Going beyond research datasets: Novel intent discovery in the industry setting

Aleksandra Chrabrowa, Tsimur Hadeliya, Dariusz Kajtoch,
Robert Mroczkowski, Piotr Rybak
ML Research at Allegro, Poznań, Poland
{firstname.lastname}@allegro.pl

Abstract

Novel intent discovery automates the process of grouping similar messages (questions) to identify previously unknown intents. However, current research focuses on publicly available datasets which have only the question field and significantly differ from real-life datasets. This paper proposes methods to improve the intent discovery pipeline deployed in a large e-commerce platform. We show the benefit of pre-training language models on in-domain data: both self-supervised and with weak supervision. We also devise the best method to utilize the conversational structure (i.e., question and answer) of real-life datasets during fine-tuning for clustering tasks, which we call *Conv*. All our methods combined to fully utilize real-life datasets give up to 33pp performance boost over state-of-the-art Constrained Deep Adaptive Clustering (CDAC) (Lin et al., 2020) model for question only. By comparison CDAC model for the question data only gives only up to 13pp performance boost over the naive baseline.

1 Introduction

Allegro is one of largest the e-commerce marketplace in Central Eastern Europe region that connects buyers and merchants. It has millions of active users. Therefore, the good functioning of the Customer Experience (CX) department is crucial as it provides the necessary support, resolves emerging issues, and answers user questions.

Task-oriented chatbots relieve humans by automatically resolving the most repetitive and trivial issues. They usually have a pre-defined set of user intents with matching template answers. Then, when a user asks a question, the intent classifier detects the question intent and returns the matching response. Creating a reliable and comprehensive chatbot requires massive work to discover, define, and maintain a set of intents with training examples. With the continuous development of marketplace platforms, new intents constantly appear as new features are introduced. Therefore, the automated intent discovery system becomes a critical component.

Novel intent discovery is performed offline on historical data. In the context of personalized intelligence assistants existing approaches (Lin et al., 2020; Gao et al., 2021; Vedula et al., 2022) focus on learning transferable features with utterance encoders that guide the discovery on unlabeled data with a handful of labeled examples belonging to known intents. However, at *Allegro* our main communication form is emails, and we have access to much richer conversational data that can improve discovery performance. A large body of historical conversational data (user questions and consultants' answers) can be leveraged in two ways. Firstly, to better initialize message encoders and secondly by performing intent discovery on conversational data as an additional signal. Additionally, a form of weak supervision is available: keywords (or tags) added by the consultants that help them understand past cases.

The paper's main contribution is the demonstration that incorporating additional signals like conversational structure or weak labels into the existing intent discovery method results in better overall performance. We pre-trained for domain adaptation three encoders using conversational data and weak labels. We devised *Conv*, a method for fine-tuning on conversational data (i.e., question and answer) for the clustering task using a three-headed encoder. To the best of our knowledge, this result was not reported in the public literature.

2 Related Work

2.1 Discovering novel intents

The goal of novel intent discovery is to identify groups of similar utterances in unlabeled data with the assistance of limited labeled data. The Con-

Findings of the Association for Computational Linguistics: EACL 2023, pages 925–941
May 2-6, 2023 ©2023 Association for Computational Linguistics

strained Deep Adaptive Clustering (Lin et al., 2020, CDAC) uses dense intent representation on top of the pre-trained BERT backbone to learn similarity functions in a semi-supervised contrastive manner. It is then utilized in the clustering algorithm. In a real-world scenario of personal assistants (Gao et al., 2021; Vedula et al., 2022) use a pre-trained BERT model as a backbone encoder with supervised contrastive learning to transfer distance function to unlabeled data for clustering. Unlike this work, the authors use only the question field and English *BERT-base* uncased model for initialization. They do not use in-domain unlabeled data or weak supervision for backbone pre-training.

2.2 Transfer learning

General-purpose pre-trained encoders like BERT are not ideal. Tasks involving domain-specific texts like, e.g., science corpus, clinical notes, or e-commerce product descriptions benefit more from additional pre-training on in-domain data due to better suited vocabulary and word embeddings to domain specific problems (Beltagy et al., 2019; Huang et al., 2019; Tracz et al., 2020; Gururangan et al., 2020). Similarly, for conversational tasks *ConveRT* (Henderson et al., 2020a) substantially outperforms BERT in neural response selection. Additionally, industrial-scale training on weakly supervised datasets leads to improvements in several NLP tasks (Bach et al., 2018).

