Paper review
Recent studies on dialog intent clustering and intent induction
(Many Legends)

Presentation: **Jeiyoon Park**
6th Generation, TAVE

TAVE Research

Outline

1. Detour: Task Description
2. Method
3. Conclusion

Outline

1. Detour: Task Description
2. Method
3. Conclusion

Detour: Task Description

1. Intent Induction from Task-Oriented Dialogue

1) Open Intent Induction?

- It aims to uncover novel intent categories from user utterances to expand the set of supported intent classes.

A wireless charging case is fancy and all, but can we get a “find my airpod” feature going?

If you have lost your AirPods, Find My iPhone can help you locate them.

{User Intent: *FindAirPods*}

Detour: Task Description

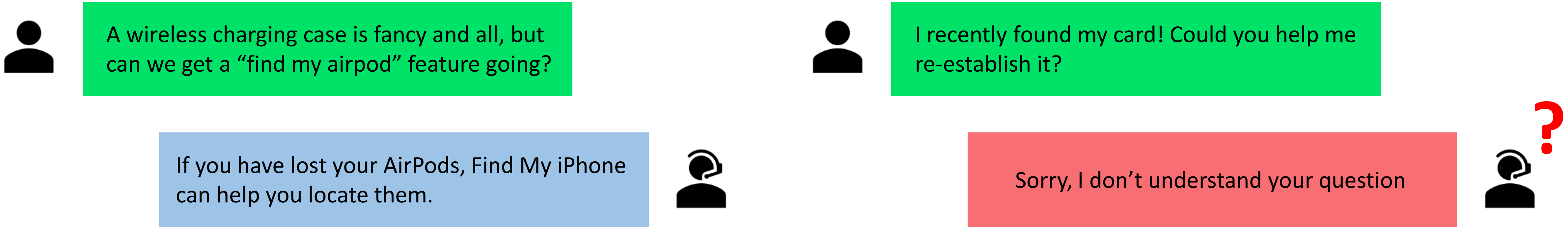
1. Intent Induction from Task-Oriented Dialogue

1) Open Intent Induction?

- It aims to uncover novel intent categories from user utterances to expand the set of supported intent classes.

2) Why Intent Induction?

- The pre-defined intents can’t fully meet customer needs



{User Intent: *FindAirPods*}

https://drive.google.com/file/d/1itlby2Ypq3sRVtOY1alr3ygjPZZdB2TT/view

Detour: Task Description

2. DSTC11 Track Proposal

1) Task 1 - Intent Clustering

- A set of conversation transcripts are given as input, with each turn in the transcripts pre-labeled with its speaker role (i.e., Agent or Customer)

```
[turn 1] Agent: How can I help you today?
[annotation] speaker_role=Agent

[turn 2] Customer: I want to pay my auto
    insurance bill.
[annotation] speaker_role=Customer,
    dialog_act=InformIntent, intent=PayBill

[turn 3] Agent: I will need your account
    number.
[annotation] speaker_role=Agent

[turn 4] Customer: It's five eight one two.
[annotation] speaker_role=Customer
```


```
{
 "conversation_id": "10001",
 "turns": [
   {
     "utterance": "How can I help you
        today?",
     "speaker_role": "Agent",
     "dialog_acts": [],
     "intents": [],
   },
   {
     "utterance": "I want to pay my auto
        insurance bill.",
     "speaker_role": "Customer",
     "dialog_acts": ["InformIntent"],
     "intents": ["PayBill"],
   },
   {
     "utterance": "I will need your
        account number.",
     "speaker_role": "Agent",
     "dialog_acts": [],
     "intents": [],
   },
   {
     "utterance": "It's five eight one
        two.",
     "speaker_role": "Customer",
     "dialog_acts": [],
     "intents": [],
   },
   ...
 ]
}
...
```

Detour: Task Description

2. DSTC11 Track Proposal

1) Task 1 - Intent Clustering

- A set of conversation transcripts are given as input, with each turn in the transcripts pre-labeled with its speaker role (i.e., Agent or Customer)

```
[turn 1] Agent: How can I help you today?
[annotation] speaker_role=Agent
```

```
[turn 2] Customer: I want to pay my auto
    insurance bill.
[annotation] speaker_role=Customer,
    dialog_act=InformIntent, intent=PayBill
```

```
[turn 3] Agent: I will need your account
    number.
[annotation] speaker_role=Agent
```

```
[turn 4] Customer: It's five eight one two.
[annotation] speaker_role=Customer
```


```
{
  "conversation_id": "10001",
  "turns": [
     {
       "utterance": "How can I help you
          today?",
       "speaker_role": "Agent",
       "dialog_acts": [],
       "intents": [],
     },
     {
       "utterance": "I want to pay my auto
          insurance bill.",
       "speaker_role": "Customer",
       "dialog_acts": ["InformIntent"],
       "intents": ["PayBill"],
     },
     {
       "utterance": "I will need your
          account number.",
       "speaker_role": "Agent",
       "dialog_acts": [],
       "intents": [],
     },
     {
       "utterance": "It's five eight one
          two.",
       "speaker_role": "Customer",
       "dialog_acts": [],
       "intents": [],
     },
     ...
  ]
}
...
```

https://drive.google.com/file/d/1itlby2Ypq3sRVtOY1alr3ygjPZZdB2TT/view

Detour: Task Description

2. DSTC11 Track Proposal

1) Task 1 - Intent Clustering

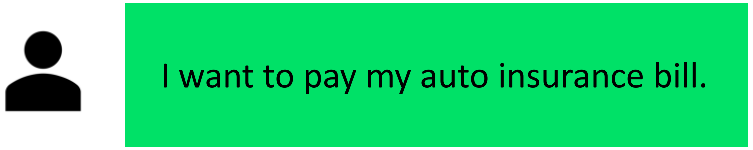
- Task1 requires participants to create a set of intents labels based on the dialouges
- Participants are required to assign an intent label to each dialog turn labeled with "InformIntent" dialog act, where customers express an intent.

```
[turn 1] Agent: How can I help you today?
[annotation] speaker_role=Agent

[turn 2] Customer: I want to pay my auto
    insurance bill.
[annotation] speaker_role=Customer,
    dialog_act=InformIntent, intent=PayBill

[turn 3] Agent: I will need your account
    number.
[annotation] speaker_role=Agent

[turn 4] Customer: It's five eight one two.
[annotation] speaker_role=Customer
```



{User Intent: *PayBill*}

Detour: Task Description

2. DSTC11 Track Proposal

1) Task 1 - Intent Clustering

- Task1 requires participants to create a set of intents labels based on the dialouges
- Participants are required to assign an intent label to each dialog turn labeled with "InformIntent" dialog act, where customers express an intent.

```
[turn 1] Agent: How can I help you today?
[annotation] speaker_role=Agent

[turn 2] Customer: I want to pay my auto
    insurance bill.
[annotation] speaker_role=Customer,
    dialog_act=InformIntent, intent=PayBill

[turn 3] Agent: I will need your account
    number.
[annotation] speaker_role=Agent

[turn 4] Customer: It's five eight one two.
[annotation] speaker_role=Customer
```

I want to pay my auto insurance bill.

{User Intent: ~~PayBill~~ *INTENT_1*}

Note that in this track, the intent labels are treated as unique IDs and are not evaluated for linguistic meaning

```
{"predicted_label": "INTENT_1", "
    conversation_id": "10001", "turn_number
    ": 0}
{"predicted_label": "INTENT_2", "
    conversation_id": "10001", "turn_number
    ": 12}
{"predicted_label": "INTENT_1", "
    conversation_id": "10002", "turn_number
    ": 6}
...
```

https://drive.google.com/file/d/1itlby2Ypq3sRVtOY1alr3ygjPZZdB2TT/view

Detour: Task Description

2. DSTC11 Track Proposal

2) Task 2 – Open Intent Induction

- Participants will not have access to the ground-truth dialog act labels in the dataset.
- The input is a set of transcripts with only speaker role labeled


```
{
  "intents": [
    {
      "label": "INTENT_1",
      "sampleUtterances": [
        "I want to buy an insurance plan
          ",
        "I'm trying to find an insurance
           plan",
        "Enroll insurance plan",
        ...
      ]
    },
    ...
  ]
}
```


