Automated Human-Readable Label Generation in Open Intent Discovery

Grant Anderson[1,2], *Emma Hart*[1], *Dimitra Gkatzia*[1], *Ian Beaver*[2]

[1]Edinburgh Napier University, UK
[2]Verint Systems Ltd, USA

grant.anderson@verint.com, emma.hart@napier.ac.uk, dimitra.gkatzia@napier.ac.uk, ian.beaver@verint.com

Abstract

The correct determination of user intent is key in dialog systems. However, an intent classifier often requires a large, labelled training dataset to identify a set of known intents. The creation of such a dataset is a complex and time-consuming task which usually involves humans applying clustering tools to unlabelled data, analysing the results, and creating human-readable labels for each cluster. While many Open Intent Discovery works tackle the problem of discovering clusters of common intent, few generate a human-readable label that can be used to make decisions in downstream systems. To address this, we introduce a novel candidate label extraction method then evaluate six combinations of candidate extraction and label selection methods on three datasets. We find that our extraction method produces more detailed labels than the alternatives and that high quality intent labels can be generated from unlabelled data without resorting to applying costly pre-trained language models.

Index Terms: open intent discovery, label generation, plm prompting

1. Introduction

In the development of modern goal-oriented dialogue systems, a crucial first step is to define the set of user intents that the system needs to be able to recognise. Given these, intent detection is typically viewed as a supervised learning, multi-class classification problem where the user utterances form the input and the model learns the mapping to a set of known intents [1, 2, 3, 4, 5, 6]. However, the task of defining a comprehensive set of intents in real-world applications can be far from trivial. For example, a domain expert may initially be able to successfully cover all intents that currently exist in a set of help-desk dialogues, however, user needs tend to change through time, resulting in an ever growing number of intents. Manual intent discovery would need to be repeated periodically to ensure any emerging intents are identified and handled. Thus, there have been a number of recent works directed towards automatically discovering the intents that are present in the data, without the need for a domain expert.

Solutions to this problem of Open Intent Discovery (OID) seek to discover one or more intents from text utterances which may not have been present in the training data. In many OID works the focus is discovering clusters of similar intent [7, 8, 9, 10, 11], however they do not generate human-readable labels to represent the clusters. If such clusters were to be put to use in production systems, a human analyst would first need to analyse their contents to decide what the intent represents and assign natural language labels to each. The resulting labelled dataset can then be used to train an intent classifier and downstream systems would be able to identify and act on the intent, e.g. a virtual assistant could perform the task that the intent requires. Very few OID works attempt to label the discovered clusters with either a manual [12] or automated means [13, 14].

In this paper, we focus on the generation of high quality labels produced by combining a candidate extraction method with a label selection method. We first propose a new candidate extraction method that addresses the shortcomings of previous work, then evaluate six combinations of extraction/selection methods on three datasets. We find that our extraction method produces detailed, high quality candidates for all evaluated datasets, without requiring the use of a Pre-trained Language Model (PLM). Our key contributions include: (1) We extend the experiments from previous intent label generation work with more challenging datasets and additional techniques. (2) We introduce a new extension[1] to the extraction method used by Liu et al. [14]. (3) We compare our results to prompting a large generative PLM to perform the task.

2. Related work

The state-of-the-art OID techniques use semi-supervised learning, utilising limited labelled data [9, 11]. DSSCC (Deep Semi-Supervised Contrastive Clustering) [11] is the current state-of-the-art in many datasets. Given a set of known intents, techniques such as DSSCC and DeepAligned [9] can estimate the number of unknown intents, using the known intents as a guide for clustering. SCL (Supervised Contrastive Learning) [10] differs from DSSCC and DeepAligned in that the representation model is trained on a labelled dataset in the same domain as the target unlabelled dataset. Unsupervised K-Means clustering is then used on the unlabelled dataset to discover new intents. Chatterjee and Sengupta proposed an unsupervised technique ITER_DBSCAN [12], a variant to the DBSCAN clustering algorithm [15] which allows for unbalanced data distribution.