3 Method

3.1 Problem statement

Given unlabeled instances $\mathcal{D}$, the goal is to automatically cluster utterances into $\mathcal{I}$ classes, which are not known *a priori*. We also assume that we are given labeled instances $\mathcal{D}^k$ with $\mathcal{I}_k$ known set of intents and $\mathcal{I} \cap \mathcal{I}_k \neq \emptyset$. Unlabeled instances may belong to both known intents $\mathcal{I}_k$ and unknown ones $\mathcal{I}_u = \mathcal{I} \setminus \mathcal{I}_k$.

3.2 Framework overview

Our novel intent discovery framework consists of representation learning (Bengio et al., 2013) and subsequent clustering with K-means (Lloyd, 1982). We propose the following to improve text representations for real-life novel intents discovery in the communication domain:

- Efficient initialization with pre-trained encoders, adapted to the e-commerce domain by optimization for weak training signals and conversational structure of the data.
- Fine-tuning for the clustering task with state-of-the-art training scheme (i.e., CDAC) adapted to use all the conversational data (i.e., question and answer). *Conv* is our proposed method to train a conversation structure-aware encoder with three-headed architecture.

In the following sections, we describe each component in more detail.

3.3 Initialization

An essential step in the deep learning process is initialization. Proper initialization is crucial in training representations for discovering new intents with clustering. The effectiveness of the existing clustering algorithms depends heavily on the quality of the representation encoder. In this work, we identified this dependency and proposed a generic approach for an efficient encoder pre-training in the conversational domain.

3.3.1 Domain specific data structure

We operate in the e-commerce domain with a two-sided marketplace. Customers can seek support by exchanging messages via email or chat. The former are typically longer and include a more formal boilerplate. A dialog may be held between merchants and CX support, buyers and CX support, and directly between buyers and merchants. All messages are written in Polish.

3.3.2 Domain adaptation

We prepared two self-supervised models based on *BERT-base* (Devlin et al., 2019) architecture. We started from a general domain encoder *HerBERT* (Mroczkowski et al., 2021). We used a training corpus of 68M conversation threads with 184M messages and 8314M words. We included both emails and chats exchanged between all parties (merchants, CX support, and buyers).

- ***AlleBERT*** is *HerBERT*fine-tuned with Masked Language Model (MLM) objective.
- ***AlleConveRT*** is *AlleBERT* further fine-tuned on the same dataset but with the mixture of MLM and Conversational Contrastive Loss (CCL) (Henderson et al., 2020b).

The details of the training procedure for each of the pre-trained encoders can be found in Appendix E.

3.3.3 Weak supervision

In the case of email communication exchanged with CX support, every message includes at least one of 512 tags. These labels roughly identify the problem solved. They are assigned by CX consultants often in a noisy manner. We utilized this weak signal and prepared ***TagBERT*** encoder in a two-stage process. Firstly, we finetuned *HerBERT* with MLM and Message Threads Structural Objective (MTSO) (Wang et al., 2020) on all internal communication data (emails and chats). Secondly, we finetuned it on a multi-label classification task on *CX weakly supervised* dataset that includes 2.5M messages in the email domain exchanged between merchants or buyers and CX support. Details of the training procedure can be found in Appendix E.

3.4 *Conv*, conversation structure aware encoder

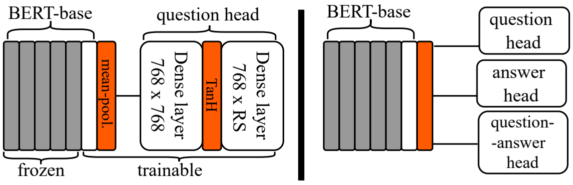

Figure 1: Representation model based on BERT-base encoder used in the discovery pipeline. On the left version with one head. On the right *Conv*, our conversational model with three separate trainable heads for the question, answer, and question-answer concatenation. The parameters of the encoder are frozen except for the last transformer block.

As depicted in Fig. 1, we used an encoder with *BERT-base* architecture (Devlin et al., 2019) followed by an average pooling[1] and three projection heads with two linear layers and *Tanh* non-linearity in between (Lin et al., 2020).

The three-headed model works with conversational input containing a pair of texts: the user's question and the consultant's answer[2]. Two heads project each input separately, and the third one handles additional signals from the question-answer concatenation into one string of text. Each of the inputs is fed into encoder separately. A common underneath encoder is updated jointly with a gradient from all heads from the total loss given by the weighted average of losses for each head:

$$\begin{aligned}\mathcal{L}_{Conv}(X, Y, \theta) = &\lambda_{\mathrm{Q}} \cdot \mathcal{L}(X_{\mathrm{Q}}, Y, \theta_{\mathrm{Q}}) \\ &+ \lambda_{\mathrm{A}} \cdot \mathcal{L}(X_{\mathrm{A}}, Y, \theta_{\mathrm{A}}) \\ &+ \lambda_{\mathrm{QA}} \cdot \mathcal{L}(X_{\mathrm{QA}}, Y, \theta_{\mathrm{QA}}).\end{aligned} \tag{1}$$