- test_utterances.jsonl


```
{"utterance": "My name is John Alley, just moved to a town house and want a quote for renter's insurance please", "utterance_id": "insurance_0026", "intent": "GetQuote"}
{"utterance": "Hi, I am looking for a renter's insurance. Can you please quote?", "utterance_id": "insurance_0027", "intent": "GetQuote"}
{"utterance": "hi, i want to get a quote for renter's insurance.", "utterance_id": "insurance_0028", "intent": "GetQuote"}
{"utterance": "Hello! I just wanted to check renter's insurance ", "utterance_id": "insurance_0029", "intent": "GetQuote"}
{"utterance": "Hello! I've been wanting to get a quote for my renters insurance, and I live in Market Park apartment", "utterance_id": "insurance_0030", "intent": "GetQuote"}
{"utterance": "I need to know the cost for renter's insurance", "utterance_id": "insurance_0031", "intent": "GetQuote"}
{"utterance": "hi my name is george hall and i want to get a quote for my renter's insurance", "utterance_id": "insurance_0032", "intent": "GetQuote"}
{"utterance": "Hello! Im from Texas and I need info of the renter's insurance.", "utterance_id": "insurance_0033", "intent": "GetQuote"}
{"utterance": "Hi, can I get a quote for renter's insurance please?", "utterance_id": "insurance_0034", "intent": "GetQuote"}
{"utterance": "Hi can I get a renter's insurance please?", "utterance_id": "insurance_0035", "intent": "GetQuote"}
{"utterance": "hi, i'd want to get a quote for renter's insurance.", "utterance_id": "insurance_0036", "intent": "GetQuote"}
{"utterance": "Hi I'm looking for a renter insurance quote. My name is Mike", "utterance_id": "insurance_0037", "intent": "GetQuote"}
{"utterance": "Hey there! Can you get me a quote for renter's insurance? The property market is simply crazy!", "utterance_id": "insurance_0038", "intent": "GetQuote"}
{"utterance": "Hi, please help me get a quote for renter's insurance", "utterance_id": "insurance_0039", "intent": "GetQuote"}
{"utterance": "Hi there, I live at Woodland Apartments and need a quote for renter's insurance, trying to insure my apartment please!", "utterance_id": "insurance_0040", "intent": "GetQuote"}
{"utterance": "hi i want to get a quote for renter's insurance.", "utterance_id": "insurance_0041", "intent": "GetQuote"}
{"utterance": "Hi, good evening. I'm wondering if I'm able to get a quote for renter's insurance?", "utterance_id": "insurance_0042", "intent": "GetQuote"}
{"utterance": "Renters' insurance for me and wife, please", "utterance_id": "insurance_0043", "intent": "GetQuote"}
{"utterance": "How much does renter's inusurance cost?", "utterance_id": "insurance_0044", "intent": "GetQuote"}
{"utterance": "Yeah, hi, I live in the Willow Lake Apartments and I'm like to get renter's insurance please, can you help with that?", "utterance_id": "insurance_0045", "intent": "GetQuote"}
{"utterance": "I'm looking to file a life insurance claim, can you help with that?", "utterance_id": "insurance_0046", "intent": "FileClaim"}
{"utterance": "hi, i want to file a life insurance claim.", "utterance_id": "insurance_0047", "intent": "FileClaim"}
{"utterance": "I'm really looking to file an insurance claim, wondering if you could help me out?", "utterance_id": "insurance_0048", "intent": "FileClaim"}
{"utterance": "i want to file a life insurance claim.", "utterance_id": "insurance_0049", "intent": "FileClaim"}
{"utterance": "Yes, hey, please help file a life insurance claim", "utterance_id": "insurance_0050", "intent": "FileClaim"}
{"utterance": "I sadly need to file a life insurance claim for my father who died of a heart attack please.", "utterance_id": "insurance_0051", "intent": "FileClaim"}
{"utterance": "Hi, please help me file a life insurance claim", "utterance_id": "insurance_0052", "intent": "FileClaim"}
```

https://drive.google.com/file/d/1itlby2Ypq3sRVtOY1alr3ygjPZZdB2TT/view

Detour: Task Description

```
{
  "intents": [
    {
      "label": "INTENT_1",
      "sampleUtterances": [
        "I want to buy an insurance plan
          ",

        "I'm trying to find an insurance
          plan",
        "Enroll insurance plan",
        ...
      ]
    },
    ...
  ]
}
```

2. DSTC11 Track Proposal

2) Task 2 – Open Intent Induction

- Participants are required to generate a set of intents
- Participants are free to choose how they'd like to produce the sample utterance.

e.g.,) the utterances can either be extracted from transcripts or generated by model.

- Participants are welcome to use their own dialog act predictions as long as the model was trained only with the provided development dataset

https://drive.google.com/file/d/1itlby2Ypq3sRVtOY1alr3ygjPZZdB2TT/view

Detour: Task Description

2. DSTC11 Track Proposal

3) Datasets

- One dev dataset and two eval datasets
- Each dataset consists of 1K customer support spoken conversations

- Dev dataset is an insurance-related customer support dataset, which contains 948 human-to-human conversations

- Eval datasets consist of customer service conversations in a new (non-insurance) domain.

[Agent] Hello this is Jane at Rivertown Insurance. How can I help you today?
[Customer] Hi, this is Joe Last and I would like to pay my auto insurance bill.
[Agent] Okay mister Last. I can help you with payment. #Ah first I will need your account number please.
[Customer] #Ah I do not have it with me. Is there another way to find my account?
[Agent] Of course mister Last. One moment. What is your current address on your account?
[Customer] Two four six eight Rural Lane. Hometown MI eight six four two zero.
[Agent] Two four six eight Rural Lane. Hometown MI eight six four two zero. Is that correct mister Last?
[Customer] #Um. Yes and you can call me Joe.
[Agent] Alright Joe. May I have your current phone number please.
[Customer] Area code five eight two one seven four zero three six nine.
[Agent] Eight five two one seven four zero three six nine, correct?
[Customer] #Uh no. It’s five eight one two.
[Agent] #Oh I’m sorry. Let’s try this again. Okay five eight two
[Customer] One seven four zero three six nine.
[Agent] Okay let me repeat five eight two one seven four zero three six nine. Correct?
[Customer] Uh huh. Yes.
[Agent] Great. Thanks Joe. Now do you remember your security question for the account?
[Customer] Oh I’m not sure
[Agent] It would have been the one you set up whe-
[Customer] #Ah I think I remember my father’ s middle name right?
[Agent] That’s corrrect Joe.
[Customer] Okay, dad’s middle name is Christop-no Charles. I hope .
[Agent] Yes that’s correct mister Joe.
[Customer] Whew. And my pin number is nine seven one three.
[Agent] Great and I also need your date of birth please.
[Customer] My birthday is one one sixty one.
[Agent] Alright Joe almost done. let’s see January first nineteen sixty one. Correct?
[Customer] Yep. I’m getting old.
[Agent] Aren’t we all Joe, aren’t we all.
[Customer] Should get a getting old discount #eh?
[Agent] #Oh I wish I could Joe, but you have some great discounts already.
[Customer] Ju-just thought I’d try.
[Agent] Okay Joe. Your account number is two three one five six four eight seven nine.
[Customer] Wait let me write that down. Okay can you repeat that, what was your name again?
[Agent] Jane, are you ready?
[Customer] Ye- oh wait a moment.
[Agent] Not a problem.
[Customer] Ready Jane. Shoot.
[Agent] Two three one.
[Customer] Two three one.
[Agent] Five six four.
[Customer] Five six four.
[Agent] Eight seven nine.
[Customer] Eight seven nine. I’m sure I’ll misplace this too.
[Agent] No problem. Whenever you call you’ll just have to go through this process and an agent can assist you.
[Customer] #Oh I’m sure I’ll meet all of the agents, soon-sooner or later.
[Agent] We love to meet our customers sooner than later.
[Customer] So what is my payment?
[Agent] Okay Joe. I’m showing that you are current right now and no payment is due until September.
[Customer] Whoa. Really. Well okay Jane. I’m going to take your word for it. Thank you. Talk to you in September.
[Agent] Is there anything else that I can help you with today mister Joe Last? Hello, Joe. Are you still there?