Thus far, the techniques discussed have not automatically generated a human-readable label for the identified intent clusters. Liu et al. address both intent discovery and label generation, and also approached the problem from a completely unsupervised perspective [14]. Their proposed two-stage technique first finds semantically similar utterances through K-Means clustering, then extracts Action(verb)-Object(noun) pairs using a dependency parser to generate candidate labels. The most frequent candidate is chosen as the final label for the cluster. Evaluation is limited to a single dataset, SNIPS [2], and the clustering stage was accurate. The generated labels were not evaluated by any similarity measure to their ground truth counterparts but the semantic mapping was very clear and would have required little human effort to understand given the low number of intents

[1]https://github.com/GAnderson01/intent-label-generation

(7 intents). However, for a more complex dataset this could be an intensive task, requiring domain knowledge. Vedula et al. [13] 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.

3. Methods

Given the abundance of intent grouping literature, our task assumes a set of clusters of utterances are grouped by similar semantic intent. We then perform automatic label generation to create a natural language label for each cluster. Candidate labels are extracted or generated from the utterances in each cluster, and a label per cluster is chosen from these candidates.

3.1. Candidate label extraction

Conceptually, every textual utterance within an identified cluster could either have a candidate intent extracted or inferred from it. The best technique for finding a candidate will differ for each type of intent that is present in a dataset. For example, an 'actionable intent' in the form of an Action(verb)-Object(noun) pair can be easily extracted by considering the *dobj* rule with a dependency parser. However, a more abstract intent such as 'query' could not be identified in this way. Also, the technique as presented in [14] does not consider compound nouns, negations, or adjectives and so could miss vital context, e.g. "no don't reschedule the delivery" and "yes reschedule the delivery" would both produce the same intent candidate. It will also fail to extract an intent in simple utterances such as 'hello', 'yes please', 'no' or 'thanks'.

We implement three techniques to extract candidates from the identified clusters and compare them:

Action-Object The first extracts Action-Object pairs from utterances as in [14]. An Action-Object pair consists of a verb/infinitive (the Action) and it's target, a noun or subject (the Object). e.g. "I need to reschedule my delivery" contains the Action-Object pair *reschedule-delivery*.

Action-Object Extension We extend Liu et al.'s Action-Object extraction method, by expanding the definition of an Action and an Object. Liu et al. require the Object to have been tagged by a dependency parser as a noun only, we remove this restriction to allow other tags such as proper noun. We use the *compound* and *amod* rules to find compound nouns or descriptive words that apply to the Object. We also utilise the *neg* rule to look for negations attached to the Action. This allows for a more descriptive candidate that takes the form:

> (NEG_)ACTION-
> (ADJECTIVES_)(COMPOUNDS_)OBJECT

where the terms in parentheses are only present if they exist in the utterance.

PLM Prompting The final candidate extraction technique utilises prompting a PLM. Prompting PLMs has shown impressive results in various tasks and could be used to generate a candidate intent, without the restrictions of the other techniques. This makes it a more flexible approach but it requires effort to craft a good prompt. We use a locally deployed PLM to generate a candidate for each utterance with the prompt below. The response from the PLM becomes a candidate intent for the cluster.

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

3.2. Intent label selection

Finally, an intent must be chosen to represent each cluster and provide a natural language label. We implement two selection techniques.

Most Frequent The most common candidate is found from the set of all candidates in a cluster, ignoring any that contain the word 'NONE' to avoid choosing an incomplete Action-Object pair (as in [14]). In the rare occasion that there are no valid candidates, the generated label will be an empty string.

PLM Prompting We also experiment with prompting a PLM at this stage. We prompt a PLM with 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 a string of the utterances in the cluster and [top_3_candidates] are the three most common candidates in the cluster. This instruction should lead the PLM towards choosing one of the candidates but still allows it the freedom to generate something unique if none are appropriate.

4. Experiments

4.1. Experimental setup

Each combination of candidate extraction and intent label selection is executed, and each set of generated labels is evaluated as described in Section 4.3. We refer to a combination of techniques as a *configuration*. The configuration that produced the best result, alongside the corresponding labels and scores are the final output. To obtain initial clusters for each dataset, we use the DeepAligned Clustering technique from [9]. While this is a semi-supervised technique, the intent label generation process is purely unsupervised and generates new labels for all identified clusters.

