An Open Intent Discovery Evaluation Framework

Grant Anderson[1,2], Emma Hart[1], Dimitra Gkatzia[1], Ian Beaver[2]
[1]Edinburgh Napier University, UK, [2]Verint Systems Ltd., USA
Correspondence: grant.anderson@verint.com

Abstract

In the development of dialog systems the discovery of the set of target intents to identify is a crucial first step that is often overlooked. Most intent detection works assume that a labelled dataset already exists, however creating these datasets is no trivial task and usually requires humans to manually analyse, decide on intent labels and tag accordingly. The field of Open Intent Discovery (OID) addresses this problem by automating the process of grouping utterances and providing the user with the discovered intents. Our OID framework allows for the user to choose from a range of different techniques for each step in the discovery process, including the ability to extend previous works with a human-readable label generation stage. We also provide an analysis of the relationship between dataset features and optimal combination of techniques for each step to help others choose without having to explore every possible combination for their unlabelled data.

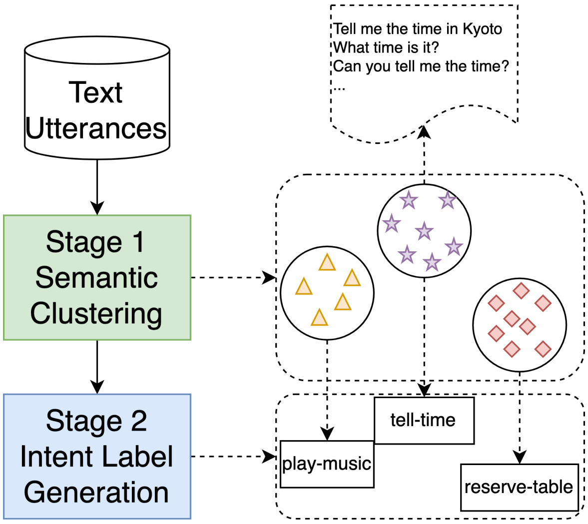

Figure 1: An example of the automated discovery and labelling of intents in an given dataset of unlabelled/partially labelled text utterances. First, the utterances are clustered for similar semantic intent, then human-readable labels are generated for each identified cluster.

1 Introduction

A major first task for a goal-oriented dialogue system is to identify the intent behind the user's utterance using a Natural Language Understanding module. This module is often implemented as a classifier, trained on a set of pre-defined intent labels (Chen et al., 2013; Coucke et al., 2018; Goo et al., 2018; Kim et al., 2016; Liu and Lane, 2016; Zhong and Li, 2019). Discovering these intents in real-world systems can be a laborious and time-consuming task involving a domain expert exploring the dataset and curating a representative set of labels. This task will also need to be repeated regularly as new intents emerge through time. The field of OID seeks to automatically discover unknown intents in a set of unlabelled/partially labelled utterances without requiring such manual effort.

There exists an issue in the current literature in that many works focus only on the development of clustering algorithms to identify utterances of similar intent, without progressing to label the cluster with a human-readable intent label (Perkins and Yang, 2019; Lin et al., 2019; Zhang et al., 2021b; Shen et al., 2021; Kumar et al., 2022). In order for downstream systems to make full use of the new intents, a human would be required to analyse the cluster manually, decide on its meaning and label it accordingly.

Evaluation methods are also inconsistent across the field. Some works report on classification or clustering metrics while others evaluate quality of generated labels, but rarely are these reported for the same datasets. There are also differences in the definition of 'intent' and the features of the datasets used for evaluation. Some works consider intents in the abstract such as 'CustomerService' or 'Baggage' in the air travel domain. Other works take a much stricter definition e.g. only an Action(verb)-Object(noun) pair. Some datasets contain a mixture

Proceedings of the 25th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pages 760–769
September 18–20, 2024.

of these intent types. These issues make it difficult to identify a truly state-of-the-art (SOTA) technique for different domains and features of dataset.

We present an OID framework which views OID as a two stage process: Semantic Clustering, and Intent Label Generation (see Figure 1). We focus on the generation of high quality labels for an unlabelled/partially labelled dataset, produced by combining a semantic representation method, clustering algorithm, candidate extraction method and a label selection method. We evaluate 20 combinations of representation/clustering/extraction/selection methods on 9 datasets. Our key contributions include: (1) We introduce our novel OID framework providing a choice of a number of different techniques at every step in the process.[1] (2) We extend previous OID work to include a human-readable intent stage. (3) A rigorous investigation into instantiating choice of representation/clustering model/extraction/selection which reveals the optimal settings for datasets and target intents.

2 Related Work

State-of-the-art OID techniques utilise semi-supervised learning such as in DSSCC (Deep Semi-Supervised Contrastive Clustering) (Kumar et al., 2022) and DeepAligned (Zhang et al., 2021b). A portion of intents are known in advance and these are used to aid the clustering stage in discovering both the known intent clusters and estimate a number of new, unknown intents. Shen et al. (2021) take a different approach, by pre-training a representation model with a labelled dataset from the same domain as the target unlabelled dataset and then using unsupervised KMeans clustering on the target dataset to discover intents.

