Intent Mining from past conversations for Conversational Agent

Ajay Chatterjee
Accenture Labs, India
ajay.chatt03@gmail.com

Shubhashis Sengupta
Accenture Labs, India
shubhashis.sengupta@accenture.com

Abstract

Conversational systems are of primary interest in the AI community. Organizations are increasingly using chatbot to provide round-the-clock support and to increase customer engagement. Many commercial bot building frameworks follow a standard approach that requires one to build and train an intent model to recognize user input. These frameworks require a collection of user utterances and corresponding intent to train an intent model. Collecting a substantial coverage of training data is a bottleneck in the bot building process. In cases where past conversation data is available, the cost of labeling hundreds of utterances with intent labels is time-consuming and laborious. In this paper, we present an intent discovery framework that can mine a vast amount of conversational logs and to generate labeled data sets for training intent models. We have introduced an extension to the DBSCAN (Ester et al., 1996) algorithm and presented a density-based clustering algorithm ITER-DBSCAN for unbalanced data clustering. Empirical evaluation on one conversation dataset, six intent dataset, and one short text clustering dataset show the effectiveness of our hypothesis. We release the datasets and code for future evaluation at https://github.com/ajaychatterjee/IntentMining. [1]

1 Introduction

In the past few years, there is a growing community and business interest in conversational systems (chatbots primarily). A key step towards designing a task-oriented conversational model is to identify and understand the intention of a user utterance. An intent in a conversational model maps semantically similar sentences to a high-level abstraction for a chatbot that can generate a similar response or perform an action. For example, "unable to log-in to the system", "can not log in", "facing issue during sign-in" - though linguistically different, are all interpreted as intents related to **login issue**. The current crop of bot-building frameworks require annotated data for building an Intent model. Many commercial chatbot building frameworks such as Microsoft Azure Bot Service[2], IBM Watson Assistant[3] support intent training in a supervised setting. The developers and domain experts typically consider past interaction logs between human-human or human-computer as a valuable resource and carry out an extensive manual process of intent labeling. The process of intent discovery and training data creation is by large manual and effort-intensive and carried out by domain experts.

Existing dialog corpora contains pre-defined intent and dialog state defined, and consequently, most of the work (Mrkšić et al., 2015; Henderson et al., 2014) ignores intent discovery during conversation design. Previous work (Haponchyk et al., 2018; P, 2016) on intent identification focuses on clustering single user query/ question using supervised or unsupervised clustering. But the tasks do not consider conversational data. Perkins (2019) discusses the realistic complexity of user intent space in a complex domain such as customer support and health care and use the conversational data for clustering and intent induction. But the previous works on intent discovery use pre-decided number of intents as a parameter

[2]https://azure.microsoft.com/en-us/services/bot-service/

[3]https://www.ibm.com/cloud/watson-assistant/

Proceedings of the 28th International Conference on Computational Linguistics, pages 4140–4152
Barcelona, Spain (Online), December 8-13, 2020

USER	Hi, is there any way to enable skype recording.
AGENT	Hello USER
USER	Hi
AGENT	As I understand, you need recording service to be enabled for Skype for Business.
Issue Description	User reported unable to record calls

Table 1: A sample conversation between a Customer (USER) and a Support Analyst (AGENT) along with Issue description is added by Agent after the conversations in IT Support. The analyst is trying to solve a problem related to Microsoft Skype for Business application.

to group the data. In real-world datasets, estimating the real number of intent is also challenging. To address these problems, we propose a set of data extraction methodology to extract a set of utterances from a conversation. These utterances later will be used for clustering and generation of parallel corpus for intent classification training.

To the best of our knowledge, ours is one of the first efforts to bridge the gap between - 1) research in dialog act tagging (Stolcke et al., 2000b; Kim et al., 2010; Chen et al., 2018), 2) state of the art research in Natural Language representation (Cer et al., 2018; Kiros et al., 2015; Logeswaran and Lee, 2018) and, 3) density-based clustering (Ester et al., 1996; McInnes et al., 2017) for automatically discovering intents. In this work, we describe an Intent Mining framework that reduces the labeling effort significantly by using two sources of information - the metadata/ short description about conversations and the conversations themselves (Refer to Table 1 for a sample conversation in the helpdesk scenario). In cases where raw conversations are presented without any metadata, we have experimented with different approaches to extract a suitable description for representing the summary/ short description of a conversation. We also experimented with the pre-trained language model (Universal Sentence Encoder (Cer et al., 2018)) for sentence representation. We use the textual descriptions to cluster conversations into unique groups, using a density-based clustering approach (discussed in section 3.2). Clusters are labeled to generate seed data for each intent. Features extracted from the labeled conversations along with intent labels are used to generate training data and train a statistical classifier. Unlabeled conversations are then labeled by the base classifier on the basis a cut-off confidence score of the model. The final training set can be used to train any supervised classification algorithm. We show that an intent model trained in this manner works with good efficacy and provides decent coverage of intents.