https://drive.google.com/file/d/1itlby2Ypq3sRVtOY1alr3ygjPZZdB2TT/view

Detour: Task Description

2. DSTC11 Track Proposal

4) Evaluation

- Normalized Mutual Information (NMI)
- Accuracy (ACC)
- Precision / Recall / F1
- Intent Example Coverage / Recall
- Adjusted Rand Index (ARI)

```
def compute_metrics_from_turn_predictions(
    turn_predictions: List[TurnPrediction],
    metrics: List[ClusteringMetric] = None,
    ignore_labels: List[str] = None,
) -> Dict[str, Any]:
    if not metrics:
        metrics = [
            NMI(),
            ARI(),
            ClusteringAccuracy(),
            ClusteringPrecision(),
            ClusteringRecall(),
            ClusteringF1(),
            ExampleCoverage(),
            NumberOfReferenceLabels(),
            NumberOfClusters()
        ]
```

Detour: Task Description

2. DSTC11 Track Proposal

4) Evaluation

- Normalized Mutual Information (NMI):

$$NMI = \frac{I(X;Y)}{\min(H(X),H(Y))} \leq 1$$

, where X denotes reference labels and Y is clustered labels

- Accuracy (ACC):

$$ACC = \frac{The\ number\ of\ \{aligned == ref\}}{The\ number\ of\ ref}\ (if\ ref\ else\ 0)$$

https://drive.google.com/file/d/1itlby2Ypq3sRVtOY1alr3ygjPZZdB2TT/view

Detour: Task Description

2. DSTC11 Track Proposal

```python
def compute_metric(self, cluster_labels: List[str], reference_labels: List[str]) -> float:
    cluster_alignment = compute_many_to_one_alignment(cluster_labels, reference_labels)
    covered_intents = set(cluster_alignment.values())
    covered_count = sum([1 for label in reference_labels if label in covered_intents])
    coverage = (100 * covered_count / len(reference_labels)) if reference_labels else 0
    return coverage
```

4) Evaluation

- Precision: A many-to-one alignment is computed from cluster labels to reference labels such that the number of correct aligned labels is maximized.

- Recall: A many-to-one alignment is computed from reference labels to cluster labels such that the number of correct aligned labels is maximized.

- Intent Example Coverage: Percent of examples whose reference intent has a corresponding predicted cluster after performing a many-to-one alignment from predicted clusters to reference clusters.

- Adjusted Rand Index (ARI): The ratio of the number of correct pairs for all possible pairs.

https://drive.google.com/file/d/1itlby2Ypq3sRVtOY1alr3ygjPZZdB2TT/view

Detour: Task Description

2. DSTC11 Track Proposal

5) Summary

```
RunID    NMI ARI ACC Precision    Recall   F1   Example Coverage     Reference K K
kmeans_all-mpnet-base-v2     59.3     32.3     46.1     66.0     47.1     54.9     96.5     22.0     42.0
kmeans_glove-840b-300d   30.5     7.0 20.6     34.6     22.2     27.0     92.2     22.0     50.0
```

```
RunID    NMI ARI ACC Precision    Recall   F1   Example Coverage     Reference K K   # Intents    # Utterances     # Utterances per Intent
kmeans_all-mpnet-base-v2     72.8     41.6     60.2     65.6     76.3     70.6     86.5     22.0     26.0     42.0     3449.0  82.1
kmeans_glove-840b-300d   50.6     24.0     32.3     40.1     64.0     49.3     66.8     22.0     29.0     49.0     3449.0  70.4
```

	Task 1 - Intent Clustering	**Task 2 - Open Intent Induction**
Goal	Assign labels to customer turns in transcripts that express intents	Induce a set of intents for creating a simple task-oriented chatbot
Input	Conversation transcripts with each turn in the transcripts marked with a speaker role and dialog act label	Conversation transcripts labeled only with speaker roles, predicted (automatic) dialog act classifier labels for each turn
Output	Intent labels assigned to dialog turns with "InformIntent" dialog act label	Intents induced from conversation transcripts, each with a set of corresponding sample utterances
Metrics	NMI and Accuracy	1. Intent Classification F1-Score 2. Prediction-based NMI and Accuracy 3. Intent Example Coverage/Recall
Baselines	K-means with Glove/BERT embeddings	K-means with Glove/BERT embeddings (using provided dialog act predictions and naive utterance selection)

Table 1: Summary of proposed tasks.

Outline

1. Detour: Task Description
2. Method
3. Conclusion

Method

1. Alternating-View K-Means (AV-KMEANS)

1) Paper: Dialog Intent Induction with Deep Multi-View Clustering (Perkins et al., EMNLP 2020)

2) Why this paper?

- The intuition that a dialog intent is not only expressed in the user query utterance but also captured in the rest of the dialog

- AV-KMEANS splits a conversation into two independent views (user query and rest of the conversation) and exploit multi-view clustering techniques for inducing the dialog intent.

- K-means-style updates (Alternating-view k-means)

https://arxiv.org/pdf/1908.11487.pdf

Method

1. Alternating-View K-Means (AV-KMEANS)

3) Notations

x_i: Each data point

$x_i^{(1)}$: Query-viewed x_i (i.e., user query)

$x_i^{(2)}$: Content view x_i (i.e., rest of the conversation)

f_ϕ: Neural network encoders

f_{ϕ_1}: Query-viewed f_ϕ (BiLSTM)

f_{ϕ_2}: Content-viewed f_ϕ (BiLSTM)

e.g.) a snippet from "AskUbuntu" dataset

View1) how can i set the software center to install software for non-root users ?

View2) you can modify the policykit permissions to allow the users to access the aptdaemon backend that software centre uses . shows that is the file specifying the actions possible on the aptdaemon backend . looking in that file , the tags specify the possible actions . you 'd probably want to allow users to install new packages from the archive , and to allow users to update the package lists . see which documents how to set local permissions on policykit actions . putting the following into will allow any user logged in to the local machine to install packages after typing their own password (even when they 're not in the admin group) and to update the package cache without typing any password .

Method

1. Alternating-View K-Means (AV-KMEANS)

4) Alternating-view k-means

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_{ϕ_1} and f_{ϕ_2}

Initialize f_{ϕ_1} and f_{ϕ_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_{ϕ_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$(\{\mathbf{x}_i^{(2)}\}, K, M, \{z_i^{(1)}\})$

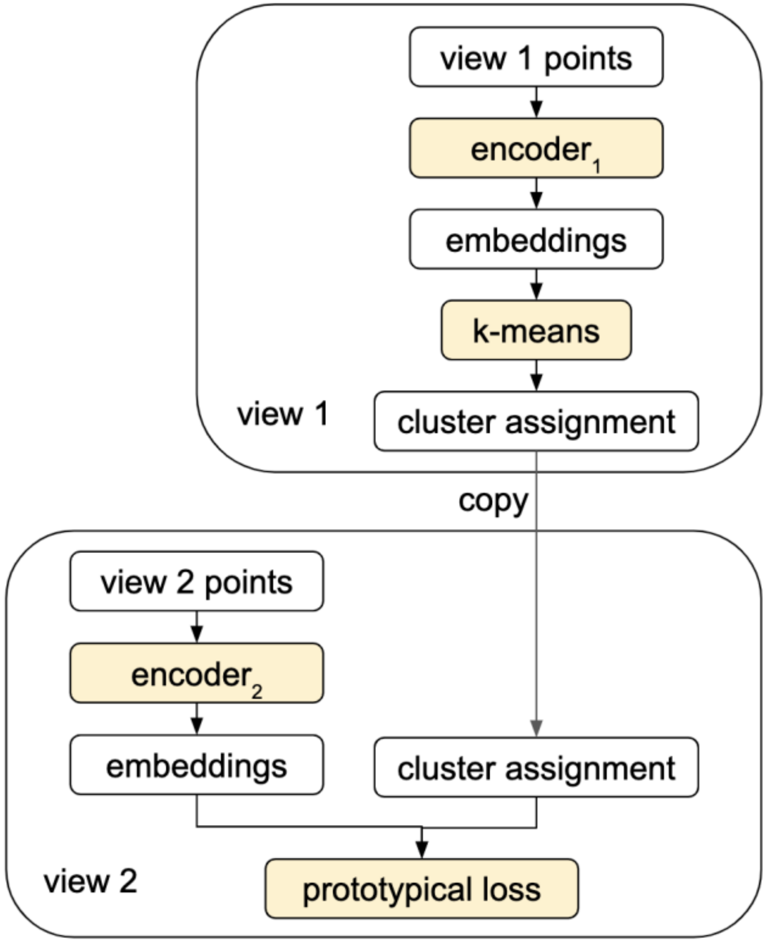
// `project cluster assignment from view 2 to view 1`