There are several works which attempt to solve the problem in an unsupervised fashion. Chatterjee and Sengupta (2020) adapted the DBSCAN clustering algorithm (Ester et al., 1996) in an attempt to handle discovering new intents in datasets with unbalanced distributions, while others such as Liu et al. (2021) use simple KMeans clustering. Liu et al. (2021) are one of the few OID works which include a label generation stage. Each cluster has candidate intent labels extracted using a dependency parser to find Action(verb)-Object(noun) pairs within the utterances and the most common pair is assigned as an auto-generated, human-readable label for the cluster. Their technique discovered the correct number of clusters for the SNIPS dataset and produced labels which were clearly semantically similar to the ground-truth intents, however no quantitative evaluation was conducted. A more challenging dataset would prove more difficult both to cluster and to evaluate by manual inspection. Vedula et al. (2020) looked at intent discovery as a sequence tagging task. A neural model sequence tagger is trained to tag action and object words in text utterances. This technique differs in that it will produce an intent for every text utterance and may produce many distinct pairs that express the same intent.

In our concurrent work, we presented experimental results for different combinations of candidate extraction and intent label selection techniques against a large generative PLM (Anderson et al., 2024). In order to produce fine-grained intents, we also proposed an extension to the Action-Object extraction method used in Liu et al. (2021) which captures more detail from the utterances by including compound nouns or adjectives that are related to the Object, and negations related to the Action.

Zhang et al. (2021a) introduced a platform for open intent recognition. They combine the related tasks of open intent detection and discovery to both identify the known intents and discover new ones. The detection module identifies known intent samples and groups unknown samples into a single class of open intent. The discovery module then performs clustering to group the unknown samples and present them as new intents. Our framework differs in that we focus only on discovery and not detection. We also include a human-readable label generation stage while TEXTOIR provides keywords to represent their discovered intents. These keywords are helpful, however, out of context they would be difficult to fully understand without further analysing the utterances themselves.

3 Methods

Many current OID techniques can fit into the same two stage pattern (see Figure 2). Stage 1 consists of semantic clustering and is split into two steps. First, semantic representations are obtained for each utterance, then these are grouped with a clustering algorithm to identify semantically similar intents. Stage 2 involves the generation of a natural language label for each cluster. First, candidate labels are extracted or generated for each cluster, finally, a label is chosen from these candidates.

[1] https://github.com/GAnderson01/open-intent-discovery

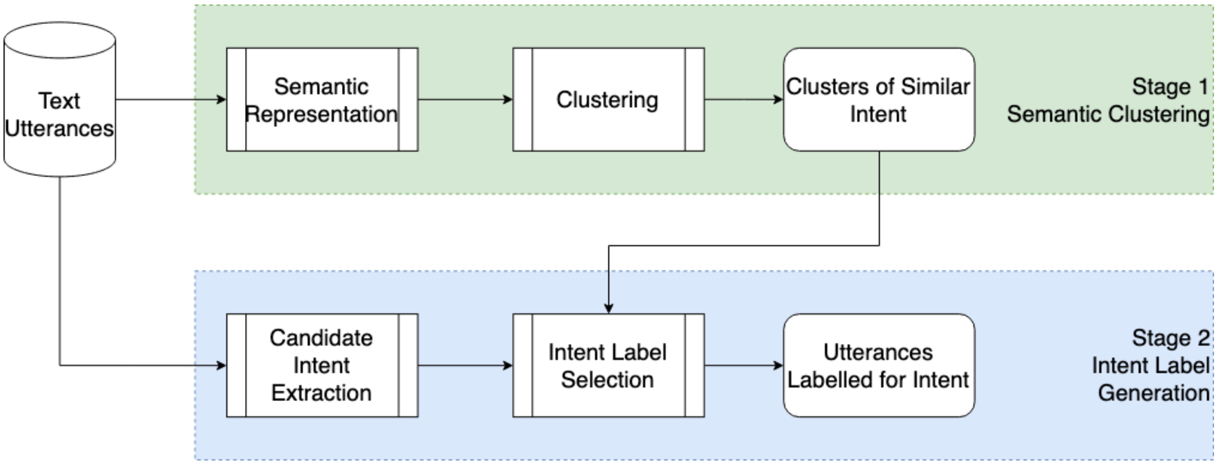

Figure 2: The Open Intent Discovery Framework is split into two main stages. In Stage 1, utterances are clustered for semantic similarity and in Stage 2, a human-readable label is produced for each cluster.