The primary contribution of the work summarized as follows :

- We present an intent discovery framework that involves 4 primary steps: 1) extraction of textual utterances from a conversation using a pre-trained domain agnostic Dialog Act Classifier (Data Extraction), 2) automatic clustering of similar user utterances (Clustering), 3) manual annotation of clusters with an intent label (Labeling) and, 4) propagation of the intent labels to the utterances from the previous step, which are not mapped to any cluster (Label Propagation) to generate intent training data from raw conversations.
- Our work present an effort to generate intent training data for raw conversations. We introduce the dialog intent mining task and present a density based clustering algorithm with novel feature extraction technique.
- The true class distribution of intents of the real world conversation data is unknown and may contain skewness. Our work presents an effort to automatically discover clusters without any prior knowledge about the intent classes.
- We study the performance of previous density based clustering algorithm in the intent discovery task. The presented algorithm, ITER-DBSCAN outperforms previous state of the art in terms of intent coverage.

2 Related Work

Intent discovery and analysis is a fundamental step to build intelligent task-oriented conversational agents. Intents are a sequence of words which are mapped to predefined categories to comprehend user request. Recent works point to two directions to build quality intent models. Re-using available conversation log, to bootstrap intent model building process (Mallinar et al., 2018; Goyal et al., 2018; Shi et al., 2018; Liu et al., 2007). The other is to allow domain experts to build an intent model by working on the model definition, labeling, and evaluation through user interfaces (Williams et al., 2015). Our work is at the intersection of these two approaches, in the sense that we mine candidate clusters in an unsupervised way and then allow domain experts to review and label the clusters (Intent Discovery).

Gathering good quality labeled data for any machine learning process is expensive. There have been significant efforts to reduce labeling effort; including work on clustering(Cheung and Li, 2012; Xu et al., 2017; Perkins and Yang, 2019), semi-supervised learning (Chapelle et al., 2010), active learning (Settles, 2012), transfer learning (Goyal et al., 2018) and also recently proposed data programming frameworks (Ratner et al., 2017; Mallinar et al., 2018). Semi-supervised, Transfer learning and active learning require seed training data for processing. Clustering is primarily used to collect the initial seed data. Most clustering algorithms fail to discover classes in a highly skewed distribution. We overcome these challenges to obtain labels on noisy data by applying a novel clustering algorithm for seed data collection and subsequently propagating labels to generate high-quality training data. Various work focus on using existing chat logs to build intent models. A transfer learning-based system has been proposed (Goyal et al., 2018) to learn from low resource settings. Data programming based (Mallinar et al., 2018) systems provide an interface to write labeling function for labeled data generation. However, one underlying assumption of using these methods is that they all require the intents to be known beforehand. This pre-condition is very difficult to meet in real-world cases.

Clustering is also an active research area for pattern mining. Popular algorithms such as centroid based clustering algorithms (K-Means (Lloyd, 1982)), density-based algorithm (DBSCAN (Ester et al., 1996)), HDBSCAN (McInnes et al., 2017)), are very useful in practical applications. Although K-Means is very fast and mostly used for clustering, it requires one to define the number of clusters as a parameter to the algorithm. Among the existing clustering approaches, a density-based algorithm particularly DBSCAN (density-based spatial clustering with noise) and its variations, is more efficient for detecting clusters with arbitrary shapes from the noisy dataset where there is no prior knowledge about the number of clusters (Ghaemi and Farnaghi, 2019; Liu et al., 2007). Many improved versions of this algorithm are also available (such as NG-DBSCAN (Lulli et al., 2016)) to overcome the scalability issues of density-based clustering, but they fail to address the ineffectiveness of density-based approaches in sparse data setting.

Although density-based clustering has limitations, it is a powerful tool for automatic data exploration and pattern mining. A key contribution of our work is to provide a better exploration strategy in unbalanced data settings. We search the feature space for different density clusters by adjusting the density definition of DBSCAN algorithm over iterations. This allows us to generate cluster with different densities and hence to find intents with low frequencies from the past chat log. Clusters are explicitly labeled by the expert to collect training data for the intent model. We apply this methodology in the publicly available intent classification dataset with highly skewed class distribution to understand the effectiveness of our clustering algorithm for intent discovery.

3 Methodology

In this section, we describe the methods used for the Intent Mining framework.

3.1 Feature Engineering

The following methods are used for extracting features from the Natural language description and conversation data .

1. **Pre-trained Sentence Embedding (USE)** : We use pre-trained sentence embedding(Universal Sentence Encoder (Cer et al., 2018)) without any fine-tuning for the downstream tasks. Here, we pass

each short description to the model[4] and extract *1-d* vector representation.

2. **Dialog Act Classifier** : Dialog Act Classifier (Stolcke et al., 2000a) is crucial to Natural Language Understanding, as it provides a general representation of speaker's intent, that is not bound to any particular dialog domain. The correct interpretation of the intent behind a speaker's utterance plays an important role in determining the success of the conversation. For example, consider these two utterances - "Book a flight for me" and "Can you book a flight". The generic intent of the first utterance is a "Command" type and where the latter is a "Question" type, and the domain dependent intent is same for both case, "book a flight". Understanding different cues of the natural language helps to generate better response. For example, for the first utterance, the dialog system can generate more human-like response, "Sure. Please wait for few minutes as I will start the booking process", whereas for the second case it can be more straight forward as "Alright. Let me start the booking process."

