Novel Feature Discovery for Task-Oriented Dialog Systems

Vinh Thinh Ho
Amazon Alexa AI
Berlin, Germany
hovnh@amazon.com

Mohamed Soliman
Amazon Alexa AI
Aachen, Germany
mohsol@amazon.com

Abdalghani Abujabal
Amazon Alexa AI
Aachen, Germany
abujabaa@amazon.com

Abstract

A novel feature represents a cluster of semantically equivalent novel user requests e.g., requests to play a song on a service or reading user's messages. Detecting and supporting novel features is crucial towards wider adoption of dialog systems by end users. Intuitively, features are represented by a combination of intents, slot types and/or their values. For example, while playing a song is a feature represented by a single intent (`PlayMusic`) only, playing a song on a service is another feature represented by the combination of `PlayMusic` intent and `ServiceName` slot type. Prior work on novelty detection limits the scope of features to those represented by novel single intents, leading to (1) giant clusters spanning several user-perceived fine-grained features belonging to the same intent, (2) incoherent interpretation of clusters from users' perspective (no direct connection to some user-perceived feature), and (3) missing those features spanning several intents. In this work, we introduce *feature discovery* as opposed to single intent discovery, which aims at discovering novel features spanning a combination of intents and slots, and present a technique for discovering novel features from user utterances. Experiments on two datasets demonstrate the effectiveness of our approach and consistently show its ability to detect novel features.

1 Introduction

Advances in Natural Language Understanding (NLU) have led to accelerated adoption of dialog systems such as Apple Siri and Amazon Alexa by end users. Standing at the core of a dialog system is an NLU model for parsing and understanding user utterances. Two of the key tasks of an NLU model are (1) Intent Classification, which classifies an utterance into a fixed set of intent labels, and (2) Slot Labeling, which classifies slot values into a predefined set of slot types (Weld et al., 2023). Determining the intent guides the dialog system to perform the proper actions as response to user's utterance. For example, user utterances *"play some music"* and *"play despacito"* express the intent `PlayMusic`, while *"how is the weather?"* and *"is it raining today"* express the intent `GetWeather`. Intents can be further grouped into domains, for instance, `PlayMusic` and `RateSong` intents belong to `Music` domain. Detecting slots and their corresponding values within an utterance gives information about objects upon which the actions should be performed. For example, *'despacito'* in *"play despacito"* is of type `SongName`.

A feature represents a user *experience* with the dialog system, for example, playing a song on a service or reading user's messages. Over time, users build up expectations about the features/experiences that the dialog system offers. Unsupported features cause friction and degrade user's experience. In terms of NLU, a novel feature could be mapped to a new combination of domain(s), intent(s), slot(s) and/or their values, where each is not necessarily novel. For example, while `PlayMusic` intent was seen by the dialog system, the combination of `PlayMusic` and `ServiceName` is never seen before, causing friction with the NLU model when parsing utterances like *"play despacito on spotify"*. Consequently, it becomes crucial to discover such features that are frequently requested by the users but are still unsupported by the NLU model, which we address in this work.

The task of novel intent discovery has been intensively studied in prior work, by harnessing unsupervised techniques (e.g., Liu et al., 2021) or semi-supervised methods for incorporating existing knowledge from labeled data (e.g., Vedula et al., 2020a; Lin et al., 2020; Zhang et al., 2021). Existing work limits the scope of novel features to those represented by novel single intents. However, this does not cover all types of features that naturally span several domains, intents, slots and their values. Consider the following user utterances, *"play* *despacito"*, *"play despacito on spotify"* and *"play despacito in 30 minutes"*. While the utterances belong to the same intent `PlayMusic`, handling each of them is different. The first requires the dialog system to play a song on the device itself, the second asks for playing the same song on a specific service, namely Spotify, while the third requires playing the song after a certain amount of time is elapsed. Moreover, each request corresponds to a different user experience and requires a different underlying implementation for responding. Assuming `PlayMusic` intent is novel, applying standard novel intent discovery will group the three utterances in a single cluster, which creates two issues: (1) having many different user-perceived experiences within the same intent (play song on device, on specific service or after some time), results in a giant cluster that needs manual inspection to be decomposed into smaller sub-groups to make meaningful business decisions, and (2) while the cluster represents a novel intent, it might not correspond to a user perceived experience – what the end users are looking for. On the other hand, assuming `PlayMusic` intent is not novel, intent discovery might miss those features at more fine-grained levels. For instance, playing a song on a service might not be detected as novel. Additionally, intent discovery cannot handle those features spanning several intents.

To allow for general feature discovery, and close the gap between user requests and the underlying models, we define a *feature* to be any combination of domain(s), intent(s) and slot(s) and/or their value(s) and move towards discovering clusters of utterances with novel feature definitions rather than only focusing on novel single intents. Novel intent discovery can be seen as a special case of feature discovery, where new features correspond to new intents unseen before.