At each step in the process there are many different options for a researcher to choose from. At the Semantic Representation step, choices include using BERT, Universal Sentence Encoder or one of many other embedding options. For clustering, one could choose KMeans, DBSCAN etc. When looking for candidate labels, possibilities include an extraction method, such as the Action-Object extraction used by Liu et al. (2021), or a label could be generated by prompting a Pre-trained Language Model (PLM) such as ChatGPT or T0pp. Finally, a label must be chosen from the candidates e.g. by choosing the most frequent candidate or even by prompting a PLM, specifying the candidates to choose from. Our framework allows for any combination of options to be evaluated. Table 1 displays the different options we explored for each step in the framework. We refer to a combination of semantic representation, clustering, candidate extraction and intent label selection techniques as a configuration.

Most related works do not progress to Stage 2, and simply present the clusters of semantically similar texts as the found intents. Using the framework, we are able to extend these with Stage 2 techniques allowing us to evaluate the quality of the final natural language labels for clusters found by all OID techniques. For each cluster, we measure both the cosine similarity and the BARTScore between the most common ground truth label in the cluster and the generated label.

One of the goals of this work is to find common patterns in the configurations for datasets with similar features. It is hoped that this will help others to choose the best configuration for their own datasets rather than having to perform a brute force search, or best guess.

The framework implements each step as a python module. Each can be run individually provided they are given any input required. When chained together, they execute the entire OID process end-to-end.

3.1 Stage 1: Semantic Clustering

The first stage is to collect the text utterances into groups of semantically similar intent. To achieve this, we first need to obtain good semantic representations of the utterances via some embedding model, then provide these to a clustering algorithm.

Semantic Representation Using PLMs to obtain embeddings for text utterances before applying these in a downstream NLP task has been repeatedly shown to perform well. However, the question of which PLM to use for a particular problem and dataset can be unclear. The semantic representation module supports any huggingface, sentence-transformers or tensorflow-hub based PLM embedding models. We use three PLMs to obtain semantic representations for the utterances in order to cluster for intent. These are as follows: bert-base-uncased (Devlin et al., 2018), all-mpnet-base-v2 (Reimers and Gurevych, 2019) and Universal Sentence Encoder (Cer et al., 2018). These PLMs have been shown to perform well in previous OID works (Zhang et al., 2021b; Kumar et al., 2022; Liu et al., 2021; Chatterjee and Sengupta, 2020).

Clustering The optimal clustering algorithm to use for a given dataset depends on the features of the dataset. For example, KMeans is more suited to

Stage 1: Semantic Clustering		Stage 2: Intent Label Generation	
Semantic Representation	**Clustering Algorithm**	**Candidate Extraction**	**Intent Label Selection**
all-mpnet	KMeans	Action-Object Pairs	Most Frequent
BERT	DBSCAN	T0pp Prompting	T0pp Prompting
Universal Sentence Encoder	ITER_DBSCAN		
	DeepAligned		

Table 1: Evaluated choices at each step of the framework

finding clusters of similar sizes (a balanced dataset), and a flat geometry, while density based methods such as DBSCAN can handle uneven cluster sizes (an imbalanced dataset) and non-flat geometry. We explore both unsupervised (KMeans, DBSCAN and ITER_DBSCAN) and semi-supervised (DeepAligned) intent clustering algorithms. Both ITER_DBSCAN and DeepAligned are intent discovery techniques which do not involve creating human readable labels, and so our framework extends them with the Stage 2 label generation techniques.

Most clustering algorithms require some hyperparameters to be set e.g. KMeans requires the target number of clusters (k). However in many cases these hyperparameters are unknown and so a tuning exercise is required. In order to find optimal hyperparameters, a search across the hyperparameter space must be conducted and each clustering result evaluated against some metric. This metric, is one of the choices that can be set in the framework.

3.2 Stage 2: Intent Label Generation

The second stage is to choose or generate a natural language label to represent the cluster as an intent. First, candidates are found from the cluster either using a dependency parser or prompting a PLM, then one of the candidates is selected by some method such as most frequent, or, again, prompting a PLM.

Candidate Label Extraction We implement two techniques to extract candidates intents for the identified clusters. The first finds Action-Object pairs in utterances as in (Liu et al., 2021). An Action-Object pair consists of a verb/infinitive (the Action) and it's target, a noun or subject (the Object). e.g. "schedule a meeting for tomorrow" contains the Action-Object pair *schedule-meeting*. If either an action or object is not present in an utterance, then the candidate contains 'NONE' in it's place. This technique assumes a very strict definition of intent and as such, could never produce a more abstract intent such as 'query' or 'confirmation'. Therefore, we also experiment with PLM Prompting, to allow for more freedom in the candidate intents.

To produce a candidate with a PLM, we obtain the response when it is given the below prompt:

> *"Given the following utterance:* [utterance]. *The intent was to"*

Intent Label Selection The final step in the framework is to choose an intent label for every cluster from one of the candidates identified. We experiment with two techniques. As in Liu et al. (2021), we choose the most frequent candidate. Where Action-Object extraction was used we ignore incomplete pairs by not considering any with the word 'NONE'. If a cluster produced no candidates, then no label will be generated. The second selection technique also prompts a PLM using the following:

> *"Given these utterances:* [cluster_utterances]. *What is the best fitting intent, if any, among the following:* [top_3_candidates]*"*

where [cluster_utterances] is all of the utterances present in the cluster and [top_3_candidates] are the three most common candidates in the cluster. This prompt was crafted to provide the PLM with some options for a suitable label while still leaving it with the possibility of generating something new.