   In the context of our work, we use ATIS Corpus (Hemphill et al., 1990); the dataset contains textual conversations related to Air Travel Information System. Utterances in the conversations are tagged with dialog act types - "Information", "Query", "Command", "Greetings", "Confirmation-Affirmation". Natural language-based features such as part-of-speech of the tokens, bi-grams of parts-of-speech are extracted from the utterances and a sequence-based classifier (CRF (Lafferty et al., 2001)) is trained. The CRF model is configured with the following parameters - a. training algorithm: lbfgs (Zhu et al., 1997) (Gradient descent using the L-BFGS method), L2 regularization: 0.001. We train a sequence classifier using python-crfsuite[5]; to tag utterances in a dialog system with dialog act type.

3.2 Cluster & Label

DBSCAN is a density based clustering non-parametric algorithm, given a set of points, it groups points together that are closely packed (points with many nearby neighbors, high-density area) and marks points that lie alone in low-density regions (whose nearest neighbors are too far away) as outliers. The primary advantages of density-based clustering one that a) it can automatically find clusters based on the definition of density, b) it can find clusters of arbitrary shape rather than being limited to "ball-shaped" ones. We propose a variation of this algorithm in our work and the primary motivation is driven by the following two research questions -

Research Question 1 : How to group a set of data points without defining a hard bound on the number of cluster?

Research Question 2: How to automatically search for clusters with different densities in a sparse data space?

DBSCAN is a popular density-based clustering algorithm that searches for clusters broadly with two parameters - a) Maximum distance and, b) Minimum number of points. In DBSCAN literature, a point is a core point if there exists a threshold number of points within a maximum distance including the core point. All the other points are classified as Noise.

Let, $\mathbf{X}$ be a set of points $\{x_1, x_2, .., x_\mathrm{i}\}$ to be clustered and the distance between any two points is defined by $D(,)$.

Let $S(\mathbf{X})$ be a subset of $\mathbf{X}$. And, $l = D(\mathrm{p}, 0) = D(0, \mathrm{p})$ such that, l is the distance between point p and origin.
Let, $\mathcal{N}(\cdot)$ be the cardinality of a set. Let, $x_\mathrm{i}, x_\mathrm{j}$ be any two points from the set $S(\mathbf{X})$. such that,

[4]https://tfhub.dev/google/universal-sentence-encoder/4
[5]https://python-crfsuite.readthedocs.io/en/latest/

$$\forall_{\mathrm{i}}\exists_{\mathrm{j}} D(x_{\mathrm{i}}, x_{\mathrm{j}}) \leq d \tag{1}$$

$$\mathcal{N}(S(\mathbf{X})) \geq K \tag{2}$$

Where, d is the maximum distance and K is the minimum number of points, according to the definition of DBSCAN.

We formulate that, there also exists a subset $P(\mathbf{X})$ and let $x_{\mathrm{i}}, x_{\mathrm{j}}$ be any two points in it. Then,

$$\exists_{P(\mathbf{X})}\forall_{\mathrm{i}}\exists_{\mathrm{j}} D(x_{\mathrm{i}}, x_{\mathrm{j}}) \leq d + \delta d \tag{3}$$

$$\mathcal{N}(P(\mathbf{X})) \geq K^{'} \tag{4}$$

where, $K > K^{'}$. Equations 3 and 4 essentially tell that less frequent classes in the dataset can be found by increasing the distance value constraint and propose a minimum number points constraint for cluster discovery to tackle unbalanced data distribution.

We modify the DBSCAN algorithm, naming it ITER-DBSCAN, to work with datasets having imbalanced class distribution (Refer to Algorithm 1). The algorithm runs iteratively to search for clusters with high-density regions to low-density regions. The low-density region search is controlled by two parameters "max-distance"and "min-points". "max-distance" parameters controls what is maximum distance to consider two items belongs to same group and "min-points" controls what is minimum number of items in a group to qualify it as a cluster. We use cosine-distance for calculating distance between points.

Algorithm 1: ITER-DBSCAN

Input: A set of Textual utterances(data-points)
parameter: featuretransformer, initial-min-distance, initial-number-of-points, delta-min-distance, delta-number-of-points, min-points, max-iteration
Output: Data-points with cluster label

```
current-minimum-distance=initial-min-distance;
current-number-of-points=initial-number-of-points;
iteration=1;
while iter ≤ max-iteration do
    if current-number-of-points == min-points then
        break;
    end
    /* compute feature representation of the data points with the
       featuretransformer method                                  */
    feature-vector=featuretransformer(data-points) ;
    Run DBSCAN Algorithm with current-minimum-distance, current-number-of-points and
     feature-vector;
    current-data-points = get data points marked as noisy points;
    set data-points with current-data-points;
    current-minimum-distance = current-minimum-distance - delta-min-distance;
    current-number-of-points = current-number-of-points - delta-number-of-points ;
    iteration = iteration + 1 ;
end
```

Parameters: ITER-DBSCAN parameters are descried below,

- **data-points**: The primary input to the algorithm is a set of data-points (textual data) for clustering.
- **featuretransformer**: Transformer function to convert the textual data into feature representation[6].
- **initial-min-distance**: Initial distance value for creating cluster.
- **initial-number-of-points**: Initial number of points in a group for cluster validation.