Here $X = (X_{\mathrm{Q}}, X_{\mathrm{A}}, X_{\mathrm{QA}})$ is the array of inputs (all examples), i.e. all questions, all answers, all question-answer concatenations respectively. Y are the input labels[3]. $\theta = (\theta_{\mathrm{Q}}, \theta_{\mathrm{A}}, \theta_{\mathrm{QA}})$ is the array of parameter sets for individual inputs *BERT-base*parameters are shared as depicted in Figure 1. The hyperparameters $\lambda = (\lambda_{\mathrm{Q}}, \lambda_{\mathrm{A}}, \lambda_{\mathrm{QA}})$ govern how conversational structure is utilized for any choice of the training scheme, whereas the precise form of the loss terms $\mathcal{L}$ depends on the choice of the training scheme described in Sec. 3.5. For example if we choose $\lambda = (1, 0, 0)$, and compute $\mathcal{L}$ according to CDAC training scheme, we follow the original CDAC setup with the question field only. By using $\lambda = (0, 0, 1)$ and computing $\mathcal{L}$ according to CDAC training scheme, we effectively only concatenate question and answer strings and feed it into the model instead of the question string.

In our method *Conv* for training conversation structure-aware encoder, we trained the representation encoder with uniform heads contribution $\lambda = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$ staring from initializations described in Section 3.3. The final representation used for clustering is an embedding from the head for question-answer concatenation.

To speed up training with large batches, we kept the weights of the encoder frozen except for the last transformer layer. The first linear layer keeps the *BERT-base* dimension of the representations (i.e., 768). The second linear block output dimension is a representation size hyperparameter.

3.5 Training scheme

Up to this point, we are able to use any framework for finetuning the representation encoder for intent discovery with clustering. With that said, we propose to use two potential approaches for real-world CX communication data.

Static. In a setup where we do not have any labeled data available, we extract text representation from the pre-trained encoder by average pooling without additional training.

[1] Unlike many implementations, the hidden states for padding tokens are not averaged.

[2] While encoding question and answer, are preceded with special tokens for question and answer.

[3] Since we deal with unsupervised/semi-supervised algorithms, some examples are unlabelled.

Constrained DAC (CDAC) (Lin et al., 2020). The method generalizes the Deep Adaptive Clustering (DAC) (Chang et al., 2017) scheme for partially labeled data and trains with a contrastive loss on both distance-based pseudo-pairs and exact pairs given by intent labels. It is semi-supervised since it utilizes both labeled and unlabeled examples from the train set. We adapted CDAC training scheme to *Conv*, our three-headed, conversation structure-aware encoder (see Sec. 3.4). Details of the DAC method are in Appendix B.1, and details of the CDAC method are in Appendix B.2.

4 Evaluation

We describe our experimental setup for novel intent discovery. We prove the efficiency of the proposed method on real-world communication datasets. To verify gains from different framework components, we present more results in the ablation section (Sec. 5).

4.1 Real-world internal datasets

We used three internal datasets: *Purchase*, *Delivery* and *Retail* from real traffic to CX support at *Allegro* in Polish language. CX consultants manually annotated the datasets with intent labels. Categories of email queries to the CX team are more fine-grained than the widely used *Banking77* (Casanueva et al., 2020) dataset. Moreover, such real-world datasets are highly imbalanced, with some intents overlapping. Basic dataset statistics are shown in the Table 1. The user emails vary in length and style and may contain irrelevant parts. Each dataset includes messages of different quality and specificity ranging from uninformative chit-chat to well-written ones. In datasets, only the first question and direct answer are included, and all further messages from the correspondence thread are omitted. The *Purchase* and *Delivery* cover conversations between buyers and CX consultants. *Retail* is communication between buyers and merchants, so conversation topics and structure are different. We use a stratified 80/10/10 train/val/test split.

We use two public benchmark English datasets from task-oriented dialog systems: *CLINC150* (Larson et al., 2019) and *Banking77* (Casanueva et al., 2020) in Dataset splits follow exactly the experimental setup used in (Zhang et al., 2020) in ablation study in Section 5.2 to increase the reproducibilty of our work. In other ablations it is impossible due to missing conversational and weak label signal.

Basic statistics of the datasets are in the Table 1. Further details are in Appendix A.

4.2 Experimental setting

We build a controlled open-world intent discovery setup, following the setup proposed in (Lin et al., 2020; Zhang et al., 2020). We prepared novel intents by randomly masking all examples from 50% of intents in the training set. The remaining intents serve as known intents and are additionally partially masked. We masked 50% of all remaining examples. We apply the representation learning framework: we take in-domain encoders described in Section 3.3.2 and 3.3.3 and do the fine-tuning step (described in Section 3.4 and 3.5). After the training phase, we cluster the whole test dataset with K-means. We performed clustering with the ground truth number of clusters (i.e., the number of intents in the dataset).