In this paper, we present DNF (Discovery of Novel Features), a semi-supervised approach for discovering novel features from a given set of user utterances with respect to an underlying NLU model. Our method consists of a cascaded system with two steps: feature clustering and novelty detection. First, we employ state-of-the-art language model BERT (Devlin et al., 2019) with multi-stage fine-tuning to produce feature/experience-aware representations of user utterances. Then, user utterances are clustered into features. Second, we classify each resulting feature cluster as either novel or already supported by the NLU model. The salient contributions of our paper are:

- We introduce *Feature Discovery*, where, given a set of user utterances and a trained natural language understanding model, we extract clusters of novel features.
- We present DNF, an approach for discovering novel features from user utterances.
- We conducted extensive experiments on two datasets, the SNIPS dataset augmented with feature labels, and our internal real-world dataset. Experimental results demonstrate the effectiveness of our method across the two datasets.

2 Related Work

Intent classification and slot labeling are two fundamental tasks in spoken language understanding, dating back to early 90's (Price, 1990). With the rise of task-oriented dialog systems, the two tasks have seen more attention, and progress has been made by applying various deep learning approaches (e.g., Abujabal and Gaspers, 2019; Abujabal et al., 2021; Goo et al., 2018; Jolly et al., 2020; Mesnil et al., 2013).

Discovering novel domains and intents from a pool of user utterances has been well addressed in earlier works, with fully unsupervised and semi-supervised settings. These include clustering user utterances with novel domains or intents individually, using various techniques such as constrained deep adaptive clustering (Lin et al., 2020), deep aligned clustering (Zhang et al., 2021), contrastive learning (Gao et al., 2021), capsule network (Liu et al., 2019; Xia et al., 2018), open intent extraction (Vedula et al., 2020b), and others (e.g., Lin and Xu, 2019; Shivakumar et al., 2019; Yan et al., 2020; Kim and Kim, 2018). Alternatively, Vedula et al., 2020a explore the joint discovery of domains and intents, using hierarchical linking to form an intent-domain taxonomy. The task is also performed jointly with slot filling in recent works (Wang et al., 2018; Goo et al., 2018; Kim et al., 2017; Castellucci et al., 2019; Gangadharaiah and Narayanaswamy, 2019; Liu and Lane, 2016). All of the above works require a small amount of labeled data as prior knowledge to guide the discovery process.

The most prominent work for detecting novel intents without any prior knowledge was proposed by Liu et al. (2021), which employs a pre-trained network for generating sentence embeddings, and

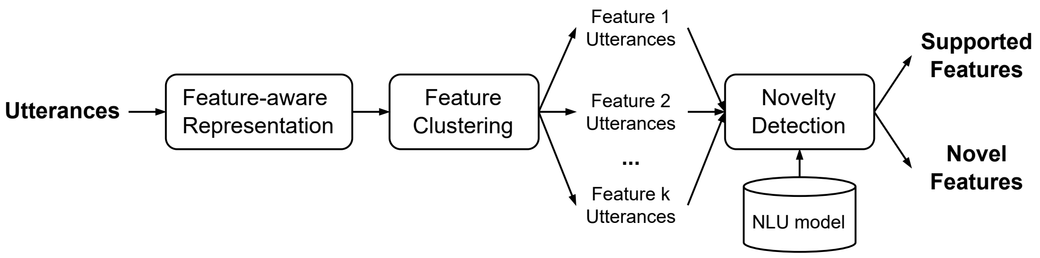

Figure 1: DNF overview with feature-aware fine-tuning, feature clustering and novelty detection.

K-Means for intent clustering. However, due to the limited supervision, the method is shown to perform poorly on novel domains.

All of the above methods are limited to a discovery objective at intent level, and thus fail to operate on more fine-grained levels (e.g., slot type and/or value) within the same intent. Moreover, they fail to detect features composed of multiple domains, intents and/or slots.

3 Methodology

Given a set of user utterances $\{u_1, u_2, ..., u_n\}$, we aim to detect a set of clusters $\{C_1, ..., C_m\}$ where each C_j is a cluster of utterances pertaining to a novel feature and m is the total number of novel features. As depicted in Figure 1, our method consists of two components, feature clustering and novelty detection. First, we assign each utterance u_i a feature label and eventually produce a set of feature-labeled utterances $(u_1, f_1), ..., (u_n, f_k)$, where $f_1, f_2, ..., f_k$ is the list of k unique features. This is performed by employing an utterance representation model specifically trained with both feature-labeled and -unlabeled data to project the input utterances into a feature-aware vector space that helps clustering the input utterances into k features. Second, a feature novelty detection model is used to classify each of the k feature clusters as either novel or already supported by the dialog system. We measure the novelty of a feature by exploiting signals from the NLU model. With DNF being a semi-supervised technique, we distinguish between two types of training data supervision: *feature-labeled* and *feature-unlabeled* utterances:

- Feature-labeled training data ($Train_L$): each utterance is annotated with its feature label f along with its intent label I and slot labels $S = \{s_1, s_2, ...\}$ where each s_i is a pair of slot type and its value in the utterance such as `SongName:despacito` and `ServiceName:spotify`.

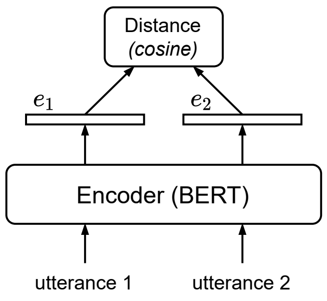

Figure 2: Fine-tuning with utterance similarity.