4 Datasets

We intentionally select a group of datasets with different features to analyse the correlation between features and optimal configurations. SNIPS (Coucke et al., 2018), AskUbuntuCorpus and WebApplications Corpus (Braun et al., 2017) all contain the Action-Object format of intents and are queries/commands in conversational style. DBPedia14 Sampled and StackOverflow (Xu et al., 2015) are labelled for Topic. DBPedia14_Sampled

Dataset	Intent Type	Number of Samples	Number of Intents	Intent Balance	Average Number of Words	Vocabulary Size
AskUbuntu	Action-Object	Small (162)	Small (5)	Imbalanced (7.13)	Short (7.94)	Small (474)
SNIPS	Action-Object	Large (13784)	Small (7)	Slightly Imbalanced (1.03)	Short (9.15)	Large (13418)
WebApplications	Action-Object	Small (89)	Small (8)	Imbalanced (23.00)	Short (8.01)	Small (300)
Banking77	Mixed	Large (13083)	Large (77)	Imbalanced (3.03)	Short (11.71)	Medium (3027)
ChatbotCorpus	Mixed	Small (206)	Small (2)	Slightly Imbalanced (1.64)	Short (7.70)	Small (173)
CLINC	Mixed	Large (22500)	Large (150)	Balanced (1.00)	Short (8.31)	Medium (6420)
PersonalAssistant	Mixed	Large (20735)	Medium (46)	Imbalanced (247.96)	Short (6.84)	Medium (7896)
DBPedia14 Sampled	Topic	Large (14000)	Medium (14)	Balanced (1.00)	Long (46.29)	XLarge (75214)
StackOverflow	Topic	Large (20000)	Medium (20)	Balanced (1.00)	Short (8.32)	Large (16773)

Table 2: Features of Each Dataset

Feature	Categories
Intent Type	Action-Object, Topic, Mixed
Size	Small (<250), Large (>= 250)
Number of Intents	Small (<10), Medium (>=10, <50) Large (>=50)
Intent Balance	Balanced (IR = 1.00), Slightly Imbalanced (IR >1, <2), Imbalanced (IR >= 2)
Average Number of Words	Short (<20), Long (>=20)
Vocabulary Size	Small (<500), Medium (>=500, <10,000) Large (>=10,000,<50,000), XLarge (>=50,000)

Table 3: Categorisations of Dataset Features

contains a sample of 14,000 entries from the DBPedia14 dataset (Lehmann et al., 2014). Banking77 (Casanueva et al., 2020), ChatbotCorpus (Braun et al., 2017), CLINC (Larson et al., 2019) and PersonalAssistant (Liu et al., 2019) contain a mix of both Action-Object and Topic form of intents. See Table 2 for full details of the features of each dataset.

4.1 Dataset Feature Definitions

We categorise the datasets by intent type, size, number of intents, whether the intents are balanced, average number of words and vocabulary size (see Table 3).

Intent Type Many works differ in their definition of intent, whether explicitly in their method or implicitly in their choice of dataset. Liu et al. (2021) define an intent as an Action(verb)-Object(noun) pair in an utterance e.g. "can you reschedule my delivery" has the pair 'reschedule-delivery'. Vedula et al. (2020) also use this definition, naming these 'actionable intents'. Other datasets have more abstract labels that are closer to topics. In these cases, methods like Action-Object extraction are unlikely to produce intents which reflect the ground-truths and so another extraction method would likely produce better results. Finally, a dataset can be mixed such that it contains both Action-Object pairs and abstract labels like topics. Therefore, we categorise all datasets used in our experiments as one of Action-Object, Topic or Mixed.

Number of Samples We use a selection of datasets of varying sizes. The smallest dataset having less than 100 samples, while the largest has almost 22.5k. We categorise the datasets as either small or large where small is defined as having less than 250 samples and large has anything over 250.

Number of Intents The number of ground-truth intent labels in a dataset can be considered the 'ideal' number of clusters that should be found by the clustering algorithm. The datasets we use range from 2 to 150 intents and we categorise this feature as small, medium and large where small is defined as having less than 10 intents, medium has between 10 and 50 and anything over 50 is large.

Intent Balance The ground-truth label distribution is also a defining feature of datasets. We use the Imbalance Ratio (IR) as a measure of imbalance. This is simply the number of majority label samples over the number of minority label samples. An IR of 1.00 represents a completely balanced dataset with equal samples for every ground-truth label. Anything above this represents an increasing magnitude of imbalance. The datasets used range from balanced to an IR of 247.96 (the majority label has almost 250 times the samples of the minority label). We categorise this feature as balanced, slightly imbalanced and imbalanced where balanced has an IR of 1.00, slightly imbalanced has IR greater than 1 but less than 2 and imbalanced has an IR of 2 and above.

