Doddington. 1990. The atis spoken language systems pilot corpus. In *Proceedings of the Workshop on Speech and Natural Language*.

Matthew Henderson, Blaise Thomson, and Jason D Williams. 2014a. The second dialog state tracking challenge. In *Proceedings of the Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*.

Matthew Henderson, Blaise Thomson, and Steve Young. 2014b. Word-based dialog state tracking with recurrent neural networks. In *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*.

Jiwoon Jeon, W Bruce Croft, and Joon Ho Lee. 2005. Finding similar questions in large question and answer archives. In *Proceedings of the ACM international conference on Information and knowledge management (CIKM)*.

Samir Kanaan-Izquierdo, Andrey Ziyatdinov, and Alexandre Perera-Lluna. 2018. Multiview and multifeature spectral clustering using common eigenvectors. *Pattern Recognition Letters*, 102.

Ashish Kathuria, Bernard J Jansen, Carolyn Hafernik, and Amanda Spink. 2010. Classifying the user intent of web queries using k-means clustering. *Internet Research*, 20.

Seokhwan Kim, Luis Fernando D'Haro, Rafael E Banchs, Jason D Williams, Matthew Henderson, and Koichiro Yoshino. 2016. The fifth dialog state tracking challenge. In *Proceedings of the IEEE Spoken Language Technology Workshop (SLT)*.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Harold W Kuhn. 1955. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2.

Abhishek Kumar, Piyush Rai, and Hal Daume. 2011. Co-regularized multi-view spectral clustering. In *Neural Information Processing Systems (NIPS)*.

Bingqian Lin, Yuan Xie, Yanyun Qu, Cuihua Li, and Xiaodan Liang. 2018. Jointly deep multi-view learning for clustering analysis. *arXiv preprint arXiv:1808.06220*.

Lajanugen Logeswaran and Honglak Lee. 2018. An efficient framework for learning sentence representations. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Nikola Mrkšić, Diarmuid Ó Séaghdha, Blaise Thomson, Milica Gasic, Pei-Hao Su, David Vandyke, Tsung-Hsien Wen, and Steve Young. 2015. Multi-domain dialog state tracking using recurrent neural networks. In *Proceedings of the Association for Computational Linguistics (ACL)*.

Eugene A Nida. 2015. *A componential analysis of meaning: An introduction to semantic structures*, volume 57. Walter de Gruyter GmbH & Co KG.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of Empirical Methods for Natural Language Processing (EMNLP)*.

Eldar Sadikov, Jayant Madhavan, Lu Wang, and Alon Halevy. 2010. Clustering query refinements by user intent. In *Proceedings of the international conference on World wide web*.

J Schmidhuber, U Meier, and D Ciresan. 2012. Multi-column deep neural networks for image classification. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Iulian Vlad Serban, Ryan Lowe, Peter Henderson, Laurent Charlin, and Joelle Pineau. 2018. A survey of available corpora for building data-driven dialogue systems: The journal version. *Dialogue & Discourse*, 9(1).

Darsh Shah, Tao Lei, Alessandro Moschitti, Salvatore Romeo, and Preslav Nakov. 2018. Adversarial domain adaptation for duplicate question detection. In *Proceedings of Empirical Methods for Natural Language Processing (EMNLP)*.

Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical networks for few-shot learning. In *Neural Information Processing Systems (NIPS)*.

Jason Williams. 2012. A belief tracking challenge task for spoken dialog systems. In *Proceedings of the NAACL-HLT Workshop on Future directions and needs in the Spoken Dialog Community*.

Jason Williams, Antoine Raux, Deepak Ramachandran, and Alan Black. 2013. The dialog state tracking challenge. In *Proceedings of the SIGDIAL Conference*.

Junyuan Xie, Ross Girshick, and Ali Farhadi. 2016. Unsupervised deep embedding for clustering analysis. In *Proceedings of the International Conference on Machine Learning (ICML)*.

Bo Yang, Xiao Fu, Nicholas D Sidiropoulos, and Mingyi Hong. 2016. Towards k-means-friendly spaces: Simultaneous deep learning and clustering. In *Proceedings of the International Conference on Machine Learning (ICML)*.