- Feature-unlabeled training data ($Train_U$): each utterance is only annotated with its intent and slot labels, however, with no feature label. In comparison to feature-labeled data, such data can be obtained in larger quantities and helps the feature discovery process as we exploit information about utterances' intent and slots.

3.1 Feature-Aware Utterance Representation

To first encode utterances as high-dimensional vectors separable in the feature space, we use a representation model specifically adapted for feature awareness. To this end, we employ the state-of-the-art language model SBERT – Sentence BERT (Reimers and Gurevych, 2019), which is a BERT model pretrained for generating sentence embeddings. Specifically, by feeding an utterance u into SBERT, we get a list of token embeddings $[CLS, t_1, t_2, ..., t_m]$, where CLS is the classification token. By applying mean-pooling, we obtain the representation vector of u:

$$e_u = mean\text{-}pooling([CLS, t_1, t_2, ..., t_m])$$

We transfer feature knowledge into the utterance representation model through a multi-stage fine-tuning process as described below.

3.1.1 Utterance Similarity

As depicted in Figure 2, we use a Siamese Neural Network (SNN) training paradigm for transferring feature knowledge into the model. The intuition

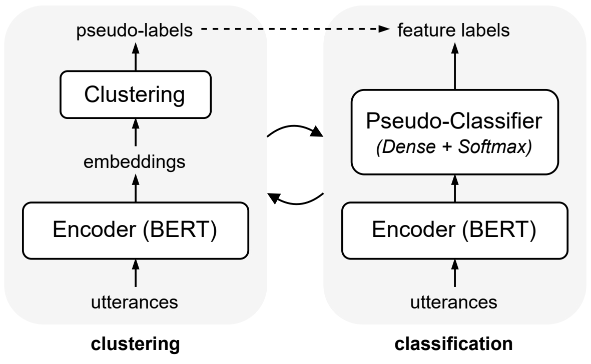

Figure 3: Fine-tuning with pseudo-classification.

behind this fine-tuning step is to directly optimize the utterance representations based on their feature similarity. In particular, a pair of utterances (u_i, u_j) is encoded into its respective embedding vectors e_i and e_j. The utterance feature similarity is computed as the cosine distance between the two vectors, and is optimized towards minimizing the distance between utterances belonging to the same feature. As our training data are both feature-labeled ($Train_L$) and feature-unlabeled ($Train_U$), we consider three kinds of training samples for this fine-tuning step:

1. Both u_i and u_j are sampled from the same $Train_L$ feature: (u_i, u_j) is a positive sample.
2. Both u_i and u_j are sampled from $Train_L$, but are from different features: (u_i, u_j) is a negative sample.
3. u_i is sampled from $Train_L$ and u_j is sampled from $Train_U$: (u_i, u_j) is a negative sample.

For case 3, it is possible that the training sample is a false-negative, since the actual feature-label of u_j could be the same as of u_i. However, we hypothesize that the number of such false-negative utterance pairs is much lower than the true-negative pairs given the large size of this training set.

As the number of utterance pairs is quadratic, we use a simple way to build a random dataset for each training epoch. For each utterance u_i in $Train_L$, we randomly sample an utterance u_j that belongs to the same feature as a positive sample, and k negative samples from both $Train_L$ and $Train_U$; hence maintaining the ratio of $1 : k$ between the number of positive and negative pairs. Empirically, we found that setting $k = 3$ provides a decent balance between positive and negative pairs.

3.1.2 Pseudo Classification

Utterance similarity considers pairwise distances between utterances, without setting global constraints across all utterances. Moreover, it optimizes the distances according to the feature-labeled clusters in $Train_L$ but does not consider the unknown feature clusters in $Train_U$. The second fine-tuning step, shown in Figure 3, aims to overcome the above issues. Inspired by the DeepCluster work (Caron et al., 2018), pseudo classification is a semi-supervised iterative training process, alternating between *clustering* and *classification*.

In the *clustering* step, we first encode the training utterances into their representation vectors. Then a clustering algorithm is used to group the representation vectors into clusters while assigning a pseudo-label to each cluster. We use COP-K-Means (Wagstaff et al., 2001) as the clustering algorithm – an extension of K-Means that allows putting constraints on the clustering process. For example, which utterances must be grouped in the same clusters, and which must not. In our case, utterances belong to the same feature cluster inside $Train_L$ must be grouped in the same candidate cluster given by COP-K-Means.

In the *classification* step, we fine-tune the BERT encoder by employing a feature classification task using the pseudo-labels generated from the clustering step as the ground truth feature labels. The pseudo-classifier consists of a dense layer followed by a softmax on top of the encoder. In the DeepCluster approach, the pseudo-classifier is reinitialized after each iteration, since the indices of the pseudo-labels are permuted randomly after each clustering step. This makes training slow as the parameters cannot be reused. To alleviate this issue, we adopt the cluster centroid alignment technique proposed by Zhang et al. (2021), where we re-assign the pseudo-label indices from the clustering step, and thus, aligning them with the pseudo-classifier trained from the previous iteration. This allows the parameters to be reusable across iterations, hence speeding up training.