There are many options available for a PLM. Given that we would be prompting the PLM with an entire dataset multiple times (once for candidate extraction and three times for label selection), we chose a model that can be deployed locally rather than a model such as OpenAI's ChatGPT or Anthropic's Claude whose pricing is based on number of tokens. We chose T0pp for our experiments as it has been shown to produce impressive results [16] but is a small enough model (11 billion parameters) to be deployed on accessible hardware. However, clusters can be very large and the aggregated lengths of the utterance strings may lead to a very long prompt which exceeds the input token limit of T0pp. We therefore build multiple prompts for these large clusters to ensure every utterance is used and the most common response is selected as the final intent label.

For configurations involving prompting, we use a g4dn.12xlarge AWS EC2 instance with 4 NVIDIA T4 GPUs. For all other configurations, a g4dn.xlarge was used.

As a baseline, we use Claude Instant v1.2[2] to perform the entire intent label generation process with the below prompt.

> "Human: Given these utterances: [cluster_utterances] Generate a short two or three words phrase to represent the user intent. \n\nAssistant: Short intent: "

We chose Claude Instant as it has a 100k token context window which was large enough to handle the largest of our clusters of utterances, and at the time of the experiments had the best price/performance ratio of large PLM options available.

[2] https://aws.amazon.com/bedrock/claude/

Table 1: *Features of Each Dataset*

Dataset	Intent Type	No. of Samples	No. of Intents
SNIPS	Action-Object	14484	7
Banking	Action-Object/Topic	13083	77
CLINC	Action-Object/Topic/Question	22500	150

4.2. Datasets

In this section we discuss the datasets used in our experiments and describe their features (Table 1).

SNIPS [2] is an intent dataset containing around 14.5K crowd-sourced natural language utterances that have been gathered from virtual personal assistant commands. These cover 7 Action-Object intents: *PlayMusic*, *AddToPlaylist*, *GetWeather*, *BookRestaurant*, *SearchScreeningEvent*, *SearchCreativeWork*, and *RateBook*. Banking77 [17] is made up of over 13K queries in the banking domain. There are 77 fine-grained intents that include both the Action-Object and topic format e.g. *card_payment_fee_charged*, *request_refund*, *lost_or_stolen_card*, *pin_blocked*, and *cancel_transfer*. CLINC [18] contains 22.5K queries across 10 domains with a total of 150 intents. The intent definition is very loose, with some taking the Action-Object form, others are topics or even specific questions e.g. *update_playlist*, *traffic*, and *how_old_are_you*. The label distribution of both SNIPS and CLINC are balanced, while Banking has an imbalance ratio of 3. All datasets are in English.

4.3. Evaluation

We evaluate each pair of extraction/selection techniques by calculating both the average cosine similarity and the average BARTScore [19] between the normalised ground-truth and generated labels. We normalise by splitting on Pascal/snake case, removing hyphens and converting to lower case. To obtain a vector representation for cosine similarity, we embed the normalised labels using Universal Sentence Encoder [20].

Each cluster is assigned to one of the ground-truth (gt) labels by finding the most common ground-truth ($mcgt$) label in that cluster. We measure the similarity score between the embeddings of the generated label for the cluster and it's $mcgt$ using cosine similarity or BARTScore ($sim(c)$). To obtain a final score for a configuration, we cannot simply take the average of these similarity scores as it is possible that one or more ground-truth labels have not been assigned to a cluster. There is also the possibility that multiple clusters have been assigned the same $mcgt$ but have different generated labels. To handle this, we calculate an average configuration score as follows.

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

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

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

Finally, we take an average of these to obtain our final average similarity score for the configuration (see Equation 2).