We run experiments with hyperparameters (i.e., representation size, batch size, and learning rate) fixed. We have described the method of their selection in Appendix D.

We use five random seeds, which govern intent masking and weight initialization. We train the model for 100 epochs on a single machine with NVIDIA V100 GPU. It takes a few hours to run a single fine-tuning experiment for all seeds for a single setting (dataset, training scheme etc.).

4.3 Metric[4].

We compute metrics based on cluster ids from K-means algorithm and ground truth labels. The discovery quality is probed with three standard clustering metrics, i.e., Accuracy (ACC) using the Hungarian algorithm, Normalized Mutual Information (NMI), and Adjusted Rand Index (ARI). We also introduce two additional metrics. First, the *binary F1-score* i.e., macro F1-score with a majority vote on cluster label calculated on the whole dataset where all known intents are one class, and all novel intents are the second class. Second, the *macro F1-score* with a majority vote on the cluster label. It turns the clustering quality problem into a multi-label classification. In the main part of the paper, we report **AVG** i.e., the average of five metrics over all seeds. AVG increases with clustering quality up to 100%. AVG is the primary metric used for

[4] We publish the code for our metrics: https://github.com/allegro/ml/tree/main/publications/intent-discovery-metrics/

Dataset	# intents	# examples	# examples per intent mean	min	max	entropy	mean length (characters) question	answer
Banking77	77	13.1k	170±33	75	227	0.992	60±40	-
CLINC150	150	22.5k	150±0	150	150	0.999	40±20	-
Purchase	22	2.7k	121±50	29	240	0.972	320±280	1060±400
Delivery	23	3.0k	130±55	57	221	0.973	330±360	860±410
Retail	105	13.8k	133±124	22	664	0.930	160±190	740±830

Table 1: Downstream tasks datasets characteristic. Class imbalance is measured by the average number of examples per intent and the normalized Shannon's entropy of the intent distribution (which is 1 for for the perfectly balanced case and lower in case of class imbalance). Further details are in Appendix A

Method	*Purchase*	*Delivery*	*Retail*
Static	37.0±4.1	31.1±1.3	28.8±0.7
CDAC	50.2±6.6	40.9±4.5	36.5±1.7
Our	**83.2±3.2**	**64.2±6.3**	**45.4±4.0**

Table 2: Static baseline and CDAC representations compared with our framework on novel intent discovery task for real-world data. Our framework combines *TagBERT*pre-trained encoder, CDAC training scheme, and *Conv* method for using the conversation structure. AVG metric averaged over five seeds.

Initialization	*Purchase*	*Delivery*	*Retail*
HerBERT	65.9±6.2	44.7±3.7	37.2±2.0
AlleBERT	66.4±6.6	49.2±6.4	44.2±2.2
AlleConveRT	73.1±8.8	57.9±5.9	**49.3±2.1**
TagBERT	**83.2±3.2**	**64.2±6.3**	45.4±4.0

Table 3: Impact of initialization for novel intent discovery task. *Conv* conversation structure-aware encoder was trained with the CDAC scheme from different initialization. AVG metric averaged over five seeds with standard deviation.

model selection. Additionally, to facilitate comparison with other research, the five metrics are listed separately in Appendix F for all experiments. In Appendix F we give more details on how we compute metrics or test for statistical significance.

4.4 Results

Table 2 shows the AVG metric for our best-performing model. Five individual metrics are listed in Table 8. We significantly improve intent discovery compared with baselines. *Our* model uses *TagBERT* (see Section 3.3.3) as initialization and is trained with the CDAC scheme. While training, we used both question and answer fields and utilized conversational structure-aware encoder *Conv* introduced in Sec. 3.4. The baselines (*Static* and *CDAC*) are based on the general domain *HerBERT* encoder and use the question field only. We improved over the second-best CDAC, depending on the dataset, by 8.9pp to 33pp. The performance gap of *our* framework to the *CDAC* baseline is greater then the superiority of *CDAC* over the naive baseline, *static* embeddings, which is between 7.7pp and 13.2pp.

5 Ablation

We attribute the improvement in performance to all three method components: domain adaptation during pre-training with conversational and weak label signal, state-of-the-art training scheme CDAC, and leveraging of conversation structure with our *Conv* method introduced in Section 3.4.

5.1 Initialization

In this section, we show the effect of initialization on the novel intent discovery task. We trained a conversation structure-aware encoder with a CDAC scheme using four different initializations.