Compared to utterance similarity, pseudo classification clusters $Train_U$ utterances, either into $Train_L$ clusters or into totally new ones. Figure 4 compares the expected effect of the two steps.

3.1.3 Slot Classification

Information about slots in user utterances is a good source for feature awareness. For instance, while *"play despacito"* and *"play despacito on spotify"* are

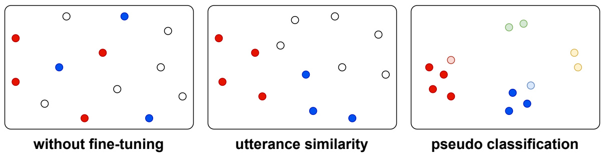

Figure 4: Utterance similarity vs. pseudo classification. Solid red and blue dots are $Train_L$ utterances, while green and yellow dots are $Train_U$ utterances grouped in new clusters.

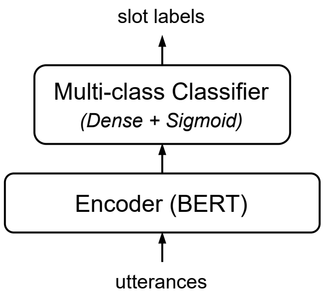

Figure 5: Fine-tuning with slot classification.

semantically similar utterances, the existence of `ServiceName:spotify` slot hints at the existence of a fine-grained user feature of playing a song on a service. In contrast to utterance similarity and pseudo classification, slot classification does not directly pull/push the utterances close to or far away from each other in the embedding space. Instead, we aim to make the representation model aware of the presence of slots in the input utterances, and we hypothesize that such information guides the model towards producing utterance representation with better feature separability.

This is modeled as a multi-class classification task (Figure 5), where the model is trained to detect which slots appear in the input utterances (e.g., `ServiceName`). Concretely, we add on top of the encoder a multi-class classifier, comprising a dense layer followed by sigmoid activation, and fine-tune the model using binary-cross-entropy loss. Note that we do not consider the exact position of the slots in the utterances, but rather their presence.

3.2 Feature Clustering

At inference time, we encode user unlabeled utterances into their representation vectors and use K-Means to cluster them into k feature clusters. To automatically choose the optimal value of k, we employ two techniques, namely the Elbow method (Thorndike, 1953) and the Silhouette score (Rousseeuw, 1987). Since utterances are a mix of novel and supported features, we classify whether each candidate feature cluster is novel using a feature novelty detection model, described next.

3.3 Feature Novelty Detection

While the trained representation model helps at projecting the utterances into a feature-separable space, it lacks information regarding feature's novelty w.r.t the NLU model. We detect novel feature clusters by exploiting signals from the NLU and utterance representation models. The NLU model is trained to jointly recognize intent and slot labels. The intent classifier (IC) and slot tagger (ST) heads are plugged on top of a BERT encoder. Slot labels follow the BIO schema (Ramshaw and Marcus, 1995). Note that our approach is agnostic to the choice of the NLU model. For novelty detection, we define the novelty confidence of each candidate feature cluster $C = \{u_1, u_2, ...\}$ as the average novelty of its utterances:

$$feat_{novel}(C) = \frac{1}{|C|} \sum_{u \in C} utt_{novel}(u)$$

where utt_{novel} is computed as follows:

$$utt_{novel}(u) = mean\big(st(u),\, ic(u),\, pc(u)\big)$$

where *slot tagging confidence (st)* is the confidence produced from the slot tagger. We tested with two variants: average over tokens *(st_{avg})*, and minimum over tokens *(st_{min})*. *Intent classification confidence (ic)* is the confidence of the most probable intent label produced by the intent classifier. *Pseudo classification confidence (pc)* is the confidence of the most probable pseudo cluster, produced by the pseudo-classifier from the utterance representation model. Feature clusters with novelty score greater than a pre-defined threshold are labeled as novel.

4 Experimental Setup

We evaluate the performance of each component of DNF independently as well as the overall discovery system using two datasets: SNIPS and a large French internal dataset.

4.1 Datasets

The SNIPS dataset (Coucke et al., 2018) consists of 14K English utterances spanning 7 intents with 72 unique slot labels. Since the dataset does not contain any feature labels, we augment it with feature labels using hand-crafted rules. We end up with a total of 43 features, each of which forms a cluster of utterances that semantically map to a user-perceived feature. Features cover three combinations:

- **Slot value features**: utterances in these features share the same intent, slots and at least one slot value. For example, *"play a song on spotify"* and *"play music on spotify"* belong to the same feature of *playing a song on Spotify service*.
- **Slot features**: utterances in these features share the same intent and same slots. For example, *"what is the weather tomorrow in new york"* and *"weather tonight in brooklyn"* belong to the same feature of *asking about weather at a specific time in a specific city*.
- **Intent features**: utterances in these features share the same intent, with potentially, several slots and/or slot values. For example, *"book me a restaurant tomorrow at 9pm"* belong to the feature of *booking a restaurant*.