Average Number of Words The majority of the datasets used are dialogue utterances and have relatively low average number of words of less than 12 while only one exceeds this at 46.29. We therefore categorise this feature as short and long where short is less than 20 and long is 20 and over.

Vocabulary Size The final dataset feature we explore is the number of unique words across all utterances in the dataset i.e. the vocabulary size. There is quite a spread across the datasets we use in our experiments and so we categorise this as small (with less than 500), medium (from 500 to 10,000), large (10,000 to 50,000) and xlarge (over 50,000).

5 Experiments

5.1 Experimental Setup

We evaluate all possible combinations of the choices in Table 1, with the only exceptions being for the previous OID works ITER_DBSCAN and DeepAligned where we use the Semantic Representation model from the original works (Universal Sentence Encoder and BERT respectively). This results in 20 configurations for each dataset for the framework to execute.

Each configuration involves a clustering algorithm and clustering measure for conducting hyperparameter tuning. Clustering is attempted for a range of hyperparameter values and evaluated using the specified measure (we use silhouette score for our experiments). The hyperparameters with the best score according to the chosen clustering measure are used for the configuration. For kmeans, we must estimate the optimal number of clusters *k*. We therefore conduct clustering for *k* between 2 and 200 or the number of utterances in the dataset, whichever is lower. We use the scikit-learn implementation of kmeans. For DBSCAN, there are at least two parameters to be set. *eps* is the maximum distance that can be between two samples to consider them as being in the same neighbourhood and *min_samples* is the minimum number of samples in a neighbourhood for a sample to be considered a 'core' sample. To keep hyperparameter tuning compute time down, we focus on tuning *eps* only, while *min_samples* is set to 5. We cluster for *eps* between 0.1 and 1.0 with increments of 0.01. Again, we use the scikit-learn implementation for DBSCAN. For ITER-DBSCAN, there are five hyperparameters to be tuned. In addition to *eps* and *min_samples*, there is also the change in these value for each iteration, *delta_eps* and *delta_min_samples* and finally, the maximum number of iterations to run *max_iteration*. An exhaustive search across these hyperparameters for every ITER-DBSCAN configuration and every dataset would be unfeasible. We therefore generate 20 random sets of hyperparameters and cluster with these for every relevant configuration. We use the implementation of ITER-DBSCAN from the original work (Chatterjee and Sengupta, 2020). For the semi-supervised technique, DeepAligned, we use the implemnetation provided by the authors with their default values (Zhang et al., 2021b).

For configurations involving PLM prompting, we chose T0pp as it is open-source, small enough to deploy on accessible hardware and has produced impressive results (Sanh et al., 2021). We utilised AWS Sagemaker Notebook to run our experiments. A g4dn.12xlarge instance was used with any configuration with T0pp prompting and a g4dn.xlarge for the others.

5.2 Evaluation

We use two automated metrics (average cosine similarity and average BARTScore (Yuan et al., 2021)) to evaluate the quality of the final generated labels compared to the ground truth intents. Both the generated and ground truth label sets are normalised by converting to lower case, splitting on Pascal/snake case and removing hyphens and embeddings obtained using Universal Sentence Encoder.

For each unique ground-truth (gt) label, we define C^* as the subset of clusters where the most common ground-truth ($mcgt$) equals gt. The similarity score for each gt is then the average of the similarity between the generated label and the $mcgt$ for each cluster in C^* ($sim(c)$). If none of the identified clusters is assigned gt then the score is 0 (see Equation 1).

$$avg\_label\_sim(gt) = \begin{cases} \frac{\sum_{c \in C*} sim(c)}{N_{C*}} & \text{, if } N_{C^*} > 0 \\ 0 & \text{, if } N_{C^*} = 0 \end{cases} \tag{1}$$

where N_{C^*} is the number of clusters in C^*.

The final average similarity score for the configuration is calculated as in Equation 2.

$$config\_score = \frac{\sum_{gt \in GT} avg\_label\_sim(gt)}{N_{GT}} \tag{2}$$

where GT is the set of all ground-truth intents and N_{GT} is the number of ground-truth intents.

The optimal configuration for each dataset is the configuration which produces the highest $config\_score$. Collecting these results from