AVG metric is reported in Table 3 and individual metrics are shown Table 9. Comparing *AlleBERT* with *HerBERT*, we can see that domain-adapted initialization improves 1 to 7pp for discovering new intents. Further adaptation of the starting encoder with the loss of ConveRT improves at least 5pp. Summarizing *AlleBERT* and *AlleConveRT* initializations bring gains for all internal datasets. For the CX domain (*Purchase* or *Delivery*), the best initialization was provided by *TagBERT*. Pre-training with weak labels introduced additional training information that turned out to be transferable for the

Training scheme	*Banking77*	*CLINC150*	*Purchase*	*Delivery*	*Retail*
Static	41.7±1.0	55.9±1.4	35.5±4.1	31.0±2.4	29.6±0.8
DAC	51.8±1.8	64.6±1.3	24.1±0.7	24.0±0.9	27.3±4.4
Supervised	**65.2±2.1**	**73.2±0.6**	38.2±2.1	33.5±2.2	30.1±0.5
CDAC	61.8±2.8	70.4±1.4	**52.9±7.3**	**42.3±3.6**	**39.2±1.2**

Table 4: Evaluation of training schemes for novel intent discovery. We report AVG metric averaged over five seed with standard deviation. Models use *BERT-base* (English datasets) or *AlleBERT* (Polish datasets) encoder and question input only. The best results are in bold.

downstream task. The simultaneous drop in quality on the *Retail* dataset originating from the domain for which we did not have noisy labels confirms this phenomenon.

5.2 Training schemes

We compare two training schemes *Static*, and *CDAC* from Sec. 3.5 with two additional baseline methods *DAC* and *Supervised*. For *Supervised* training scheme, we use Large Margin Cosine Loss (LMCL) (Wang et al., 2018) to learn representation from labels. We discard unlabeled data from the train set. We train the models for all four schemes with question input only and *BERT-base* (Devlin et al., 2019) for English and *AlleBERT* for Polish datasets.

This ablation study is the only case when we can use two public benchmark English datasets from task-oriented dialog systems: *CLINC150* (Larson et al., 2019) and *Banking77* (Casanueva et al., 2020). Unfortunately, public benchmark datasets lack the answer data, a large amount of unlabeled data, and weak labels. However, including them in this ablation study increases the reproducibility of our work and brings interesting insights.

AVG metric is reported in Table 4 and individual metrics can be found in Table 10. For all datasets, there is a gain from using intent labels (*Supervised* and *CDAC*). For public datasets among unsupervised methods, DAC outperforms static representations. However, supervised training is better than semi-supervised CDAC. The results are the opposite for the internal datasets. DAC is better than static representations, and semi-supervised CDAC is better than supervised training. We hypothesize that different real-world and benchmark datasets results might be due to dataset quality and size differences. In general, benchmark datasets are larger and more balanced. Moreover, mail messages from real-world e-commerce are longer and noisier on average. It is an open question how this trend holds for other real-life datasets.

To sum up, there is a gain from intent labels for all datasets. Optimal solutions for public benchmarks and real-world internal datasets differ. CDAC is the best training scheme that uses intent labels for internal datasets.

5.3 Conversational structure

We examine if any further gains in performance can be obtained from incorporating the answer field signal. We conduct experiments only on the internal datasets. We use only the best training scheme, i.e., *CDAC*. We examine four training configurations: only question representation Q trained with $\lambda = (1, 0, 0)$, only answer representation A trained with $\lambda = (0, 1, 0)$, question-answer concatenation *QA concatenation* trained with $\lambda_3 = (0, 0, 1)$, using question and answer in a simpler two-headed model *QA two heads* trained with $\lambda = (\frac{1}{2}, \frac{1}{2}, 0)$ and full three-headed conversational model *Conv* trained with $\lambda = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$ described in detail in section Sec. 3.4.[5]

AVG metric is reported in Table 5 and individual metrics can be found in Table 11. The answer alone performs worse than the question alone. We hypothesize that it is due to many non-informative generic answers[6]. Perhaps for other real-world datasets consultant's answer may be superior to the user's questions. Passing only the question signal is a strong baseline. Let us check if it is possible to incorporate signals from both question and answer fields in a way that improves performance over Q, question field only baseline. The most straightforward extension, *QA concatenation*, which requires only inputting different inputs to the same model is slightly better but does not pass the statistical

[5] For multi-headed encoders, we chose the best of all possible final representations (output from any head, or concatenations of outputs from multiple heads).