Given the nature of the dataset, it was not possible to create cross-intent features that are semantically sensible. Out of the 43 features, 11 are used as a test set, while 32 as a training set, where 19 out of the 32 features are feature-labeled $Train_L$, and 13 features are feature-unlabeled training set $Train_U$. These 32 features are deemed supported by the NLU model, i.e., not novel. Out of the 43 features, 26 are slot value features, 15 are slot features, while 2 are intent features. We randomly sample utterances out of 32 features and add them to the test set so that our test set contains a mix of supported and novel features. The final dataset contains 32 features as training set and 43 features in the test set (11 of which are novel). On average, we have 210 utterances per feature cluster.

Our internal dataset has a total of 273K utterances comprising 41 features with different combinations. 15 out of the 41 features span single intent while the other features span two or more intents. 31 features are included in the feature-labeled training set $Train_L$. The number of utterances without feature labels in $Train_U$ set is in the order of millions. The remaining 10 features are part of the test set. We also sample utterances out of 31 features and add them to the test set. The final test set contains 41 features (10 of which are novel). The dataset covers 14 domains, 89 intents, 128 slot labels and has, on average, 6.6K utterances per feature cluster. All utterances were pre-processed such that users are not identified.

Table 1: Feature-aware utterance representation performance on SNIPS and internal datasets.

Training Strategy	SNIPS			Internal Dataset		
	NMI	*ARI*	*ACC*	*NMI*	*ARI*	*ACC*
No fine-tuning	0.626	0.309	0.396	==== *baseline* ====		
US only	0.737	0.451	0.512	+0.096	+0.147	+0.149
PC only	0.728	0.374	0.471	+0.116	+0.161	+0.130
US→PC	0.749	0.475	0.537	+0.116	+0.178	**+0.166**
SC→US→PC	0.766	0.474	0.557	+0.104	+0.166	+0.129
(SC+US)→PC	**0.782**	**0.531**	**0.605**	**+0.137**	**+0.216**	+0.140

To assess the ability of our model to accurately cluster cross-domain features, we split the internal dataset into:

- **Single-domain features**, where utterances belong to the same domain. 32 out of the 41 features are single domain features (25 train and 7 test), and
- **Cross-domain features**, where utterances belong to multiple domains (e.g., `Music` and `SmartHome`). 9 out of the 41 features are cross-domain features (6 train and 3 test). Features in these two splits cover different combinations.

4.2 Feature-Aware Utterance Representation

We use SBERT as a baseline utterance representation model, and compare different variants fine-tuned on different feature-related tasks. We consider the following fine-tuning strategies:

- *No fine-tuning*: We directly use the SBERT base model for utterance representation.
- *Utterance Similarity (US)*: The model is fine-tuned with only utterance similarity.
- *Pseudo Classification (PC)*: The model is fine-tuned with only pseudo classification.
- *US→PC*: The model is fine-tuned with both utterance similarity and pseudo classification sequentially.
- *Slot Classification (SC)→US→PC*: The model is fine-tuned with slot classification, utterance similarity and pseudo classification sequentially.
- *(SC+US)→PC*: Slot classification and utterance similarity are jointly trained to prevent overfitting, and then pseudo classification.

We use `paraphrase-mpnet-base-v2`, a pre-trained SBERT model, where we unfreeze all the layers during fine-tuning. We use AdamW as our optimizer (Loshchilov and Hutter, 2017), with an initial learning rate of $5e^{-5}$ and a weight decay

Table 2: Performance across different feature types.

Feature Type		SNIPS			Internal Dataset		
		NMI	*ARI*	*ACC*	*NMI*	*ARI*	*ACC*
By Novelty	*supported*	0.836	0.698	0.746	0.949	0.870	0.849
	novel	0.741	0.566	0.637	0.772	0.640	0.662
By Combination	*slot*	0.753	0.610	0.626	–	–	–
	slot value	0.811	0.590	0.646	–	–	–
	intent	0.547	0.512	0.804	–	–	–

Table 3: Intent-agnostic vs. intent-targeted discovery.

Intent	Intent-agnostic			Intent-targeted		
	NMI	*ARI*	*ACC*	*NMI*	*ARI*	*ACC*
AddToPlaylist	0.544	0.402	0.595	**0.671**	**0.573**	**0.789**
PlayMusic	0.700	0.429	0.527	**0.747**	**0.625**	**0.672**
SearchCreat.Work	0.631	0.431	0.598	**0.766**	**0.602**	**0.707**
SearchScrn.Event	0.639	0.474	0.634	**0.740**	**0.541**	**0.678**

of 0.01. We set the batch size to 16 and apply early stopping whenever we observe no improvement on a development set. On average, all fine-tuning strategies converge after 4 epochs.

Evaluation Metrics. To evaluate clustering quality against ground-truth, we follow previous work and report on the following metrics: Normalized Mutual Information (NMI), Adjusted Rand Index (ARI), and clustering Accuracy (ACC). All metrics range from 0 to 1. The higher the score, the better the clustering quality. We report the relative gains/losses over the baseline for the internal dataset. Since we are only interested in evaluating the quality of the fine-tuned representations, we use the reference number of clusters k in subsequent experiments and run a separate experiment to find the optimal k when evaluating the end-to-end system.