[6]In this work the feature representation is referred as numerical feature.

- **delta-min-distance**: Single distance value is not enough to cluster sparse dataset, at each iteration the distance value is increased by delta-min-distance parameter to search for new cluster.
- **delta-number-of-points**: Minimum number of points parameter is decreased by delta-number-of-points parameter at each iteration for finding low density cluster.
- **min-points**: Iteration is terminated when the minimum number of points for cluster creation reaches min-points.
- **max-iteration**: max-iteration is the maximum number of times algorithm runs and updates delta-min-distance and delta-number-of-points for cluster discovery.

3.3 Label Propagation

Our clustering approach provides a set of labeled conversation-intent pair and a set of unlabeled conversations, as the density-based clustering might not group all the data points. A statistical classifier such as Logistic Regression[7] learns a mapping function between labelled conversations and intents. The trained classifier propagates the intent to the unlabelled conversations to generate a final intent classification training dataset.

3.4 Approaches for Description Extraction form Conversation

In industrial service desk scenario, the metadata or description about the conversation is added later by the service agent after the issue is resolved and may not available in many cases. In this section, we describe two methods to extract textual descriptions from the raw conversation logs which can then be fed into our clustering model -

- The agent answering to the service call always clarifies the intent with the user. Therefore, we can extract all the question asked by the Agent during the conversation with Dialog Act Classifier model (an SVM classifier created using trained Fasttext word embeddings as feature) and apply our clustering and label propagation approach to find different set of questions asked by the agent. We mark a special type of questions asked by the agent is "intent_clarfication" to clarify the intent. For example - "As I understand you need recording service to be enabled for Skype for Business" (Refer to Table 1) where Agent clarifies the request with the request with user. We can extract this sentence for Short description of the conversation.
- We can also extract top-3 user utterances of "Information" or "Question" type using Dialog Act Classifier model. This utterance set can be also used for representing a short description about the conversation. This design choice is made from the observation that a user informs about her queries in the top few messages and DAC model filters some of the unrelated utterances (such as greetings and command type) leaving behind the ones that can be used in our purpose. COMMAND type utterance removal is a special case, since in our conversation dataset users request rather than command agents for help. But in other scenario, we might need to add COMMAND type utterances for representing short description of the conversation.

4 Data

Existing task oriented dialog corpora such as MultiWOZ dataset (Budzianowski et al., 2018; Ramadan et al., 2018; Eric et al., 2019), Microsoft Dialog Dataset (Li et al., 2018; Li et al., 2016), ATIS (Hakkani-Tur et al., 2016) comprise of dialog intent defined in narrow domain like Train, Restaurant, car booking. Recently, Perkins (2019) published a curated complex human-human conversation dataset gathered from Twitter airline customer support. The tweets comprise conversations between customer support agents of some airline companies and their customers. The conversations constitutes various topics for intent mining task. We also consider various open-source short text dataset and evaluate the generalization of our algorithm. In , we present the overview of the datasets. Table 2 also presents the intent distribution i.e., the maximum and the minimum number of utterances pertaining to one intent.

[7]Other statistical classifier or neural network-based model might provide better accuracy, but this part is out of the scope of our current research

Dataset	Utterances	Intents	Max Intent count	Min Intent count
Airlines Twitter Conversation	491	14	107	10
FinanceData	10003	77	187	35
AskUbuntuCorpus	162	5	57	8
ChatbotCorpus	206	2	128	78
WebApplicationCorpus	88	7	23	5
ATIS	4972	17	3666	6
Personal Assistant	25312	64	1218	171
Stackoverflow Dataset	20000	20	1000	1000

Table 2: Overview of the datasets used in the evaluation.

4.1 Conversation Dataset

The Airlines Twitter Conversation dataset (Perkins and Yang, 2019) is a human-human conversation dataset[8] related to user queries posted on various topics on Twitter about airline-related travel. We extract a short description from these conversations, discussed in section 3.4.

4.2 Dialog Intent and Short Text Dataset

The Finance dataset (Casanueva et al., 2020) contains various online banking queries annotated with their corresponding intents[9] publised by PolyAI team. The AskUbuntuCorpus, ChatbotCorpus and WebApplicationCorpus, this three corpora (Braun et al., 2017) collected from StackExchange and Telegram Chatbot contains utterances with intent labels[10]. The ATIS dataset[11], which provides large number of messages and their associated intents, is useful for intent discovery/ classification problems including chatbots. Personal Assistant is another large scale dataset (Xingkun Liu and Rieser, 2019) consisting of messages posted by a personal assistant. The dataset[12] contains 25K+ messages with 64 intent label. Stackoverflow Dataset[13] is a short text dataset used for classification and clustering of extracted queries from StackOverflow website.

5 Experiments

In this section, we evaluate ITER-DBSCAN on the 8 datasets discussed in Section 4. We compare ITER-DBSCAN algorithm with state-of-the-art density-based clustering algorithms such as DBSCAN and HDBSCAN. For the conversation dataset, we also evaluate our results with a recently published multi-view clustering, AV-KMeans Algorithm. We use USE sentence embedding method to convert the natural language to numerical feature.