[6] e.g., *Thank you for your message. Let me check some details and reply later.*

	Purchase	***Delivery***	***Retail***
Q	52.9±7.3	42.3±3.6	39.2±1.2
A	51.7±5.5	37.6±4.5	30.5±1.5
QA concat.	55.1±3.8	47.3±3.4	43.4±3.1
QA two head.	56.4±5.9	46.9±5.4	40.2±1.7
Conv	**66.4±6.6**	**49.2±6.4**	**44.2±2.2**

Table 5: Evaluation of conversational structure for novel intent discovery. We report AVG metric averaged over five seed runs with standard deviation. Models use *AlleBERT* initialization, CDAC training scheme, and various inputs, i.e., question *Q*, answer *A*, or both fields (QA) in three model variants; *QA concatenation*, *QA two heads*, and *Conv*. The best results are in bold.

significance test. The same goes for the more sophisticated *QA two heads* variant. Only our method *Conv*, a three-headed encoder is better than *Q* with statistical significance. Incorporating both question and answer signal leads to further improvements.

To sum up, after examining multiple ways to include the conversational signal, we conclude that our method *Conv* with a three-headed encoder improves the performance by 5 to 13.5pp.

6 Commercial deployment

6.1 Production pipeline overview

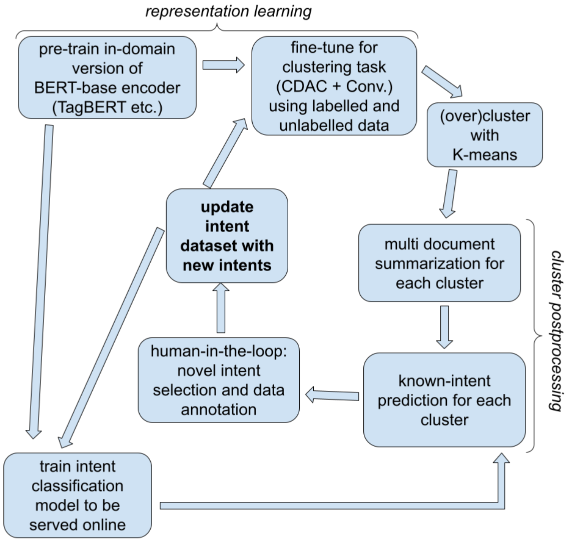
representation learning
pre-train in-domain version of BERT-base encoder (TagBERT etc.)
fine-tune for clustering task (CDAC + Conv.) using labelled and unlabelled data
(over)cluster with K-means
multi document summarization for each cluster
known-intent prediction for each cluster
cluster postprocessing
human-in-the-loop: novel intent selection and data annotation
update intent dataset with new intents
train intent classification model to be served online

Figure 2: Intent discovery pipeline deployed at *Allegro* with human-in-the-loop carrying out the novel intent selection and data annotation. Representation learning components are subject to experiments in this paper. The main outcome of the pipeline is an updated intent detection dataset, which can be used to train a better intent classification model.

The method we described and verified experimentally is a part of a larger multi-component system for continuous intent discovery deployed commercially, shown in Fig. 2. Here we briefly list the major components of our production pipeline to give the bigger picture:

1. *Representation learning.* Representation learning plays a core role in our pipeline. This component is subject to experiments in this paper and consists of two subcomponents:
   (a) *In-domain pre-training of encoders.* Encoders with *BERT-base* architecture are pre-trained on large chunks of historical data. We include additional signals such as conversational structure (i.e. question and answer) and weak label signal (Section 3.3.2 and 3.3.3). The encoders are reused for the intent classification model.
   (b) *Fine-tuning for the clustering task.* We further train in-domain encoders. If there exists annotated data, we use semi-supervised CDAC with *Conv* (Section 3.4). Otherwise, we use static embeddings.

2. *(Over)clustering with K-Means.* We cluster representations to discover intent groups in the data. The number of novel intents is required by K-Means. We overestimate this value as it is less time-consuming to manually merge clusters with the same intent.

3. *Cluster postprocessing.* Various postprocessing steps make analyzing the clusters by the human annotators more efficient:
   (a) *Multi-document summarization.* The summarization module, provides human-readable candidates for the intent name instead of cluster ids. First, we train a logistic regression classifier with bag-of-words features to predict cluster ids. Then, we identify the most informative sentence in each message using the classifier coefficients (Angelidis and Lapata, 2018). Finally, we select the five most central sentences across all messages (Zheng and Lapata, 2019).
   (b) *Known intent prediction.* We need to distinguish clusters with known intents from clusters with potentially novel intents.

Since the labeled messages are typically a small subset of the training dataset, we infill intents for the unlabeled examples with an intent classifier and present this information to human annotators.

4. *Novel intent selection and data annotation.* Human annotators manually analyze all discovered clusters and choose which novel intents to include in the taxonomy. They annotate all messages from clusters to be included in the labeled dataset to ensure the high coherence of newly discovered intents.