Dataset	Semantic Representation	Clustering Algorithm	Candidate Extraction	Label Selection	No. Clusters	Avg. Cosine Similarity	Avg. BART Score
AskUbuntu	use	KMeans	Action-Object	T0pp Prompting	6(+1)	0.4661	-5.7580
SNIPS	use	KMeans	Action-Object	T0pp Prompting	8(+1)	0.6163	-3.9832
WebApplications	all-mpnet	KMeans	Action-Object	T0pp Prompting	6(-2)	0.4993	-5.4204
Banking77	all-mpnet	KMeans	Action-Object	Most Frequent	196(+119)	0.4678	-5.4880
ChatbotCorpus	use	KMeans	T0pp Prompting	Most Frequent	4(+2)	0.4384	-4.9715
CLINC	use/ all-mpnet	KMeans	Action-Object	T0pp Prompting/ Most Frequent	163(+13)/ 155(+5)	0.5050/ 0.5044	-4.7101/ -4.5701
PersonalAssistant	all-mpnet	KMeans	Action-Object	T0pp Prompting	60(+14)	0.3843	-5.2462
DBPedia14 Sampled	use/ all-mpnet	KMeans	T0pp Prompting	Most Frequent	11(-3)/ 10(-4)	0.3378/ 0.3091	-5.3313/ -5.3169
StackOverflow	use/ all-mpnet	KMeans	Action-Object	T0pp Prompting	23(+3) / 21(+1)	0.4861/ 0.3922	-5.2722/ -5.2692

Table 4: Unsupervised configurations producing the optimal labels for each dataset. The difference in number of clusters and ground-truth intents in shown in brackets. Where the evaluation metrics disagree on a configuration choice, both are reported as (*cosine similarity score*/*BART score*)

datasets of different features allows us to analyse the optimal configurations alongside the features in order to infer any dependencies between them.

6 Results and Analysis

6.1 Unsupervised Clustering

Table 4 shows the optimal configurations together with the average scores that they achieved in the unsupervised clustering setting. Table 5 shows a sample of the final labels generated with unsupervised clustering for each dataset. Many of these labels are of high quality and would be useful in downstream systems. In all unsupervised settings, KMeans produced the clusters for the optimal configuration. In most cases, the number of clusters exceeded the number of ground-truth intents. This results in some clusters being assigned the same $mcgt$. The labels are however, highly semantically similar with their ground-truth counterparts. It appears that the configurations using ITER_DBSCAN have produced a great overestimation of the number of clusters e.g. for SNIPS, the best performing configuration using ITER_DBSCAN produced 39 clusters. The generated labels are still semantically similar to their ground-truths, however there is more variety per ground-truth label due to the finer-grained clusters generating different final labels, resulting in lower performance according to the evaluation metrics.

Where Action-Object candidate extraction was used it has resulted in some generated labels being less descriptive than would perhaps be desired, e.g. in SNIPS *find-schedule* for **SearchScreeningEvent** is too generic. The samples for this intent are look-

Ground Truth	Generated Label
	SNIPS
AddToPlaylist	add-song
BookRestaurant	book-restaurant
GetWeather	give-forecast
PlayMusic	play-music/find-soundtrack
RateBook	rate-novel
SearchCreativeWork	find-show
SearchScreeningEvent	find-schedule
	Banking77
card_arrival	received-card/track-card
edit_personal_details	edit-details?/change-address.
exchange_charge	exchange-currencies/exchanging-currencies?
getting_virtual_card	get-card?
passcode_forgotten	reset-password?/reset-passcode?
request_refund	get-refund/give-refund
verify_my_identity	verify-identify?
verify_source_of_funds	get-funds/verify-source
	CLINC
how_old_are_you	ask-age/tell-birthday
improve_credit_score	improve-score
oil_change_when	change-oil
plug_type	need-converter
schedule_meeting	reserve-room/set-meeting
text	tell-text
transactions	show-transactions
who_do_you_work_for	tell-brand
	StackOverflow
apache	redirect-requests/using-proxy
cocoa	Cocoa/converting-string
hibernate	Hibernate
linq	using-linq
qt	Qt: How to end line with QTextEdit [Qt] [C++],
spring	Spring
wordpress	get-posts
visual-studio	Visual Studio 2008

Table 5: Sample labels produced by the optimal configurations. Where multiple clusters are assigned the same $mcgt$, we report two sample generated labels.

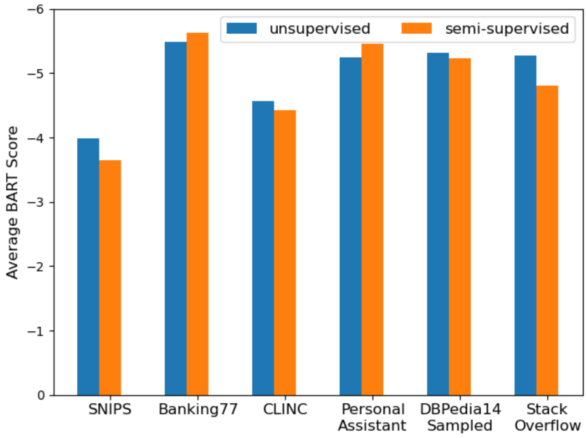

Figure 3: Average BART Scores for the optimal unsupervised configs vs optimal semi-supervised configs for each dataset. Closer to zero is better.