5.1 Metrics

We use standard clustering metrics to evaluate the algorithms. We adapt metrics such as precision, recall, F1 score, and unsupervised clustering accuracy from the work of Perkins (2019) to compare our results for the conversation datasets. To evaluate the short text datasets, we primarily use two metrics: a) Normalized Mutual Information (Vinh et al., 2009), b) Adjusted Rand Index (Hubert and Arabie, 1985).

Normalized Mutual Information(NMI): NMI is designed as a combined measure for the accuracy of clustering and the total number of clusters. NMI is an entropy based metric that computes the amount of common information between two partitions -

$$\text{NMI} = \frac{2 * I(Y;C)}{H(Y) + H(C)} \tag{5}$$

where, Y is class labels, C is cluster labels, H(.) is Entropy and I(Y;C) is Mutual information between Y and C.

[8]https://github.com/asappresearch/dialog-intent-induction
[9]https://github.com/PolyAI-LDN/task-specific-datasets
[10]https://github.com/sebischair/NLU-Evaluation-Corpora
[11]https://www.kaggle.com/hassanamin/atis-airlinetravelinformationsystem
[12]https://github.com/xliuhw/NLU-Evaluation-Data
[13]https://github.com/jacoxu/StackOverflow

Adjusted Rand Index(ARI): ARI computes a similarity measure between two clusterings by considering all pairs of samples and counting pairs that are assigned in the same or different clusters in the predicted and true clusterings. ARI is an improved version of Rand Index, which is defined as follows:

$$\text{ARI} = \frac{\sum_{ij}\binom{n_{ij}}{2} - \left[\sum_i \binom{a_i}{2}\sum_j \binom{b_j}{2}\right] / \binom{n}{2}}{\frac{1}{2}\left[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}\right] - \left[\sum_i \binom{a_i}{2}\sum_j \binom{a_j}{2}\right] / \binom{n}{2}} \tag{6}$$

The range of AMI and NMI is from 0 to 1, a larger value indicates a higher agreement between ground truth and the predicted partition for the dataset.

5.2 Parameter Settings

We evaluate DBSCAN, HDBSCAN and ITER-DBSCAN on a wide variety of parameters. We use the scikit-learn implementation of DBSCAN[14] and HDBSCAN[15]. We generate a wide range based on the combination of the parameters described below. All the other parameters are kept at default according to the implementation. We use cosine distance function for evaluating the clustering algorithms. To evaluate the results better, we keep the minimum cluster size as 3 for all the density based clustering algorithm.

- **DBSCAN:** For evaluaton of DBSCAN, we select the minimum distance parameter between a range 0.09 to 0.40 with a change of 0.01 (Example: [0.09, 0.10, 0.11, .., 0.40]). We also configure the minimum sample parameter between a range 3 to 20 with a change of 1.

- **HDSCAN:** We configure the minimum cluster size parameter between a range of 3 to 15 with a change of 1. We set the minimum samples parameter between a range 3 to 15 with a change of 1.

- **ITER-DBSCAN**: We set a range of values for three parameters of ITER-DBSCAN for evaluation. We use five initial distance value 0.09, 0.12, 0.15, 0.20, 0.30. We set the maximum iteration and initial minimum sample parameters between a range 10 to 15 and 10 to 25 respectively, with a change step of 1. We keep the other parameters constant - such as, delta-min-distance as 0.01, delta-number-of-points as 1, minimum points as 3.

5.3 Results

In Table 3, we present the evaluation of our algorithm on Twitter airline conversation dataset(TwACS) with DBSCAN and AV-KMeans. We use 4 evaluation metrics adapted from the work of Perkins (2019). The HDBSCAN algorithm did not find any clusters for this dataset, hence not reported. We evaluate the dataset with the top-3 utterances extracted from conversation, with (and without) dialog act classifier based feature extraction. We report the effectiveness of feature extraction methodology in Table 3.

In Table 4, we present our algorithm ITER-DBSCAN results as a comparison to DBSCAN and HDBSCAN. For each dataset, we describe the total number of intents and the number of intents the algorithm identified. We also present the Normalized Mutual information score and Adjusted Rand Score for clustering evaluation. In most of the task, our algorithm achieves state-of-the-art results on the intent discovery and different clustering metrics.

5.4 Parameter Study

In this section, we study the growth of number of clusters to identify different number of intents. We present the result of all datasets in figure 1. We plot the change of intent counts in x axis and the change

[14]https://scikit-learn.org/stable/modules/generated/sklearn.cluster.DBSCAN.html
[15]https://github.com/scikit-learn-contrib/hdbscan

Corpus	Algorithm	Precision	Recall	F1	ACC
TwACS	DBSCAN	31.8	**65.4**	42.8	31.8
	AV-KMEANS	**48.9**	43.8	46.2	39.9
	ITER-DBSCAN (without feature extraction)	42.7	48.3	47.4	37.5
	ITER-DBSCAN (with feature extraction)	48.5	54.4	**51.3**	**48.5**