Update f_{ϕ_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



https://arxiv.org/pdf/1908.11487.pdf

Method

1. Alternating-View K-Means (AV-KMEANS)

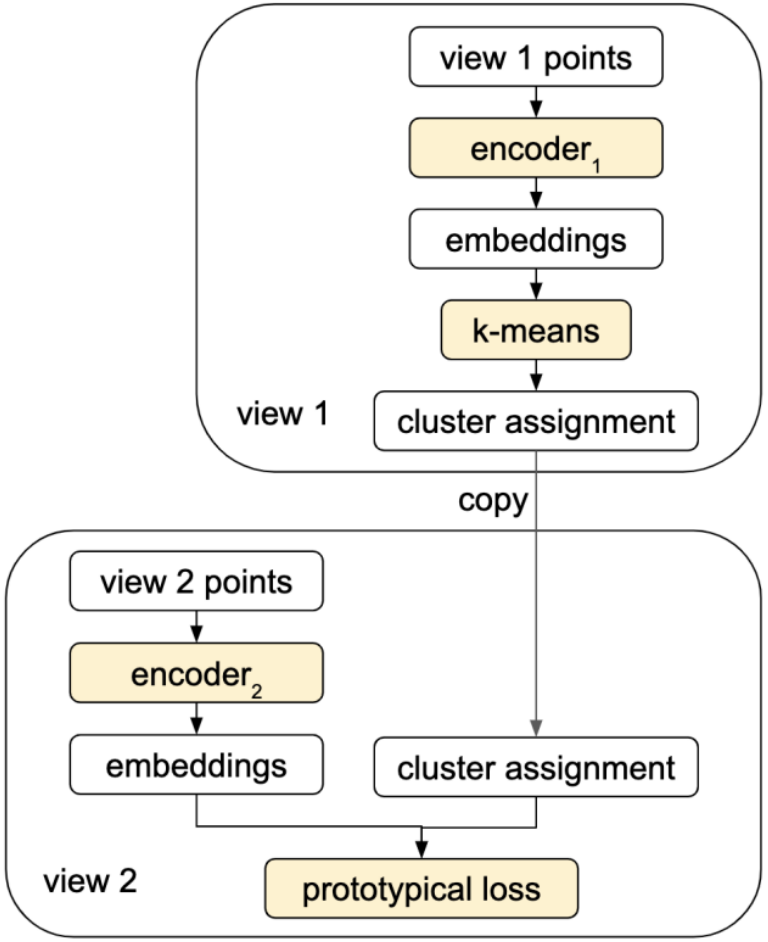
4) Alternating-view k-means

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_{ϕ_1} and f_{ϕ_2}

Initialize f_{ϕ_1} and f_{ϕ_2} (§ 2.3)
$\{z_i^{(1)}\} \leftarrow \text{K-MEANS}(\{f_{\phi_1}(x_i^{(1)})\}, K)$
for $t = 1, \cdots, T$ **do**
  // `project cluster assignment from view 1 to view 2`
  Update f_{ϕ_2} with pseudo training instances $\{(x_i^{(2)}, z_i^{(1)})\}$ (§ 2.2)
  Encode view-2 inputs: $\{\mathbf{x}_i^{(2)} \leftarrow f_{\phi_2}(x_i^{(2)})\}$
  $\{z_i^{(2)}\} \leftarrow \text{K-MEANS}(\{\mathbf{x}_i^{(2)}\}, K, M, \{z_i^{(1)}\})$

  // `project cluster assignment from view 2 to view 1`
  Update f_{ϕ_1} with pseudo training instances $\{(x_i^{(1)}, z_i^{(2)})\}$ (§ 2.2)
  Encode view-1 inputs: $\{\mathbf{x}_i^{(1)} \leftarrow f_{\phi_1}(x_i^{(1)})\}$
  $\{z_i^{(1)}\} \leftarrow \text{K-MEANS}(\{\mathbf{x}_i^{(1)}\}, K, M, \{z_i^{(2)}\})$
end



Method

1. Alternating-View K-Means (AV-KMEANS)

4) Alternating-view k-means

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_{ϕ_1} and f_{ϕ_2}

Initialize f_{ϕ_1} and f_{ϕ_2} (§ 2.3)

$\{z_i^{(1)}\} \leftarrow \text{K-MEANS}(\{f_{\phi_1}(x_i^{(1)})\}, K)$

for $t = 1, \cdots, T$ **do**

// project cluster assignment from view 1 to view 2

Update f_{ϕ_2} with pseudo training instances $\{(x_i^{(2)}, z_i^{(1)})\}$ (§ 2.2)

Encode view-2 inputs: $\{\mathbf{x}_i^{(2)} \leftarrow f_{\phi_2}(x_i^{(2)})\}$

$\{z_i^{(2)}\} \leftarrow \text{K-MEANS}(\{\mathbf{x}_i^{(2)}\}, K, M, \{z_i^{(1)}\})$

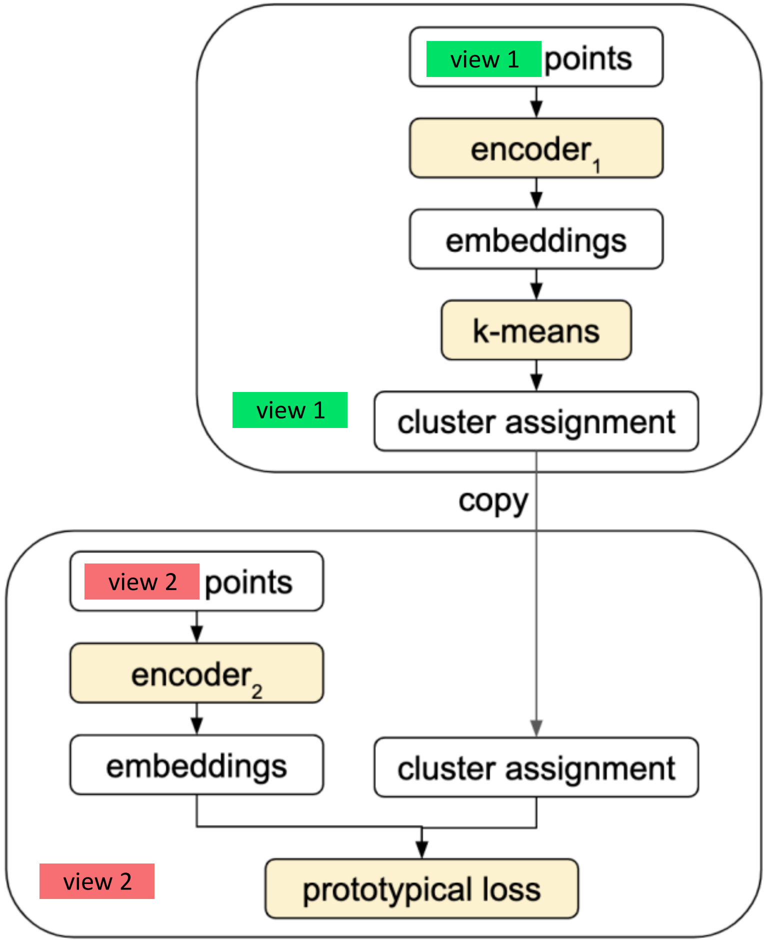
// project cluster assignment from view 2 to view 1

Update f_{ϕ_1} with pseudo training instances $\{(x_i^{(1)}, z_i^{(2)})\}$ (§ 2.2)