Feature	Semantic Representation	Clustering Algorithm	Extraction Method	Selection Method
Intent Type				
Action-Object	use	KMeans	Action-Object	T0pp Prompting
Topic	all-mpnet	KMeans	No Majority	No Majority
Mixed	all-mpnet	KMeans	Action-Object	Most Frequent
Size				
Small	use	KMeans	Action-Object	T0pp Prompting
Large	all-mpnet	KMeans	Action-Object	No Majority
Num. Intents				
Small	use	KMeans	Action-Object	T0pp Prompting
Medium	all-mpnet	KMeans	Action-Object	T0pp Prompting
Large	all-mpnet	KMeans	Action-Object	Most Frequent
Imbalance				
Balanced	all-mpnet	KMeans	Action-Object	Most Frequent
Slightly Imbalanced	use	KMeans	No Majority	No Majority
Imbalanced	all-mpnet	KMeans	Action-Object	T0pp Prompting
Avg. Num. Words				
Short	all-mpnet	KMeans	Action-Object	T0pp Prompting
Long	all-mpnet	KMeans	T0pp Prompting	Most Frequent
Vocab. Size				
Small	use	KMeans	Action-Object	T0pp Prompting
Medium	all-mpnet	KMeans	Action-Object	Most Frequent
Large	No Majority	KMeans	Action-Object	T0pp Prompting
XLarge	all-mpnet	KMeans	T0pp Prompting	Most Frequent

Table 6: Most common options by dataset features when evaluating using BART Score

ing for the movie schedules at cinemas and often ask for "the movie schedule". Also, there are many fine-grained intents in Banking77 which require more detail to be immediately useful e.g. a ground-truth intent such as **get_disposable_virtual_card** could not be produced using Action-Object extraction as in (Liu et al., 2021). It would therefore, be useful to extend the Action-Object candidate extraction to include compound nouns and adjectives to capture further details in the candidates.

6.2 Semi-supervised Clustering

Figure 3 shows the difference in BART Score for the optimal configurations using unsupervised clustering vs the optimal config that used the semi-supervised clustering method DeepAligned. The quality of the generated labels mostly outperform their unsupervised counterparts. However, DeepAligned produces poorer results for both Banking77 and Personal Assistant. These datasets are both large in size and imbalanced which may cause the DeepAligned model to overfit to the majority samples. DeepAligned also failed to complete for the small datasets, possibly due to a lack of training samples to complete an optimizer step.

6.3 Mapping Features to Configuration

Table 6 shows how the various dataset features affect the optimal unsupervised configuration when evaluating using the BART Score. Each value represents the most commonly used option for a given dataset feature and step in the framework, e.g. for datasets with Action-Object as the target intent type, Universal Sentence Encoder was the majority optimal choice for Semantic Representation. This table can act as an aid in the choice of config for a new, unlabelled dataset. For example, if we consider CLINC to be our unlabelled set, we could choose our configuration from this table rather than at random (to make this a fair example, we remove CLINC's results from the table). With little domain knowledge, we can infer that the CLINC utterances contain Mixed intents (both Action-Object and Topics) and estimate that there are a Large number of intents (more than 50). A clustering algorithm could be used to estimate the IR, showing that it is a Balanced set. The dataset size is Large, containing 22,500 utterances which are made up of Short sentences of less than 20 words with a total vocabulary size of 6420 words (Medium). For these features, the table agrees on all-mpnet, KMeans and Action-Object on every feature. There is a disagreement on the Selection Method and so we choose Most Frequent as it is less compute intensive. As shown in Table 4, this is the optimal configuration for CLINC when evaluating on BART Score. Were we to naively choose T0pp Prompting for both Candidate Extraction and Label Selection, in the belief that a more flexible approach would be best, the final labels produced would be of lower quality overall (average BART of -5.0281 compared to -4.5701). Many of the labels generated by this configuration are simply 'ask a question' or in one case 'Yes' for a cluster with $mcgt$ **ingredient_substitution**. Such issues could be overcome with further prompt tuning, however we can already obtain high quality labels from simpler, less hardware and time expensive methods.

7 Conclusions and Future Work

We have shown that our framework for OID can produce high quality labels for many datasets of differing intent type. The modular nature of the framework allows for further improvements to be utilised when new techniques are discovered for each step. We have evaluated a number of configurations based on the final generated label quality, including extending previous OID works which originally do not generate a human-readable intent label. We have also presented an initial analysis of the mapping between dataset features and the optimal configuration to use for a new, unlabelled dataset which can help reduce the initial effort required to choose the combination of techniques. In future work, we plan to add our Action-Object Extension technique (proposed in Anderson et al. (2024)) to the framework and update the optimal configuration results. We also hope to curate more intent datasets of varying features in order to develop a model for predicting a 'best guess' configuration, given a new dataset's features, rather than having to try every one in turn.

Limitations

Our work is limited to the set of techniques chosen for each step in the framework. There exists many other appropriate semantic representation models, clustering algorithms, candidate extraction and selection methods which could possibly produce higher quality labels. Also, the evaluation of the intent labels is based on semantic similarity to the ground-truth labels. This has the implicit assumption that the ground-truth labels are the best representation for the intent which may not necessarily be the case.