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

Table 2: *Scores and timings for each configuration and the Claude baseline. Standard deviations are shown in brackets. Results with * are considered statistically significant when compared with the baseline using a Paired T-test.*[3]

Candidate Extraction	Label Selection	Avg. Cosine Similarity	Avg. BART Score	Time
		SNIPS		
Action-Object	Most Frequent	0.5773 (0.3140)	-4.5967 (2.4445)	1m 59s
Action-Object	T0pp Prompting	0.6904 (0.3135)	-3.6534 (2.2614)*	14m 42s
Action-Object Ext.	Most Frequent	0.6899 (0.3021)	-3.7344 (2.4313)	1m 39s
Action-Object Ext.	T0pp Prompting	**0.7022** (0.2948)	**-3.6065** (2.2460)*	14m 56s
T0pp Prompting	Most Frequent	0.4940 (0.1949)	-5.0576 (2.0696)	3h 39m 59s
T0pp Prompting	T0pp Prompting	0.6043 (0.2578)	-4.2548 (2.4151)	3h 52m 48s
Claude Instant v1.2 Prompting		*0.5220 (0.2870)*	*-5.0056 (1.7390)*	*58s*
		Banking		
Action-Object	Most Frequent	**0.3974** (0.2438)*	-5.6216 (1.2992)*	1m 43s
Action-Object	T0pp Prompting	0.2648 (0.2581)*	-6.1862 (1.4438)*	16m 23s
Action-Object Ext.	Most Frequent	0.3816 (0.2848)*	**-5.4319** (1.6261)*	1m 30s
Action-Object Ext.	T0pp Prompting	0.2962 (0.2731)*	-6.0337 (1.5213)*	16m 21s
T0pp Prompting	Most Frequent	0.3495 (0.2828)*	-5.6323 (1.6879)*	2h 50m 1s
T0pp Prompting	T0pp Prompting	0.2766 (0.2630)*	-6.0236 (1.6680)*	3h 4m 5s
Claude Instant v1.2 Prompting		***0.5518***	***-4.4626***	*1m 20s*
		CLINC		
Action-Object	Most Frequent	**0.5502** (0.2751)	**-4.4197** (1.6293)*	2m 47s
Action-Object	T0pp Prompting	0.5273 (0.2667)*	-4.5449 (1.5749)	20m 2s
Action-Object Ext.	Most Frequent	0.4911 (0.2998)*	-4.9655 (1.9709)	2m 32s
Action-Object Ext.	T0pp Prompting	0.5001 (0.2936)*	-4.9461 (1.8958)	19m 59s
T0pp Prompting	Most Frequent	0.4787 (0.2898)*	-4.7637 (1.7488)	5h 7m 13s
T0pp Prompting	T0pp Prompting	0.4348 (0.2724)*	-5.0062 (1.7494)	5h 25m 18s
Claude Instant v1.2 Prompting		***0.5855***	*-4.8130*	*2m 13s*

where GT is the set of all ground-truth intents and N_{GT} is the number of ground-truth intents. We score each possible configuration for each dataset and find it's optimal configuration (best *config_score*).

5. Results and analysis

Table 2 shows the average cosine similarity, average BARTScore and end-to-end timings for each configuration and dataset. In Table 3 the final labels generated for the best configurations are shown in italics. Alongside these we show the alternate Action-Object extraction method, to highlight the benefits of our extension. A sample of 10 labels are shown for Banking and CLINC due to space restrictions.

We can see that the labels produced are of very good quality considering the label generation process uses completely unlabelled data. For SNIPS, DeepAligned clustering has clustered the intents correctly, with 7 clusters identified, each with a unique most common ground truth. Both metrics agree that Action-Object Extension combined with T0pp Prompting produces the highest quality labels. Three generated labels, *rate-book, Play-music* and *book-restaurant*, were identical to their ground truths. *add-song, Find-movie_schedule* and *Tell-weather_forecast* are semantically the same as their ground truths. *Find-TV_show* is similar to **SearchCreativeWork**, however, it may be too specific to be of use without some human validation and modification.