5 Experimental Results

We evaluate (1) the effect of our fine-tuning strategies to produce feature-aware representation on feature clustering, and (2) our feature novelty detection model.

5.1 Fine-tuning Results

Table 1 shows clustering performance using different fine-tuning strategies. Across datasets, our fine-tuning strategies outperform the vanilla SBERT baseline across all metrics. Stacking different fine-tuning tasks consistently results in better models. Fine-tuning with slot classification individually either before or after the other tasks (e.g., SC$\rightarrow$US$\rightarrow$PC) yields inferior performance compared to jointly running slot classification with other tasks. The best performing strategy is fine-tuning with slot classification and utterance similarity jointly and then running pseudo classification. This shows that slot classification is able to improve utterance representations in the feature space. In all subsequent experiments, we use the best observed model *(SC+US)$\rightarrow$PC*.

Table 4: Ablation study results.

Model	SNIPS			Internal Dataset		
	NMI	*ARI*	*ACC*	*NMI*	*ARI*	*ACC*
Standard	0.782	0.531	0.605	==== *baseline* ====		
Ablation	0.769	0.444	0.523	-0.081	-0.132	-0.087
Upperbound	0.812	0.530	0.672	–	–	–

Table 5: Choosing the number of clusters k.

Method	SNIPS				Internal Dataset			
	k	*NMI*	*ARI*	*ACC*	k	*NMI*	*ARI*	*ACC*
Gold k	43	0.782	0.531	0.605	41	==== *baseline* ====		
Silhouet.	24	0.761	0.511	0.593	35	-0.030	-0.063	-0.013
Elbow	30	0.790	0.573	0.632	39	+0.002	+0.003	+0.020

Performance breakdown across feature types. In Table 2, we report the performance of the best representation model on different feature types. First, we split the features in the test set by their novelty (whether the feature has been seen during training). Across the two datasets, our model clusters utterances from supported features better than from novel ones, which is expected.

In the second half of the table, we show a breakdown per combination. On SNIPS dataset, we achieve better clustering results for slot and slot value features than for intent features, which is reasonable as our training data contains only 2 features at intent level. On the internal dataset, clustering single-domain features is slightly better than cross-domain ones. For example, the model achieves an NMI of 0.867 for single domain features, and 0.823 NMI for cross-domain features.

Intent-agnostic vs. Intent-targeted discovery. For deeper analysis, we consider another discovery setup in an intent-targeted way, in which we only train and test with features from the same intent. This setup is particularly useful in cases where we focus on fine-grained discovery where the intent is assumed to be known. In Table 3, we report the results for this study on four SNIPS intents separately. The models trained with features belonging to the same intent perform generally better than when being trained with cross-intent features.

Ablation study. During fine-tuning, we leverage both feature-labeled ($Train_L$) and feature-unlabeled ($Train_U$) data. To understand their im-

Table 6: Performance of the feature novelty detection.

Signal	SNIPS			Internal Dataset		
	Prec.	*Rec.*	*F1*	*Prec.*	*Rec.*	*F1*
st_{avg}	0.407	1.000	0.579	==== *baseline* ====		
st_{min}	**0.750**	0.545	0.632	+0.076	0.000	+0.043
ic	0.500	0.636	0.560	-0.083	-0.100	-0.091
pc	0.615	0.727	**0.667**	**+0.167**	-0.200	-0.020
$feat_{novel}$	**0.750**	0.545	0.632	**+0.167**	0.000	**+0.091**

pact on model performance, we compare our model, trained with both kinds of data (standard model), against a model trained with only $Train_L$ data (ablation model). Moreover, we compare both models to an *upperbound* model, in which we also include the true feature labels of $Train_U$. Hence, the upperbound model is also trained with feature-labeled data only, similar to the ablation model, but with more features. We did not build an upperbound model on the internal dataset since all utterances in $Train_U$ are not annotated with feature labels.

As shown in Table 4, the standard model outperforms the ablation model on all metrics across both datasets, with NMI and ACC gains reaching 8-9%, and ARI gains of 13% on the internal dataset. This shows that our model benefits from feature-unlabeled data during training. Naturally, an abundance of feature-labeled data, although impractical, provides better feature-aware representations.

Choosing the number of feature clusters. As we are interested in evaluating the quality of the fine-tuned representations, we use the number of ground-truth features k from test data. However, this number is unknown in practice. We experimented with two popular techniques for predicting k: the Elbow method (Thorndike, 1953) and the Silhouette score (Rousseeuw, 1987). As shown in Table 5, on both datasets, Elbow method works slightly better than Silhouette score, with the predicted k closer to the ground-truth value. In terms of other metrics, Elbow method even produces better feature clusters than the baseline with gold k.

5.2 Results of Feature Novelty Detection

The base NLU model is a pre-trained BERT with 12 hidden layers, each with a size of 768. On top of the *CLS* classification token, we plug a linear projection head followed by softmax to perform intent classification, and a similar head on top of each token to perform slot tagging. To train the NLU model, we unfreeze all 12 hidden layers and fine-tune the two heads jointly.