Table 3: Experiment result of Twitter airline conversation dataset

CorpusName	Algorithm	Total Intents	Intents Found	NMI	ARI
AskUbuntuCorpus	DBSCAN	5	3	0.35	0.23
	HDBSCAN	5	4	0.44	0.33
	ITER-DBSCAN	5	4	**0.51**	**0.45**
ATIS	DBSCAN	17	13	0.28	0.26
	HDBSCAN	17	11	0.24	0.24
	ITER-DBSCAN	17	14	**0.55**	**0.66**
ChatbotCorpus	DBSCAN	2	2	0.59	0.61
	HDBSCAN	2	2	0.59	0.61
	ITER-DBSCAN	2	2	**0.63**	**0.68**
FinanceData	DBSCAN	77	76	0.44	0.17
	HDBSCAN	77	75	0.53	0.31
	ITER-DBSCAN	77	77	**0.79**	**0.6**
PersonalAssistant	DBSCAN	64	64	0.45	0.25
	HDBSCAN	64	64	0.64	0.48
	ITER-DBSCAN	64	64	**0.65**	**0.5**
Stackoverflow	DBSCAN	20	20	0.48	0.34
	HDBSCAN	20	20	**0.72**	0.62
	ITER-DBSCAN	20	20	0.71	**0.63**
WebApplicationsCorpus	DBSCAN	7	4	0.33	0.22
	HDBSCAN	7	4	0.32	0.23
	ITER-DBSCAN	7	5	**0.45**	**0.39**

Table 4: Experiment result of Intent and short text clustering datasets.

of number of clusters in y axis. We also study the effect of parameter configurations for the Finance Dataset in figure 2, and how it changes the number of intents and clusters. In x-axis of the figure 2, we plot the difference between initial minimum distance and maximum iteration which can be regarded as the minimum possible cluster size. In y-axis of the figure 2, we plot the number of intents (left) and number of clusters (right). In figure 2, we lay the change of intent count with respect to different initial distance. The plot also shows that the number of clusters decreases as the difference between as minimum cluster size increases and initial distance between 0.12 to 0.20 provides better stability in discovering intents. So, in practice we can use this two parameters to balance between the number of clusters and the coverage of the intents.

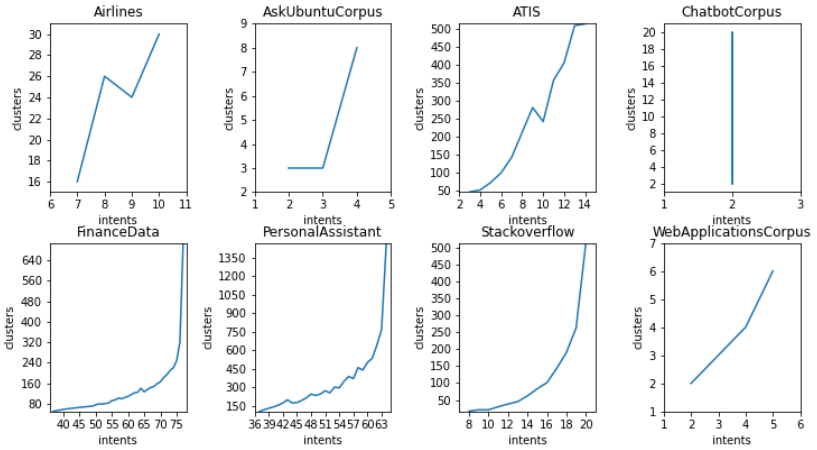

Figure 1: Growth of number of clusters with respect to intent.

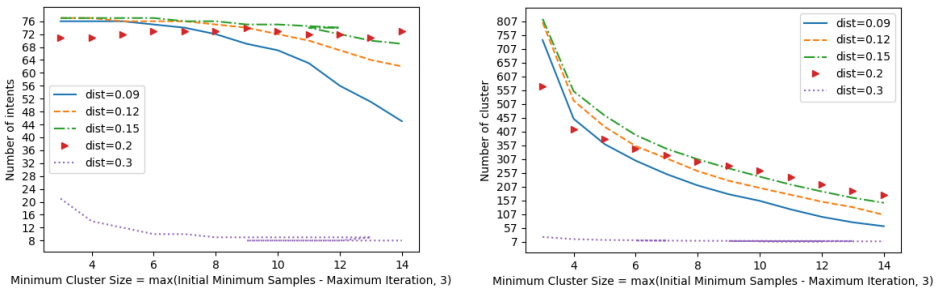

Figure 2: Change of Intent and cluster count with respect to different parameter configuration.

5.5 Implantation Details and Time Cost Analysis

Density based clustering algorithm uses local topological structure to create meaningful clusters. Time cost increases with data dimension and number of points (Gan and Tao, 2015). To overcome this complexity, we partition large dataset into distinct subsets and apply our algorithm to this subsets in parallel. We use K-Means clustering for generating subsets of data when the dataset size is more than 10K. We use the following function (5) - to set the number of clusters for K-Means algorithm :

$$NumberOfCluster = Max(data\_size/10000, 3) \tag{7}$$

In figure 3, we plot the time taken by density-based clustering algorithms to process the datasets of section 4. Due to parallelization, the time complexity of our algorithm becomes almost linear with dataset size (after reaching volume of 10K).