For Banking, while 77 clusters are identified, only 68 intents are assigned as the most common ground truths as 8 intents were assigned to 2 or 3 clusters. The metrics disagree on the best configuration. Action-Object with Most Frequent achieved the highest cosine similarity score while Action-Object Extension with Most Frequent scores best according to BARTScore. Many of the generated labels are clearly semantically similar to the most common ground truth, e.g. in both configurations, *verify-source* was generated for **verify_source_of_funds**

[3] https://en.wikipedia.org/wiki/Student%27s_t-test

Table 3: *Comparison between the labels produced using Action-Object Extension and Action-Object extraction paired with the label selection method from the datasets best configuration. Label sets in italics scored the highest in one or both metrics.*

Most Common GT	Generated Label w/Action-Object Ext.	Generated Label w/Action-Object
	SNIPS	
PlayMusic	*Play-music*	play-music
SearchCreativeWork	*Find-TV_show*	find-show
RateBook	*rate-book*	rate-book
SearchScreeningEvent	*Find-movie_schedule*	find-schedule
BookRestaurant	*book-restaurant*	book-restaurant
AddToPlaylist	*add-song*	add-song
GetWeather	*Tell-weather_forecast*	give-forecast
	Banking	
getting_virtual_card	*get-virtual_card*	*get-card*
verify_source_of_funds	*verify-source*	*verify-source*
verify_my_identity	*need-What*	*do-check?*
passcode_forgotten	*reset-password*	*reset-password?*
get_disposable_virtual_card	*get-disposable_virtual_card*	*get-card?*
card_payment_fee_charged	*charge-fee*	*charged-fee*
card_arrival	*receive-card*	*receive-card?*
request_refund	*get-refund*	*get-refund*
edit_personal_details	*change-name*	*change-name*
exchange_charge	*exchange_currencies*	*exchanging-currencies?*
	CLINC	
transactions	show-transactions	*show-transactions*
play_music	play-song	*play-song*
schedule_meeting	schedule-meeting	*schedule-meeting*
plug_type	need-socket_converter	*need-converter*
oil_change_when	change-oil	*change-oil*
how_old_are_you	tell_me	*tell-age*
text	send-text	*send-text*
pto_balance	put-pto_request	*have-days*
who_do_you_work_for	say-who	*take-orders*
improve_credit_score	improve-credit_score	*improve-score*

and *reset-password* for **passcode_forgotten**.

For CLINC, 150 clusters are identified by DeepAligned, but only 144 intents are assigned as a most common ground truth as 6 intents were assigned to multiple clusters. The metrics agree that Action-Object paired with Most Frequent is the optimal configuration. Again, we have generated labels which are very semantically similar to the most common ground truth in many clusters e.g. *show-transactions* for **transactions**, *play-song* for **play_music** and *schedule-meeting* for **schedule_meeting**.

Table 3 highlights the benefits of our Action-Object Extension method even in the cases where it does not produce the best similarity metric scores. Consider the fine-grained intent **get_disposable_virtual_card** in Banking, with simple Action-Object candidates the chosen label is too generic and contains punctuation: *get-card?*. With Action-Object Extension we get a complete, detailed label which matches the ground truth. It is also worth noting that the configuration with the highest cosine similarity scores only marginally higher than the configuration using Action-Object Extension instead. We see the same lack of detail in CLINC labels e.g. Action-Object produces *improve-score* for **improve_credit_score** while Action-Object extension matches the true label. Given the true labels, we can see the semantic similarity between the labels generated with Action-Object, however, if we had not known the ground truths, it would be difficult to understand the true intent of some clusters without further investigation. There was one case where Action-Object produced zero candidate intents for a cluster with *mcgt*: **no**, and so the generated label was an empty string.

Our strong baseline, Claude Instant v1.2, is able to generate high quality labels, however it did not always achieve the best scores. We see a large difference in SNIPS where Claude achieves the second worst quality labels. It appears to generate labels that are not generic enough to represent the intent, e.g. *Play classics* for **PlayMusic** and *Alana Davis add* for **AddToPlaylist**. In the CLINC dataset, Claude achieves the highest average cosine similarity (by a small margin), but not the best BARTScore. Although we prompted Claude to generate a short two or three word phrase, it breaks that condition often and generates multiple intents rather than just one e.g. *Check transaction, recent transaction, transaction history* for **transactions**. Claude will also sometimes continue to generate another 'Human' message, e.g. *confirm identity Human: Okay, here are some* was generated for a cluster with *mcgt* **are_you_a_bot**. Claude achieves the best scores in Banking by a large margin. This improvement over other configurations is quite striking compared to the other datasets. Our hypothesis is that Banking77 was included in Claude's training dataset but we cannot verify this as its pre-training datasets were not made public. Again, Claude suffers from the issue of generating multiple intents for a single cluster in Banking as well e.g. *Troubleshoot contactless\nReset password* for **passcode_forgotten**.