To evaluate novelty detection in isolation from clustering, we use the ground-truth feature clusters and run novelty detection on top, i.e., to decide whether a ground-truth feature cluster is *novel* or *supported*. We harness different signals from the NLU model: st_{avg}, st_{min}, *ic*, *pc* and the combined signal $feat_{novel}$. As shown in Table 6, the combined signal $feat_{novel}$ performs the best across all metrics on the internal dataset. On SNIPS, $feat_{novel}$ excels against other signals in terms of precision, and places the second best in F1.

5.3 End-to-end DNF Evaluation

In this experiment, we first ran the clustering step with Elbow method to generate feature clusters. This resulted in 30 predicted clusters on SNIPS, and 39 clusters on the internal dataset (see Table 5). Then, we perform novelty detection on the predicted clusters.

To generate ground-truth labels for the predicted clusters, we assign a feature label l_C for each predicted cluster C by taking the majority vote of the labels of the utterances within the cluster. Table 7 shows the number of novel clusters predicted using each signal and their quality metrics. *all* is the baseline, where we assume all predicted feature clusters as novel. $feat_{novel}$ signal performs the best in terms of precision on SNIPS, together with st_{min}, which shows that the NLU slot confidence is a strong indicator of the novelty of an utterance. On the internal dataset, $feat_{novel}$ also achieves highest precision and F1, while discovering almost all novel features in the data (9 out of 10).

6 Conclusion

We introduced *feature discovery*, where, given a set of user utterances and a trained NLU model, we extract clusters of novel features composed of a combination of domains, intents, slots and/or their values. To this end, we presented DNF, a semi-supervised approach for extracting novel user features from a set of raw utterances, utilizing minimal feature knowledge from labeled data combined with feature-unlabeled data. DNF supports several fine-tuning strategies to improve utterance representation and make them separable in the feature space. We evaluated DNF on two datasets and observed significant improvements over baselines, showing the effectiveness of our method. In the future, we plan to explore various fine-tuning strategies for better utterance representations, as well as extending DNF to support different languages.

Table 7: End-to-end DNF evaluation.

Method	SNIPS				Internal Dataset			
	#novel features	*Precision*	*Recall*	*F1*	*#novel features*	*Precision*	*Recall*	*F1*
all	30	0.233	0.636	0.341	39	==== *baseline* ====		
st_{avg}	23	0.238	0.455	0.312	16	+0.295	0.000	+0.288
st_{min}	5	**0.600**	0.273	0.375	12	+0.462	0.000	+0.400
ic	21	0.238	0.455	0.312	8	+0.545	-0.200	+0.340
pc	10	0.400	0.364	**0.381**	9	+0.462	-0.200	+0.305
$feat_{novel}$	5	**0.600**	0.273	0.375	9	**+0.573**	-0.100	**+0.410**

7 Limitations

While we empirically showed that our approach performs well for novel feature discovery and generalizes across different datasets, we can identify avenues for improvement in terms of efficiency and model training. With DNF relying on good feature-aware sentence representations, obtaining such representations requires expensive fine-tuning steps. For example, using a single GPU, our cascaded fine-tuning strategy takes on average 7 hours on our internal dataset to reach convergence. Moreover, in workflows where NLU model refresh is frequent, the model's intent classification and slot tagging confidence distribution can shift over time. With DNF relying on observing confidence signals from the NLU model to determine feature novelty, a retraining/tuning of the novelty detection component would have to be performed frequently. Furthermore, our approach requires a small manually annotated feature-labeled dataset (feature labels in addition to intent and slot labels). These additional annotations require expertise and time, which poses a challenge during the data collection phase.

References

Abdalghani Abujabal, Claudio Delli Bovi, Sungho Ryu, Turan Gojayev, Fabian Triefenbach, and Yannick Versley. 2021. Continuous model improvement for language understanding with machine translation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Industry Papers, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 56–62. Association for Computational Linguistics.

Abdalghani Abujabal and Judith Gaspers. 2019. Neural named entity recognition from subword units. In *Interspeech 2019, 20th Annual Conference of the International Speech Communication Association, Graz, Austria, 15-19 September 2019*.

Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. 2018. Deep clustering for unsupervised learning of visual features. In *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part XIV*.

Giuseppe Castellucci, Valentina Bellomaria, Andrea Favalli, and Raniero Romagnoli. 2019. Multi-lingual intent detection and slot filling in a joint bert-based model. *CoRR*.

Alice Coucke, Alaa Saade, Adrien Ball, Théodore Bluche, Alexandre Caulier, David Leroy, Clément Doumouro, Thibault Gisselbrecht, Francesco Caltagirone, Thibaut Lavril, Maël Primet, and Joseph Dureau. 2018. Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces. *CoRR*.

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 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*.

Rashmi Gangadharaiah and Balakrishnan Narayanaswamy. 2019. Joint multiple intent detection and slot labeling for goal-oriented dialog. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*.

Xibin Gao, Radhika Arava, Qian Hu, Thahir Mohamed, Wei Xiao, Zheng Gao, and Mohamed AbdelHady. 2021. Graphire: Novel intent discovery with pretraining on prior knowledge using contrastive learning. In *KDD 2021 Workshop on Pretraining: Algorithms, Architectures, and Applications*.