Acknowledgments

Our thanks go to our internal reviewers Xinyu Chen and Cynthia Freeman, for their helpful feedback on the first draft. We also thank all anonymous SIGDIAL reviewers for their comments and constructive suggestions.

References

Grant Anderson, Emma Hart, Dimitra Gkatzia, and Ian Beaver. 2024. Automated Human-Readable Label Generation in Open Intent Discovery. In *Proc. INTERSPEECH 2024*, page tbc.

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. Association for Computational Linguistics.

Iñigo Casanueva, Tadas Temcinas, Daniela Gerz, Matthew Henderson, and Ivan Vulic. 2020. Efficient intent detection with dual sentence encoders. *CoRR*, abs/2003.04807.

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.

Ajay Chatterjee and Shubhashis Sengupta. 2020. Intent mining from past conversations for conversational agent. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4140–4152, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Zhiyuan Chen, Bing Liu, Meichun Hsu, Malu Castellanos, and Riddhiman Ghosh. 2013. Identifying intention posts in discussion forums. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1041–1050, Atlanta, Georgia. Association for Computational Linguistics.

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*, abs/1805.10190.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

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.

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, Volume 2 (Short Papers)*, pages 753–757, New Orleans, Louisiana. Association for Computational Linguistics.

Joo-Kyung Kim, Gokhan Tur, Asli Celikyilmaz, Bin Cao, and Ye-Yi Wang. 2016. Intent detection using semantically enriched word embeddings. In *2016 IEEE Spoken Language Technology Workshop (SLT)*, pages 414–419.

Rajat Kumar, Mayur Patidar, Vaibhav Varshney, Lovekesh Vig, and Gautam Shroff. 2022. Intent detection and discovery from user logs via deep semi-supervised contrastive clustering. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1836–1853, Seattle, United States. Association for Computational Linguistics.

Stefan Larson, Anish Mahendran, Joseph J. Peper, Christopher Clarke, Andrew Lee, Parker Hill, Jonathan K. Kummerfeld, Kevin Leach, Michael A. Laurenzano, Lingjia Tang, and Jason Mars. 2019. An evaluation dataset for intent classification and out-of-scope prediction. 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 1311–1316, Hong Kong, China. Association for Computational Linguistics.

Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick Van Kleef, Sören Auer, and Christian Bizer. 2014. Dbpedia - a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web Journal*, 6.

Ting-En Lin, Hua Xu, and Hanlei Zhang. 2019. Discovering new intents via constrained deep adaptive clustering with cluster refinement. *CoRR*, abs/1911.08891.

Bing Liu and Ian Lane. 2016. Attention-based recurrent neural network models for joint intent detection and slot filling.

Pengfei Liu, Youzhang Ning, King Keung Wu, Kun Li, and Helen Meng. 2021. Open intent discovery through unsupervised semantic clustering and dependency parsing. *CoRR*, abs/2104.12114.

Xingkun Liu, Arash Eshghi, Pawel Swietojanski, and Verena Rieser. 2019. Benchmarking natural language understanding services for building conversational agents. *CoRR*, abs/1903.05566.

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. Association for Computational Linguistics.

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*. Association for Computational Linguistics.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H. Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Fevry, Jason Alan Fries, Ryan Teehan, Stella Biderman, Leo Gao, Tali Bers, Thomas Wolf, and Alexander M. Rush. 2021. Multitask prompted training enables zero-shot task generalization.

X.Y. Shen, Ying Sun, Yao zhong Zhang, and Mani Najmabadi. 2021. Semi-supervised intent discovery with contrastive learning. *Proceedings of the 3rd Workshop on Natural Language Processing for Conversational AI*.

Nikhita Vedula, Nedim Lipka, Pranav Maneriker, and Srinivasan Parthasarathy. 2020. Open intent extraction from natural language interactions. In *Proceedings of The Web Conference 2020*, WWW '20, page 2009–2020, New York, NY, USA. Association for Computing Machinery.

Jiaming Xu, Peng Wang, Guanhua Tian, Bo Xu, Jun Zhao, Fangyuan Wang, and Hongwei Hao. 2015. Short text clustering via convolutional neural networks. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pages 62–69, Denver, Colorado. Association for Computational Linguistics.

Weizhe Yuan, Graham Neubig, and Pengfei Liu. 2021. Bartscore: Evaluating generated text as text generation. In *Advances in Neural Information Processing Systems*, volume 34, pages 27263–27277. Curran Associates, Inc.

Hanlei Zhang, Xiaoteng Li, Hua Xu, Panpan Zhang, Kang Zhao, and Kai Gao. 2021a. TEXTOIR: An integrated and visualized platform for text open intent recognition. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 167–174, Online. Association for Computational Linguistics.

Hanlei Zhang, Hua Xu, Ting-En Lin, and Rui Lv. 2021b. Discovering new intents with deep aligned clustering. In *AAAI*.

Junmei Zhong and William Li. 2019. Predicting customer call intent by analyzing phone call transcripts based on cnn for multi-class classification.