Prompting T0pp was only in the top configuration for one dataset. In SNIPS, it is only used for the final step, in choosing the label from the candidates. This is not too surprising, as SNIPS, Banking and CLINC are predominantly made up of either commands to a personal assistant or specific requests for help, meaning there are plenty of Action-Object candidates to choose from. Inspecting the generated labels in Banking when using Prompting for extraction and Most Frequent for selection shows some very good and detailed intents, such as *get a disposable virtual card* for **get_disposable_virtual_card** and *verify the source of funds* for **verify_source_of_funds**, however in many cases it fails to capture the intent and seems to generate something generic such as *ask a question*. It is possible that the labels could be improved with more rigorous prompt tuning, however it is encouraging to see that high quality labels can be generated, completely unsupervised, without having to resort to querying a PLM. Candidate extraction using T0pp had the longest duration of all techniques, taking almost 3 hours to complete for the smallest dataset. This may have been in part due to the fact that it was deployed across 4 GPUs as we were unable to obtain a single GPU device with sufficient memory.

6. Conclusions and future work

We have presented experimental results for different combinations of techniques in automated human-readable label generation for Open Intent Discovery. We introduced an extension to the Action-Object extraction technique used in [14], which can extract more detailed and usable intents. We also performed additional experiments on more challenging datasets than previously tested. Our results show that high quality, human-readable intent labels can be generated for an unlabelled dataset without requiring the use of an expensive PLM. Future work may include expanding the evaluation datasets to either further validate our findings or apply the approaches to languages other than English. One limitation to our approach is we assume the existing ground truth intent label is the best representation, therefore any metric based on similarity will naturally score higher the more similar the generated label is. Therefore, a technique for intent label generation could generate a label that better represents the cluster but differs in content from the ground truth, and thus receive a worse score according to the metrics. Further investigation of alternative automated evaluation metrics and a human evaluation would be beneficial.

7. References

[1] Z. Chen, B. Liu, M. Hsu, M. Castellanos, and R. Ghosh, "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*, L. Vanderwende, H. Daumé III, and K. Kirchhoff, Eds. Atlanta, Georgia: Association for Computational Linguistics, Jun. 2013, pp. 1041–1050. [Online]. Available: https://aclanthology.org/N13-1124

[2] A. Coucke, A. Saade, A. Ball, T. Bluche, A. Caulier, D. Leroy, C. Doumouro, T. Gisselbrecht, F. Caltagirone, T. Lavril, M. Primet, and J. Dureau, "Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces," *CoRR*, vol. abs/1805.10190, 2018. [Online]. Available: http://arxiv.org/abs/1805.10190

[3] C.-W. Goo, G. Gao, Y.-K. Hsu, C.-L. Huo, T.-C. Chen, K.-W. Hsu, and Y.-N. Chen, "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)*, M. Walker, H. Ji, and A. Stent, Eds. New Orleans, Louisiana: Association for Computational Linguistics, Jun. 2018, pp. 753–757. [Online]. Available: https://aclanthology.org/N18-2118

[4] J.-K. Kim, G. Tur, A. Celikyilmaz, B. Cao, and Y.-Y. Wang, "Intent detection using semantically enriched word embeddings," in *2016 IEEE Spoken Language Technology Workshop (SLT)*, 2016, pp. 414–419.

[5] B. Liu and I. Lane, "Attention-based recurrent neural network models for joint intent detection and slot filling," 2016.

[6] J. Zhong and W. Li, "Predicting customer call intent by analyzing phone call transcripts based on cnn for multi-class classification," 2019.