CX intent dataset updated with new intent is the end product of our intent discovery pipeline. Its primary purpose is to train an intent classifier to be served in real-time to CX consultants. It is a complex pipeline of its own. It has similar architecture to the representation learning model in the intent discovery pipeline and it reuses pre-trained encoders. Even though the consultant's answer and the consultant's weak label are not known at the serving time of the intent classification model, we leverage these signals to build a better intent dataset and directly train a better intent classification model.

6.2 Commercial benefits case study

Thanks to the deployed pipeline, we doubled the number of defined intents for customer support within one year. Initially, the taxonomy consisted of 100 classes manually defined by the CX consultants. The commercial deployment of the intent discovery pipeline happened at the moment when the domain experts failed to find any new intents manually. Roughly 50 new intents were discovered thanks to our intent discovery pipeline. The selected clusters were reasonably pure: over 90% (mean and median) of examples from the selected clusters were labeled as the given intent. Additional examples for the new intents were further added (active learning etc.) and at the moment, the examples from the clustering process are at least 40% of all examples for 50 automatically discovered intents. Currently, after extending our taxonomy from other sources as well, our taxonomy has roughly 180 intents.

In addition, the pipeline decreased the time required to define novel intents from weeks to days with the additional benefit of analyzing several-fold more messages. The more comprehensive taxonomy significantly impacts the total benefit from the automation process, improves user experience by providing faster responses, and saves the cost of hiring additional CX consultants.

7 Conclusions

This paper describes an intent discovery pipeline deployed on a large e-commerce platform. The access to real-life datasets allows extending the established intent discovery models to better leverage vast amounts of unlabelled data, its conversational structure, and additional signals like weak labels. In particular, we learn the following lessons:

1. Among multiple ways to handle conversational data, *Conv*, our generalization of the CDAC model to a three-headed encoder to use all available conversational data (i.e., question and answer) increases the performance of the intent discovery pipeline the most. See Section 5.3.

2. The significant gains also come from pretraining the encoder on an unlabelled in-domain dataset with conversational structure and weak labels (*TagBERT*). See Section 5.1. Therefore, we recommend a system architecture that enables weak labeling by the consultants by design.

3. Even though the consultant's answer and weak labels are not available at the serving time of the intent classification model, they can be used offline for novel intent discovery to build a better dataset and directly improve the intent classification. It happened for our comercially deployed pipeline. See Section 6.

4. Gains from incorporating additional signals (*Conv* method, *TagBERT*) are larger than gains from using state-of-the-art methods (CDAC) on datasets without additional signals. See Section 4.4. We advocate for a shift both in construction and research on intent detection datasets.

8 Limitations

We are aware of two major factors that may affect the generality of our research: shortcomings of the simulated novel intent discovery setup and the assumption that intent detection is a classification problem.

Simulated experiments. In the experimental section, we use small, entirely annotated datasets to analyze different design choices of the representation learning component. We naturally include only already discovered intents (does not mean these are all possible). Our masking procedure that follows research papers (Lin et al., 2020; Zhang et al., 2020) has three drawbacks. Firstly, when we mask most of the dataset, we effectively do few-shot learning, whereas, in reality, the amount of annotated data is much larger. The observed differences between design choices may be mitigated once more data is available. Secondly, real class imbalance may not be reflected in the experimental dataset due to the annotation procedure. Lastly, the ratio between batch size and dataset size is much smaller for real datasets since, in general, we are training with a large amount of unannotated data. It directly affects batch-based pair statistics when using a random sampler in CDAC algorithm. The chance that annotated examples will be present in the batch is low, and effectively we are almost entirely learning from pseudo-pairs during the semi-supervised stage.

Intent detection as classification. We treat the intent discovery as classification i.e. each utterance has only one intent. In reality, users may have more than one goal that transforms the problem into a multi-label scenario. Naturally, we could treat multi-label examples as yet another class, but we do not explore their influence on pipeline performance since they were in a significant minority.

Acknowledgements

We thank Bartosz Ludwiczuk, Tomi Wójtowicz, Karol Grzegorczyk for valuable discussions and contributions to the source code.

References

Stefanos Angelidis and Mirella Lapata. 2018. Summarizing opinions: Aspect extraction meets sentiment prediction and they are both weakly supervised. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3675–3686, Brussels, Belgium. Association for Computational Linguistics.

Stephen H. Bach, Daniel Rodriguez, Yintao Liu, Chong Luo, Haidong Shao, Cassandra Xia, Souvik Sen, Alexander Ratner, Braden Hancock, Houman Alborzi, Rahul Kuchhal, Christopher Ré, and Rob Malkin. 2018. Snorkel drybell: A case study in deploying weak supervision at industrial scale.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: A pretrained language model for scientific text. 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 3615–3620, Hong Kong, China. Association for Computational Linguistics.