Encode view-1 inputs: $\{\mathbf{x}_i^{(1)} \leftarrow f_{\phi_1}(x_i^{(1)})\}$

$\{z_i^{(1)}\} \leftarrow \text{K-MEANS}(\{\mathbf{x}_i^{(1)}\}, K, M, \{z_i^{(2)}\})$

end



Method

1. Alternating-View K-Means (AV-KMEANS)

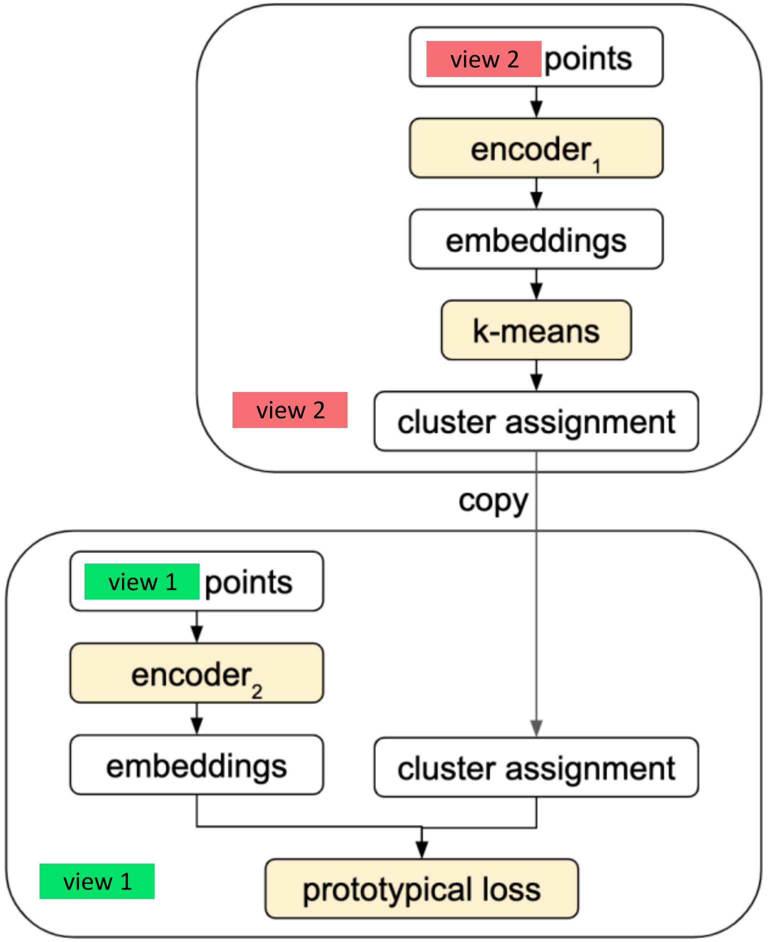
4) Alternating-view k-means

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_{ϕ_1} and f_{ϕ_2}

Initialize f_{ϕ_1} and f_{ϕ_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_{ϕ_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$(\{\mathbf{x}_i^{(2)}\}, K, M, \{z_i^{(1)}\})$

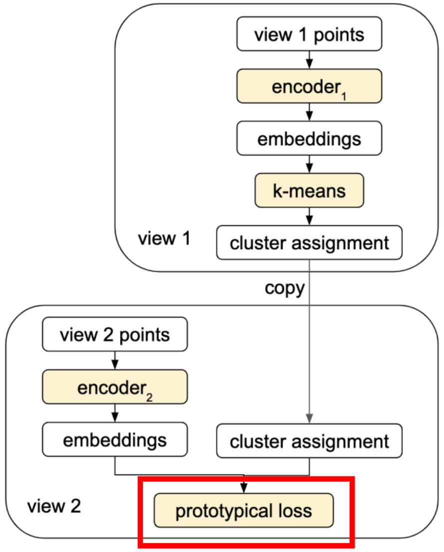
  // project cluster assignment from view 2 to view 1
  Update f_{ϕ_1} with pseudo training instances $\{(x_i^{(1)}, z_i^{(2)})\}$ (§ 2.2)
  Encode view-1 inputs: $\{\mathbf{x}_i^{(1)} \leftarrow f_{\phi_1}(x_i^{(1)})\}$
  $\{z_i^{(1)}\} \leftarrow$ K-MEANS$(\{\mathbf{x}_i^{(1)}\}, K, M, \{z_i^{(2)}\})$
end



https://arxiv.org/pdf/1908.11487.pdf

Method

1. Alternating-View K-Means (AV-KMEANS)



Jointly optimized

5) Prototypical episode training

- Given input data $\{(x_i, z_i)\}$ and encoder f_ϕ, prototypical networks compute a D-dimensional representation c_k (or prototype):

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

- Prototypical networks produce a distribution over classes based on a Softmax over distances to the prototypes in the embedding space:

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

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

https://arxiv.org/pdf/2201.06731.pdf

Method

2. Density-based Deep Clustering Ensemble (DDCE)

1) Paper: Dialog Intent Induction via Density-based Deep Clustering Ensemble (Pu et al., DSTC10 at AAAI2022)

2) Why this paper?

- Previous works are mostly based on K-means algorithm, which has two limitations:
  - “K” is challenging to determine before clustering
  - K-means-based methods can’t effectively exclude outliers

- Outliers are irrelevant user utterances that should not be mapped to any dialog intent.

- DDCE is a clustering ensemble framework that combines multiple base clustering model with corresponding text encoders.

意图: 你从学校里学到了什么
Intent: What have you learned from school

- 今天上学学了什么？
  What did you learn in school today?
- 今天学会了什么？
  What did you learn today?
- 今天上课学了什么呀
  What did you learn in class today?
- 上学学了啥
  What did you learn in school?
- 老师教你什么
  What your teacher teaches you
- *今天(^_^)(^_^)(^_^)*
  today (^_^)(^_^)(^_^)
- *什么*
  what

outliers

Method

2. Density-based Deep Clustering Ensemble (DDCE)

3) Notations

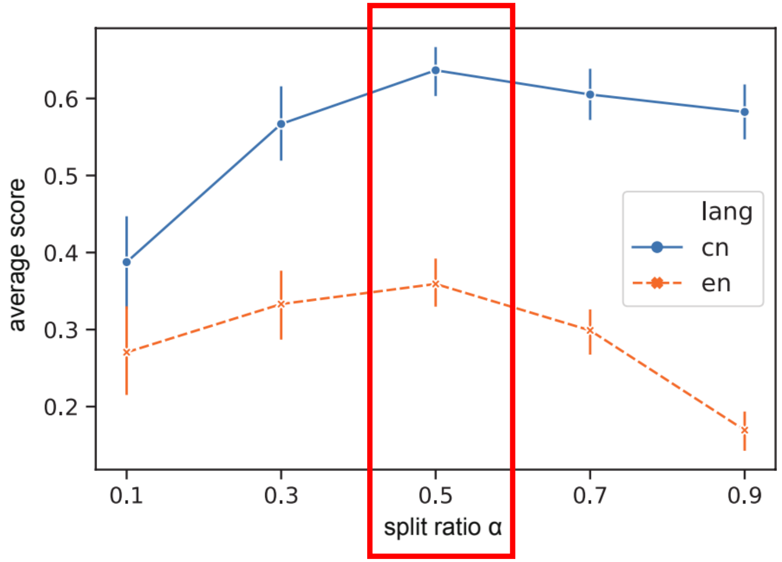
D_l: Labeled dataset

$D_l = D_l^{rl} + D_l^{hs}$, where D_l^{rl} is for fine-tuning the encoder and D_l^{hs} is for searching the best hyperparameter (Both are not overlapped)

Y: Pre-defined intent

O: The size of Y

- αO intents correspond to D_l^{hs}
- $(1-\alpha)O$ intents correspond to D_l^{rl}



Method

2. Density-based Deep Clustering Ensemble (DDCE)

4) DDCE

Algorithm 1: DDCE

Input: $D_l = \{(x_j, y_j), j = 1, ..., N\}$,
$D_{ul} = \{u_i, i = 1, ..., M\}$;
Ouput: $T_{ul} = \{t_i, i = 1, ..., M\}$;
Require: The number of base cluster models K, the collection of predefined intents Y, the size of Y is O, the split ratio α, hyperparameter search space hp;
Training:
for $k = 1, ..., K$ **do**
  split D_l into D_l^{rl} which contains examples corresponding to $(1-\alpha)O$ intents, and D_l^{hs} which contains examples corresponding to αO intents
  Initialize text encoder f_θ^k with pre-trained weights
  Update θ^k after trained on D_l^{rl}
  Compute embeddings E_l^{hs} of D_l^{hs} using f_θ^k
  Search the best hyperparameters hp^k on E_l^{hs}
  Calculate $score_c^k$ on D_l^{hs} with hp^k
end
Inference:
for $k = 1, ..., K$ **do**
  Compute embeddings E_{ul} of D_{ul} using f_θ^k
  Do clustering over E_{ul} with hyperparameters hp^k
end
Apply a consensus function (e.g. BOKV) on $(T_{ul}^1, ..., T_{ul}^K; score_c^1, ..., score_c^K)$ to obtain T_{ul}