[7] H. Perkins and Y. Yang, "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)*. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 4016–4025. [Online]. Available: https://aclanthology.org/D19-1413

[8] T. Lin, H. Xu, and H. Zhang, "Discovering new intents via constrained deep adaptive clustering with cluster refinement," *CoRR*, vol. abs/1911.08891, 2019. [Online]. Available: http://arxiv.org/abs/1911.08891

[9] H. Zhang, H. Xu, T.-E. Lin, and R. Lv, "Discovering new intents with deep aligned clustering," in *AAAI*, 2021.

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

[11] R. Kumar, M. Patidar, V. Varshney, L. Vig, and G. Shroff, "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*. Seattle, United States: Association for Computational Linguistics, Jul. 2022, pp. 1836–1853. [Online]. Available: https://aclanthology.org/2022.naacl-main.134

[12] A. Chatterjee and S. Sengupta, "Intent mining from past conversations for conversational agent," in *Proceedings of the 28th International Conference on Computational Linguistics*. Barcelona, Spain (Online): International Committee on Computational Linguistics, Dec. 2020, pp. 4140–4152. [Online]. Available: https://aclanthology.org/2020.coling-main.366

[13] N. Vedula, N. Lipka, P. Maneriker, and S. Parthasarathy, "Open intent extraction from natural language interactions," in *Proceedings of The Web Conference 2020*, ser. WWW '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 2009–2020. [Online]. Available: https://doi.org/10.1145/3366423.3380268

[14] P. Liu, Y. Ning, K. K. Wu, K. Li, and H. Meng, "Open intent discovery through unsupervised semantic clustering and dependency parsing," *CoRR*, vol. abs/2104.12114, 2021. [Online]. Available: https://arxiv.org/abs/2104.12114

[15] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "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*, ser. KDD'96. AAAI Press, 1996, p. 226–231.

[16] V. Sanh, A. Webson, C. Raffel, S. H. Bach, L. Sutawika, Z. Alyafeai, A. Chaffin, A. Stiegler, T. L. Scao, A. Raja, M. Dey, M. S. Bari, C. Xu, U. Thakker, S. S. Sharma, E. Szczechla, T. Kim, G. Chhablani, N. Nayak, D. Datta, J. Chang, M. T.-J. Jiang, H. Wang, M. Manica, S. Shen, Z. X. Yong, H. Pandey, R. Bawden, T. Wang, T. Neeraj, J. Rozen, A. Sharma, A. Santilli, T. Fevry, J. A. Fries, R. Teehan, S. Biderman, L. Gao, T. Bers, T. Wolf, and A. M. Rush, "Multitask prompted training enables zero-shot task generalization," 2021.

[17] I. Casanueva, T. Temcinas, D. Gerz, M. Henderson, and I. Vulic, "Efficient intent detection with dual sentence encoders," *CoRR*, vol. abs/2003.04807, 2020. [Online]. Available: https://arxiv.org/abs/2003.04807

[18] S. Larson, A. Mahendran, J. J. Peper, C. Clarke, A. Lee, P. Hill, J. K. Kummerfeld, K. Leach, M. A. Laurenzano, L. Tang, and J. Mars, "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)*, K. Inui, J. Jiang, V. Ng, and X. Wan, Eds. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 1311–1316. [Online]. Available: https://aclanthology.org/D19-1131

[19] W. Yuan, G. Neubig, and P. Liu, "Bartscore: Evaluating generated text as text generation," in *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds., vol. 34. Curran Associates, Inc., 2021, pp. 27 263–27 277. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2021/file/e4d2b6e6fdeca3e60e0f1a62fee3d9dd-Paper.pdf

[20] D. Cer, Y. Yang, S. Kong, N. Hua, N. Limtiaco, R. S. John, N. Constant, M. Guajardo-Cespedes, S. Yuan, C. Tar, Y. Sung, B. Strope, and R. Kurzweil, "Universal sentence encoder," *CoRR*, vol. abs/1803.11175, 2018. [Online]. Available: http://arxiv.org/abs/1803.11175