Chih-Wen Goo, Guang Gao, Yun-Kai Hsu, Chih-Li Huo, Tsung-Chieh Chen, Keng-Wei Hsu, and Yun-Nung Chen. 2018. Slot-gated modeling for joint slot filling and intent prediction. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human*

Language Technologies, NAACL-HLT, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 2 (Short Papers).

Shailza Jolly, Tobias Falke, Caglar Tirkaz, and Daniil Sorokin. 2020. Data-efficient paraphrase generation to bootstrap intent classification and slot labeling for new features in task-oriented dialog systems. In *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020 - Industry Track, Online, December 12, 2020.*

Joo-Kyung Kim and Young-Bum Kim. 2018. Joint learning of domain classification and out-of-domain detection with dynamic class weighting for satisficing false acceptance rates. In *Interspeech 2018, 19th Annual Conference of the International Speech Communication Association, Hyderabad, India, 2-6 September 2018.*

Young-Bum Kim, Sungjin Lee, and Karl Stratos. 2017. ONENET: joint domain, intent, slot prediction for spoken language understanding. In *2017 IEEE Automatic Speech Recognition and Understanding Workshop, ASRU 2017, Okinawa, Japan, December 16-20, 2017.*

Ting-En Lin and Hua Xu. 2019. Deep unknown intent detection with margin loss. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers.*

Ting-En Lin, Hua Xu, and Hanlei Zhang. 2020. Discovering new intents via constrained deep adaptive clustering with cluster refinement. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020.*

Bing Liu and Ian R. Lane. 2016. Attention-based recurrent neural network models for joint intent detection and slot filling. In *Interspeech 2016, 17th Annual Conference of the International Speech Communication Association, San Francisco, CA, USA, September 8-12, 2016.*

Han Liu, Xiaotong Zhang, Lu Fan, Xuandi Fu, Qimai Li, Xiao-Ming Wu, and Albert Y. S. Lam. 2019. Reconstructing capsule networks for zero-shot intent classification. 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 2019, Hong Kong, China, November 3-7, 2019.*

Pengfei Liu, Youzhang Ning, King Keung Wu, Kun Li, and Helen Meng. 2021. Open intent discovery through unsupervised semantic clustering and dependency parsing. *CoRR.*

Ilya Loshchilov and Frank Hutter. 2017. Fixing weight decay regularization in adam. *CoRR.*

Grégoire Mesnil, Xiaodong He, Li Deng, and Yoshua Bengio. 2013. Investigation of recurrent-neural-network architectures and learning methods for spoken language understanding. In *INTERSPEECH 2013, 14th Annual Conference of the International Speech Communication Association, Lyon, France, August 25-29, 2013.*

Patti J. Price. 1990. Evaluation of spoken language systems: the ATIS domain. In *Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania, USA, June 24-27, 1990.*

Lance A. Ramshaw and Mitch Marcus. 1995. Text chunking using transformation-based learning. In *Third Workshop on Very Large Corpora, VLC@ACL 1995, Cambridge, Massachusetts, USA, June 30, 1995.*

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019.*

Peter J Rousseeuw. 1987. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics.*

Prashanth Gurunath Shivakumar, Mu Yang, and Panayiotis G. Georgiou. 2019. Spoken language intent detection using confusion2vec. In *Interspeech 2019, 20th Annual Conference of the International Speech Communication Association, Graz, Austria, 15-19 September 2019.*

Robert L Thorndike. 1953. Who belongs in the family? *Psychometrika.*

Nikhita Vedula, Rahul Gupta, Aman Alok, and Mukund Sridhar. 2020a. Automatic discovery of novel intents & domains from text utterances. *CoRR.*

Nikhita Vedula, Nedim Lipka, Pranav Maneriker, and Srinivasan Parthasarathy. 2020b. Open intent extraction from natural language interactions. In *WWW '20: The Web Conference 2020, Taipei, Taiwan, April 20-24, 2020.*

Kiri Wagstaff, Claire Cardie, Seth Rogers, and Stefan Schrödl. 2001. Constrained k-means clustering with background knowledge. In *Proceedings of the Eighteenth International Conference on Machine Learning (ICML 2001), Williams College, Williamstown, MA, USA, June 28 - July 1, 2001.*

Yu Wang, Yilin Shen, and Hongxia Jin. 2018. A bi-model based RNN semantic frame parsing model for intent detection and slot filling. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 2 (Short Papers).*

Henry Weld, Xiaoqi Huang, Siqu Long, Josiah Poon, and Soyeon Caren Han. 2023. A survey of joint intent detection and slot filling models in natural language understanding. *ACM Comput. Surv.*, 55(8):156:1–156:38.

Congying Xia, Chenwei Zhang, Xiaohui Yan, Yi Chang, and Philip S. Yu. 2018. Zero-shot user intent detection via capsule neural networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*.

Guangfeng Yan, Lu Fan, Qimai Li, Han Liu, Xiaotong Zhang, Xiao-Ming Wu, and Albert Y. S. Lam. 2020. Unknown intent detection using gaussian mixture model with an application to zero-shot intent classification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*.

Hanlei Zhang, Hua Xu, Ting-En Lin, and Rui Lyu. 2021. Discovering new intents with deep aligned clustering. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*.