Method

2. Density-based Deep Clustering Ensemble (DDCE)

4) DDCE

Algorithm 1: DDCE

Input: $D_l = \{(x_j, y_j), j = 1, ..., N\}$,
$D_{ul} = \{u_i, i = 1, ..., M\}$;
Ouput: $T_{ul} = \{t_i, i = 1, ..., M\}$;
Require: The number of base cluster models K, the collection of predefined intents Y, the size of Y is O, the split ratio α, hyperparameter search space hp;
Training:
for $k = 1, ..., K$ **do**
  split D_l into D_l^{rl} which contains examples corresponding to $(1-\alpha)O$ intents, and D_l^{hs} which contains examples corresponding to αO intents
  Initialize text encoder f_θ^k with pre-trained weights
  Update θ^k after trained on D_l^{rl}
  Compute embeddings E_l^{hs} of D_l^{hs} using f_θ^k
  Search the best hyperparameters hp^k on E_l^{hs}
  Calculate $score_c^k$ on D_l^{hs} with hp^k
end
Inference:
for $k = 1, ..., K$ **do**
  Compute embeddings E_{ul} of D_{ul} using f_θ^k
  Do clustering over E_{ul} with hyperparameters hp^k
end
Apply a consensus function (e.g. BOKV) on $(T_{ul}^1, ..., T_{ul}^K; score_c^1, ..., score_c^K)$ to obtain T_{ul}

D_l: Labeled dataset
D_{ul}: Unlabeled dataset
t_i: Cluster label

https://arxiv.org/pdf/2201.06731.pdf

Method

2. Density-based Deep Clustering Ensemble (DDCE)

4) DDCE

Algorithm 1: DDCE

Input: $D_l = \{(x_j, y_j), j = 1, ..., N\}$,
$D_{ul} = \{u_i, i = 1, ..., M\}$;
Ouput: $T_{ul} = \{t_i, i = 1, ..., M\}$;
Require: The number of base cluster models K, the collection of predefined intents Y, the size of Y is O, the split ratio α, hyperparameter search space hp;
Training:
for $k = 1, ..., K$ **do**
- split D_l into D_l^{rl} which contains examples corresponding to $(1-\alpha)O$ intents, and D_l^{hs} which contains examples corresponding to αO intents
- Initialize text encoder f_θ^k with pre-trained weights
- Update θ^k after trained on D_l^{rl}
- Compute embeddings E_l^{hs} of D_l^{hs} using f_θ^k
- Search the best hyperparameters hp^k on E_l^{hs}
- Calculate $score_c^k$ on D_l^{hs} with hp^k

end
Inference:
for $k = 1, ..., K$ **do**
- Compute embeddings E_{ul} of D_{ul} using f_θ^k
- Do clustering over E_{ul} with hyperparameters hp^k

end
Apply a consensus function (e.g. BOKV) on $(T_{ul}^1, ..., T_{ul}^K; score_c^1, ..., score_c^K)$ to obtain T_{ul}

Training $K(= 5)$ base clustering model over D_l

Calculating the performance score $score_c$ respectively

($score_c$: Harmonic mean of ARI and Recall of non-outlier samples)

Method

2. Density-based Deep Clustering Ensemble (DDCE)

4) DDCE

Algorithm 1: DDCE

Input: $D_l = \{(x_j, y_j), j = 1, ..., N\}$,
$D_{ul} = \{u_i, i = 1, ..., M\}$;
Ouput: $T_{ul} = \{t_i, i = 1, ..., M\}$;
Require: The number of base cluster models K, the collection of predefined intents Y, the size of Y is O, the split ratio α, hyperparameter search space hp;
Training:
for $k = 1, ..., K$ **do**
- split D_l into D_l^{rl} which contains examples corresponding to $(1 - \alpha)O$ intents, and D_l^{hs} which contains examples corresponding to αO intents
- Initialize text encoder f_θ^k with pre-trained weights
- Update θ^k after trained on D_l^{rl}
- Compute embeddings E_l^{hs} of D_l^{hs} using f_θ^k
- Search the best hyperparameters hp^k on E_l^{hs}
- Calculate $score_c^k$ on D_l^{hs} with hp^k

end
Inference:
for $k = 1, ..., K$ **do**
- Compute embeddings E_{ul} of D_{ul} using f_θ^k
- Do clustering over E_{ul} with hyperparameters hp^k

end
Apply a consensus function (e.g. BOKV) on $(T_{ul}^1, ..., T_{ul}^K; score_c^1, ..., score_c^K)$ to obtain T_{ul}

Do clustering over D_{ul}, obtaining K groups of cluster labels $T_{ul} = \{t_i \mid i = 1, ..., M\}$

Method

2. Density-based Deep Clustering Ensemble (DDCE)

4) DDCE

Algorithm 1: DDCE

Input: $D_l = \{(x_j, y_j), j = 1, ..., N\}$,
$D_{ul} = \{u_i, i = 1, ..., M\}$;
Ouput: $T_{ul} = \{t_i, i = 1, ..., M\}$;
Require: The number of base cluster models K, the collection of predefined intents Y, the size of Y is O, the split ratio α, hyperparameter search space hp;
Training:
for $k = 1, ..., K$ **do**
  split D_l into D_l^{rl} which contains examples corresponding to $(1 - \alpha)O$ intents, and D_l^{hs} which contains examples corresponding to αO intents
  Initialize text encoder f_θ^k with pre-trained weights
  Update θ^k after trained on D_l^{rl}
  Compute embeddings E_l^{hs} of D_l^{hs} using f_θ^k
  Search the best hyperparameters hp^k on E_l^{hs}
  Calculate $score_c^k$ on D_l^{hs} with hp^k
end
Inference:
for $k = 1, ..., K$ **do**
  Compute embeddings E_{ul} of D_{ul} using f_θ^k
  Do clustering over E_{ul} with hyperparameters hp^k
end
Apply a consensus function (e.g. BOKV) on $(T_{ul}^1, ..., T_{ul}^K; score_c^1, ..., score_c^K)$ to obtain T_{ul}

Best of K (BOK)

$$T_{ul}^* = \underset{T \in \mathbb{T}_K}{\arg\max} \sum_{j=1}^{K} \mathrm{NMI}\left(T, T_j\right)$$

Applying a consensus function (BOKV) over the results:

$$u_i = \begin{cases} 1 & \text{if } \operatorname{argmax}_{t_j} \left\{\sum_{j=1}^{K} t_k\right\} = l^{out} \\ 0 & \text{if } \operatorname{argmax}_{t_j} \left\{\sum_{j=1}^{K} t_k\right\} \neq l^{out} \end{cases}$$

($I_{out} = \{i \mid u_i = 1\}$, by majority voting)

$$T_{ul}^{nout} = \underset{T^{nout} \in \mathbb{T}_K^{nout}}{\arg\max} \sum_{j=1}^{K} \mathrm{NMI}\left(T^{nout}, T_j^{nout}\right)$$

($T^{nout} = \{t_i, i \in I_{nout}\}$, non-outlier)

Method

3. Multi-Task Pre-Training and Contrastive Learning with Nearest Neighbors (MTP-CLNN)

1) Paper: New Intent Discovery with Pre-training and Contrastive Learning (Zhang et la., ACL 2022)

2) Why this paper?

- First, a multi-task pre-training strategy is applied

- Then, a new contrastive loss is employed to a novel self-supervised clustering

- The code is available 

Method

3. Multi-Task Pre-Training and Contrastive Learning with Nearest Neighbors (MTP-CLNN)

3) Notations

C_k: a set of defined (known) intents
C_u: a set of unknown intents

$$D_{known}^{labeled} = \{(x_i, y_i) |\ y_i \in C_k\}$$
$$D^{unlabeled} = \{x_i |\ y_i \in \{C_k, C_u\}\}$$