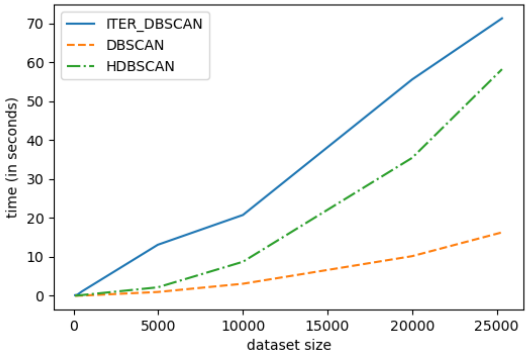

Figure 3: Time complexity analysis of algorithms.

6 Conclusion and Feature Work

In this work, we presented a framework that can cluster textual data using a state-of-art sentence representation method with our algorithm. That provides better intent discovery for complex conversation datasets and short text datasets. Our algorithms are able to identify intents from imbalanced dataset with greater accuracy than previous state of the art algorithms. We also presented a feature extraction method using Dialog Act Classification model to extract a short description from conversations for intent mining task.

In future, we would like to extend our work by incorporating various other features from conversations, such as different form of questions asked by the agent to resolve a functional task and other linguistic features, for improved clustering. We would also like to explore the use of the neural network to learn generic conversation representation from chat logs for better feature representation.

References

Daniel Braun, Adrian Hernandez-Mendez, Florian Matthes, and Manfred Langen. 2017. Evaluating natural language understanding services for conversational question answering systems. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 174–185, Saarbrücken, Germany, August. Association for Computational Linguistics.

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

Iñigo Casanueva, Tadas Temcinas, Daniela Gerz, Matthew Henderson, and Ivan Vulic. 2020. Efficient intent detection with dual sentence encoders. In *Proceedings of the 2nd Workshop on NLP for ConvAI - ACL 2020*, mar. Data available at https://github.com/PolyAI-LDN/task-specific-datasets.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. 2018. Universal sentence encoder. *CoRR*, abs/1803.11175.

Olivier Chapelle, Bernhard Schlkopf, and Alexander Zien. 2010. *Semi-Supervised Learning*. The MIT Press, 1st edition.

Zheqian Chen, Rongqin Yang, Zhou Zhao, Deng Cai, and Xiaofei He. 2018. Dialogue act recognition via crf-attentive structured network. In *The 41st International ACM SIGIR Conference on Research Development in Information Retrieval*, SIGIR '18, page 225–234, New York, NY, USA. Association for Computing Machinery.

Jackie Chi Kit Cheung and Xiao Li. 2012. Sequence clustering and labeling for unsupervised query intent discovery. In *Proceedings of the Fifth ACM International Conference on Web Search and Data Mining*, WSDM '12, page 383–392, New York, NY, USA. Association for Computing Machinery.

Mihail Eric, Rahul Goel, Shachi Paul, Abhishek Sethi, Sanchit Agarwal, Shuyag Gao, and Dilek Hakkani-Tur. 2019. Multiwoz 2.1: Multi-domain dialogue state corrections and state tracking baselines. *arXiv preprint arXiv:1907.01669*.

Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. 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*, KDD'96, page 226–231. AAAI Press.

Junhao Gan and Yufei Tao. 2015. Dbscan revisited: Mis-claim, un-fixability, and approximation. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, SIGMOD '15, page 519–530, New York, NY, USA. Association for Computing Machinery.

Zeinab Ghaemi and Mahdi Farnaghi. 2019. A varied density-based clustering approach for event detection from heterogeneous twitter data. *ISPRS International Journal of Geo-Information*, 8:82, 02.

Anuj Kumar Goyal, Angeliki Metallinou, and Spyros Matsoukas. 2018. Fast and scalable expansion of natural language understanding functionality for intelligent agents. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 3 (Industry Papers)*, pages 145–152, New Orleans - Louisiana, June. Association for Computational Linguistics.

Dilek Hakkani-Tur, Gokhan Tur, Asli Celikyilmaz, Yun-Nung Chen, Jianfeng Gao, Li Deng, and Ye-Yi Wang. 2016. Multi-domain joint semantic frame parsing using bi-directional rnn-lstm. In *Proceedings of Interspeech*.

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 the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2310–2321, Brussels, Belgium, October-November. Association for Computational Linguistics.

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*, HLT '90, pages 96–101, Stroudsburg, PA, USA. Association for Computational Linguistics.

Matthew Henderson, Blaise Thomson, and Steve J. Young. 2014. Word-based dialog state tracking with recurrent neural networks. In *SIGDIAL Conference*.

Lawrence Hubert and Phipps Arabie. 1985. Comparing partitions. *Journal of Classification*, 2(1):193–218, Dec.

Su Nam Kim, Lawrence Cavedon, and Timothy Baldwin. 2010. Classifying dialogue acts in one-on-one live chats. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 862–871, Cambridge, MA, October. Association for Computational Linguistics.