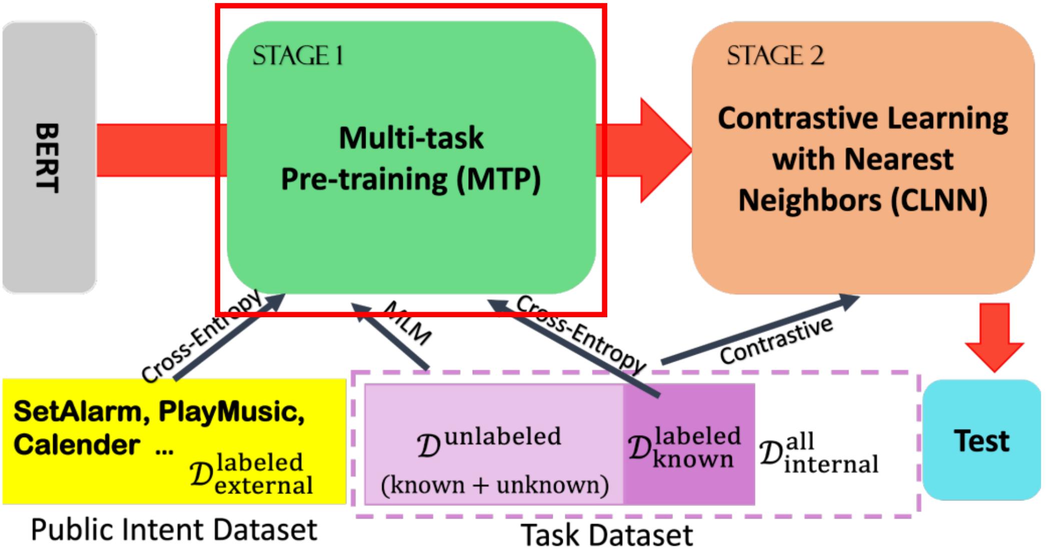
Method

3. Multi-Task Pre-Training and Contrastive Learning with Nearest Neighbors (MTP-CLNN)

4) Stage 1: Multi-task Pre-training (MTP)

$$\mathcal{L}_{\mathrm{stg1}} = \underbrace{\mathcal{L}_{\mathrm{ce}}(\mathcal{D}^{\mathrm{labeled}}_{\mathrm{external}};\theta)}_{\text{supervised}} + \underbrace{\mathcal{L}_{\mathrm{mlm}}(\mathcal{D}^{\mathrm{all}}_{\mathrm{internal}};\theta)}_{\text{self-supervised}},$$

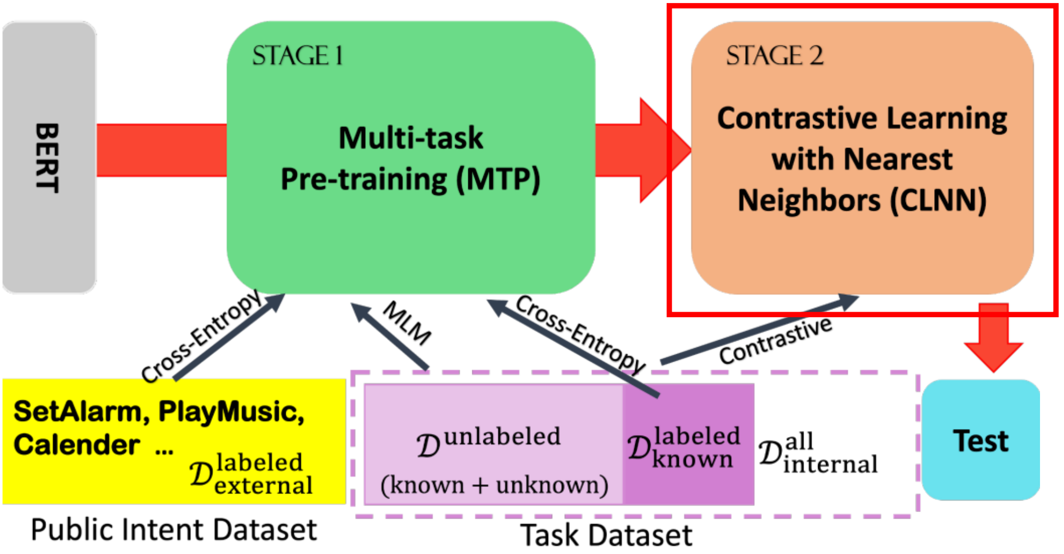
where θ are model parameters.



- MTP-CLNN employs a joint pre-training loss (From both supervised classification task and self-supervised MLM task)

Method

3. Multi-Task Pre-Training and Contrastive Learning with Nearest Neighbors (MTP-CLNN)



4) Stage 2: Contrastive Learning with Nearest Neighbors (CLNN)

- During training, a minibatch of utterances $\mathcal{B} = \{x_i\}_{i=1}^{M}$ are sampled

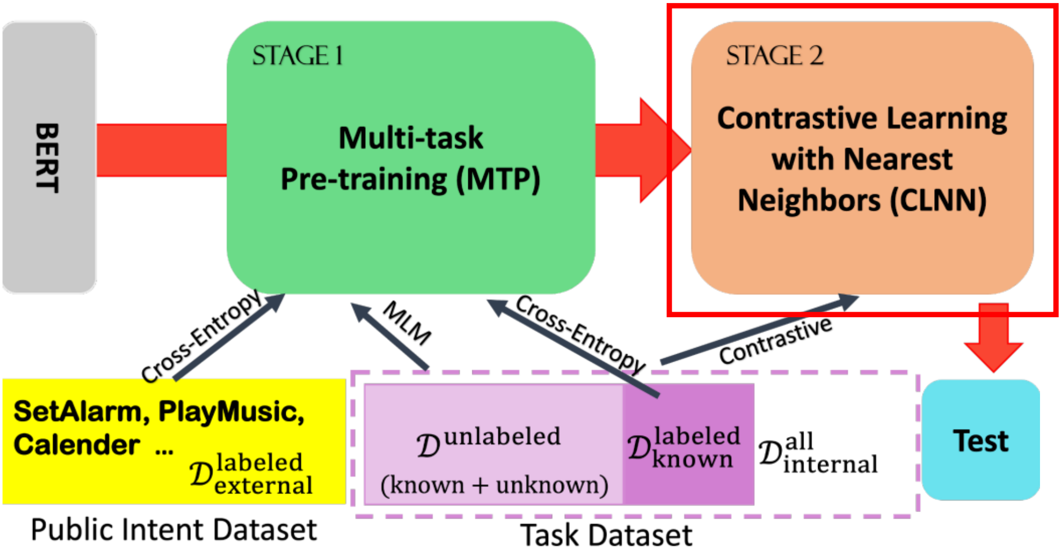
- One neighbor x_i' from its neighborhood $\mathcal{N}_i$ is sampled uniformly

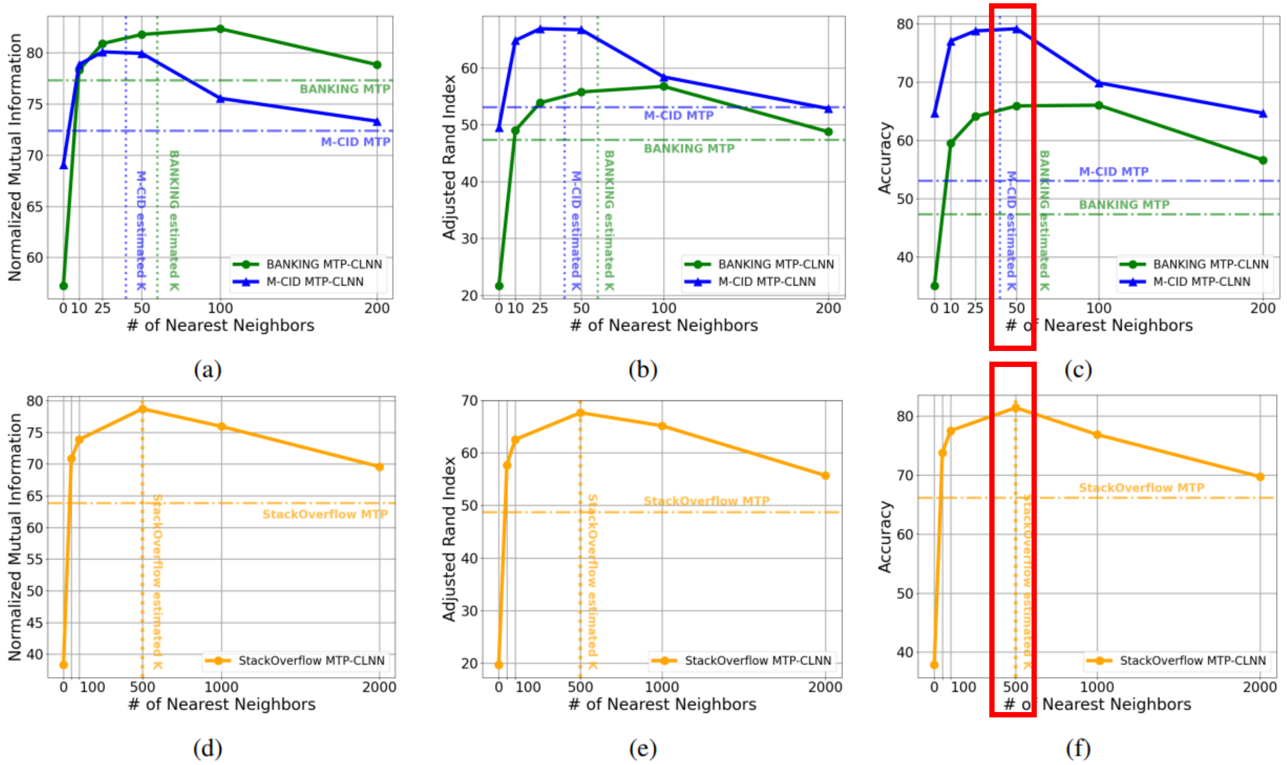
- Data augmentation is used to generate $\widetilde{x}_i$ and $\widetilde{x_i'}$ → $\mathcal{B}' = \{\widetilde{x}_i, \widetilde{x_i'}\}_{i=1}^{M}$