Ryan Kiros, Yukun Zhu, Ruslan Salakhutdinov, Richard S. Zemel, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. 2015. Skip-thought vectors. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'15, page 3294–3302, Cambridge, MA, USA. MIT Press.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pages 282–289, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Xiujun Li, Zachary C Lipton, Bhuwan Dhingra, Lihong Li, Jianfeng Gao, and Yun-Nung Chen. 2016. A user simulator for task-completion dialogues. *arXiv preprint arXiv:1612.05688*.

Xiujun Li, Sarah Panda, Jingjing Liu, and Jianfeng Gao. 2018. Microsoft dialogue challenge: Building end-to-end task-completion dialogue systems. *arXiv preprint arXiv:1807.11125*.

P. Liu, D. Zhou, and N. Wu. 2007. Vdbscan: Varied density based spatial clustering of applications with noise. In *2007 International Conference on Service Systems and Service Management*, pages 1–4, June.

Stuart P. Lloyd. 1982. Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28:129–137.

Lajanugen Logeswaran and Honglak Lee. 2018. An efficient framework for learning sentence representations. *CoRR*, abs/1803.02893.

Alessandro Lulli, Matteo Dell'Amico, Pietro Michiardi, and Laura Ricci. 2016. Ng-dbscan: Scalable density-based clustering for arbitrary data. *Proc. VLDB Endow.*, 10(3):157–168, November.

Neil Mallinar, Abhishek Shah, Rajendra Ugrani, Ayush Gupta, Manikandan Gurusankar, Tin Kam Ho, Qinghan Liao, Yunfeng Zhang, Rachel K. E. Bellamy, Robert Yates, Chris Desmarais, and Blake McGregor. 2018. Bootstrapping conversational agents with weak supervision. *CoRR*, abs/1812.06176.

Leland McInnes, John Healy, and Steve Astels. 2017. hdbscan: Hierarchical density based clustering. 2(11), mar.

Nikola Mrkšić, Diarmuid Ó Séaghdha, Blaise Thomson, Milica Gašić, 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 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 794–799, Beijing, China, July. Association for Computational Linguistics.

Deepak P. 2016. MixKMeans: Clustering question-answer archives. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1576–1585, Austin, Texas, November. Association for Computational Linguistics.

Hugh Perkins and Yi Yang. 2019. Dialog intent induction with deep multi-view clustering. 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)*, pages 4016–4025, Hong Kong, China, November. Association for Computational Linguistics.

Osman Ramadan, Paweł Budzianowski, and Milica Gasic. 2018. Large-scale multi-domain belief tracking with knowledge sharing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, volume 2, pages 432–437.

Alexander Ratner, Stephen H. Bach, Henry Ehrenberg, Jason Fries, Sen Wu, and Christopher Ré. 2017. Snorkel: Rapid training data creation with weak supervision. *Proc. VLDB Endow.*, 11(3):269–282, November.

Burr Settles. 2012. *Active Learning*. Morgan & Claypool Publishers.

Chen Shi, Qi Chen, Lei Sha, Sujian Li, Xu Sun, Houfeng Wang, and Lintao Zhang. 2018. Auto-dialabel: Labeling dialogue data with unsupervised learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 684–689, Brussels, Belgium, October-November. Association for Computational Linguistics.

Andreas Stolcke, Noah Coccaro, Rebecca Bates, Paul Taylor, Carol Van Ess-Dykema, Klaus Ries, Elizabeth Shriberg, Daniel Jurafsky, Rachel Martin, and Marie Meteer. 2000a. Dialogue act modeling for automatic tagging and recognition of conversational speech. *Comput. Linguist.*, 26(3):339–373, September.

Andreas Stolcke, Klaus Ries, Noah Coccaro, Elizabeth Shriberg, Rebecca Bates, Daniel Jurafsky, Paul Taylor, Rachel Martin, Carol Van Ess-Dykema, and Marie Meteer. 2000b. Dialogue act modeling for automatic tagging and recognition of conversational speech. *Computational Linguistics*, 26(3):339–374.

Nguyen Xuan Vinh, Julien Epps, and James Bailey. 2009. Information theoretic measures for clusterings comparison: Is a correction for chance necessary? In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, page 1073–1080, New York, NY, USA. Association for Computing Machinery.

Jason Williams, Nobal B. Niraula, Pradeep Dasigi, Aparna Lakshmiratan, Carlos Garcia Jurado Suarez, Mouni Reddy, and Geoffrey Zweig. 2015. Rapidly scaling dialog systems with interactive learning. January.

Pawel Swietojanski Xingkun Liu, Arash Eshghi and Verena Rieser. 2019. Benchmarking natural language understanding services for building conversational agents. In *Proceedings of the Tenth International Workshop on Spoken Dialogue Systems Technology (IWSDS)*, pages xxx–xxx, Ortigia, Siracusa (SR), Italy, April. Springer.

Jiaming Xu, Bo Xu, Peng Wang, Suncong Zheng, Guanhua Tian, Jun Zhao, and Bo Xu. 2017. Self-taught convolutional neural networks for short text clustering. *Neural Networks*, 88:22–31, Apr.

Ciyou Zhu, Richard H. Byrd, Peihuang Lu, and Jorge Nocedal. 1997. Algorithm 778: L-bfgs-b: Fortran subroutines for large-scale bound-constrained optimization. *ACM Trans. Math. Softw.*, 23(4):550–560, December.