Method

3. Multi-Task Pre-Training and Contrastive Learning with Nearest Neighbors (MTP-CLNN)

4) Stage 2: Contrastive Learning with Nearest Neighbors (CLNN)



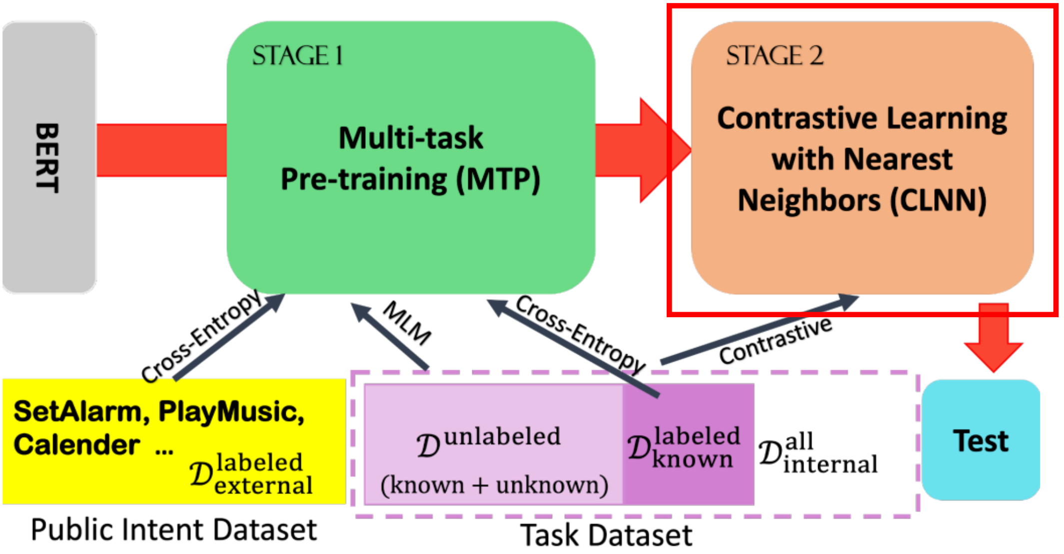


Dataset	domain	#Intents	#Utterances
CLINC150	general	120	18,000
BANKING	banking	77	13,083
StackOverflow	questions	20	20,000
M-CID	covid-19	16	1,745

Table 1: Dataset statistics.

Method

3. Multi-Task Pre-Training and Contrastive Learning with Nearest Neighbors (MTP-CLNN)



4) Stage 2: Contrastive Learning with Nearest Neighbors (CLNN)

- Contrastive loss:

$$l_i = -\frac{1}{|\mathcal{C}_i|} \sum_{j \in \mathcal{C}_i} \log \frac{\exp(\mathrm{sim}(\tilde{h}_i, \tilde{h}_j)/\tau)}{\sum_{k \neq i}^{2M} \exp(\mathrm{sim}(\tilde{h}_i, \tilde{h}_k)/\tau)}$$

$$\mathcal{L}_{\text{stg2}} = \frac{1}{2M} \sum_{i=1}^{2M} l_i,$$

where $\mathcal{C}_i \equiv \{\boldsymbol{A}'_{i,j} = 1 | j \in \{1, ..., 2M\}\}$ denotes the set of instances having positive relation with

https://arxiv.org/pdf/2205.12914.pdf

Method

3. Multi-Task Pre-Training and Contrastive Learning with Nearest Neighbors (MTP-CLNN)

5) Data augmentation

- Authors observe that the intent of an utterance can be expressed by only a small subset of words such as *"suggest restaurant"* or *"book a flight"*

- Randomly replacing a small amount of tokens in it with some random tokens from the library will not affect intent semantics much

	BANKING		StackOverflow		M-CID	
Methods	NMI	ARI	NMI	ARI	NMI	ARI
dropout	79.52	50.83	75.60	57.67	79.64	66.14
shuffle	79.02	49.72	75.70	58.95	79.68	66.09
EDA	78.29	49.02	71.50	49.80	79.73	66.39
SWR(Ours)	82.03	56.18	78.48	67.15	79.23	65.74
RTR(Ours)*	81.80	55.75	78.71	67.63	79.95	66.71

Stop-words Replacement (SWR)
Random Token Replacement (RTR)

Outline

1. Detour: Task Description
2. Method
3. Conclusion

Conclusion

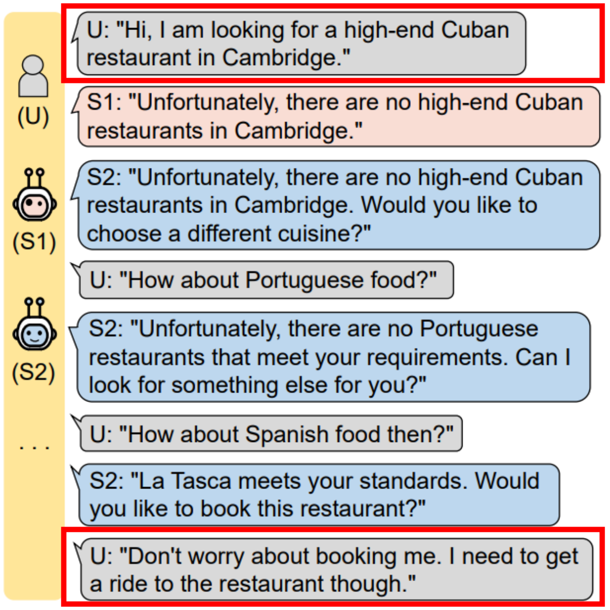
1. Any drawbacks?

1) Assumption at AV-KMEANS

- 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.

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!*
What if iOS is 10.2?
Agent 2: *Hi there! With iOS 10.3 or later, Find My iPhone can help you locate missing AirPods.*



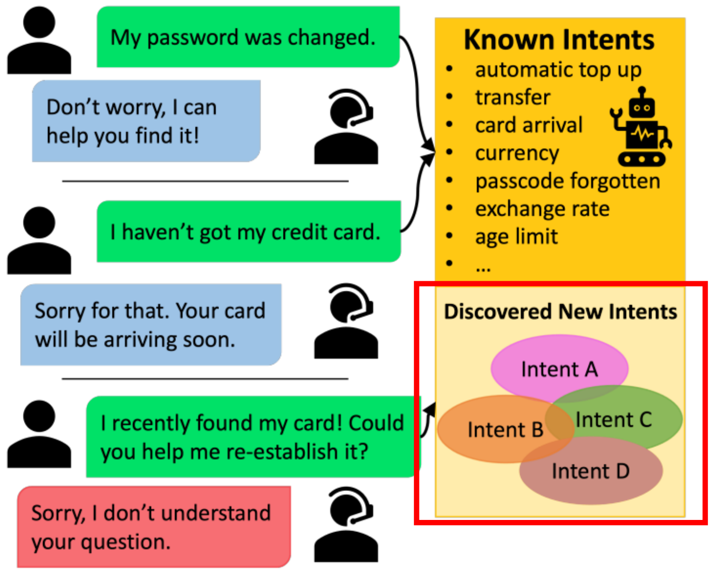
Conclusion

1. Any drawbacks?

2) Inherent limitations

- Most works are evaluated on balanced data unlike real-life scenarios

- This task can’t generate a valid intent name for each cluster, leaving semantic information behind



Thank you

https://jeiyoon.github.io/

TAVE Research