Yoshua Bengio, Aaron Courville, and Pascal Vincent. 2013. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828.

Iñigo Casanueva, Tadas Temčinas, Daniela Gerz, Matthew Henderson, and Ivan Vulić. 2020. Efficient intent detection with dual sentence encoders. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, pages 38–45, Online. Association for Computational Linguistics.

J. Chang, L. Wang, G. Meng, S. Xiang, and C. Pan. 2017. Deep adaptive image clustering. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 5880–5888.

Sławomir Dadas. 2019. A repository of polish NLP resources. Github.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

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*.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online. Association for Computational Linguistics.

Matthew Henderson, Iñigo Casanueva, Nikola Mrkšić, Pei-Hao Su, Tsung-Hsien Wen, and Ivan Vulić. 2020a. ConveRT: Efficient and accurate conversational representations from transformers. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2161–2174, Online. Association for Computational Linguistics.

Matthew Henderson, Iñigo Casanueva, Nikola Mrkšić, Pei-Hao Su, Tsung-Hsien Wen, and Ivan Vulić. 2020b. ConveRT: Efficient and accurate conversational representations from transformers. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2161–2174, Online. Association for Computational Linguistics.

Kexin Huang, Jaan Altosaar, and Rajesh Ranganath. 2019. Clinicalbert: Modeling clinical notes and predicting hospital readmission. *arXiv:1904.05342*.

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. *CoRR*, abs/1909.02027.

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*, pages 8360–8367. AAAI Press.

Stuart P. Lloyd. 1982. Least squares quantization in pcm. *IEEE Trans. Inf. Theory*, 28:129–136.

Robert Mroczkowski, Piotr Rybak, Alina Wróblewska, and Ireneusz Gawlik. 2021. HerBERT: Efficiently pretrained transformer-based language model for Polish. In *Proceedings of the 8th Workshop on Balto-Slavic Natural Language Processing*, pages 1–10, Kiyv, Ukraine. 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 and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2020. Making monolingual sentence embeddings multilingual using knowledge distillation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4512–4525, Online. Association for Computational Linguistics.

Piotr Rybak, Robert Mroczkowski, Janusz Tracz, and Ireneusz Gawlik. 2020. KLEJ: Comprehensive benchmark for Polish language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1191–1201, Online. Association for Computational Linguistics.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. Mpnet: Masked and permuted pre-training for language understanding. In *Advances in Neural Information Processing Systems*, volume 33, pages 16857–16867. Curran Associates, Inc.

Janusz Tracz, Piotr Iwo Wójcik, Kalina Jasinska-Kobus, Riccardo Belluzzo, Robert Mroczkowski, and Ireneusz Gawlik. 2020. BERT-based similarity learning for product matching. In *Proceedings of Workshop on Natural Language Processing in E-Commerce*, pages 66–75, Barcelona, Spain. Association for Computational Linguistics.

Nikhita Vedula, Rahul Gupta, Aman Alok, Mukund Sridhar, and Shankar Ananthakrishnan. 2022. Advin: Automatically discovering novel domains and intents from user text utterances. In *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7627–7631.

Hao Wang, Yitong Wang, Zheng Zhou, Xing Ji, Dihong Gong, Jingchao Zhou, Zhifeng Li, and Wei Liu. 2018. Cosface: Large margin cosine loss for deep face recognition. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5265–5274.

Wei Wang, Bin Bi, Ming Yan, Chen Wu, Jiangnan Xia, Zuyi Bao, Liwei Peng, and Luo Si. 2020. Structbert: Incorporating language structures into pre-training for deep language understanding. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Hanlei Zhang, Hua Xu, Ting-En Lin, and Rui Lv. 2020. Discovering new intents with deep aligned clustering. *CoRR*, abs/2012.08987.

Hao Zheng and Mirella Lapata. 2019. Sentence centrality revisited for unsupervised summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6236–6247, Florence, Italy. Association for Computational Linguistics.

A Dataset details

We further describe real-world internal datasets introduced in 4.1 and compare them to public benchmark datasets. Table 6 exemplifies the domain diversity of the datasets: it contains three sample intent names per dataset.

We visualize the datasets. We use publicly available pre-trained models to enable simple visual comparisons between our real-world internal datasets and any other datasets. Sentence-BERT produces English sentence embeddings by fine-tuning on semantic textual similarity STS pairs (Reimers and Gurevych, 2019). We use a variation of Sentence-BERT trained from MPNet (Song et al., 2020). Polish version has been obtained following knowledge distillation procedure (Reimers and Gurevych, 2020; Dadas, 2019). [7] We compute sentence embeddings for the question field or if the answer field is present, for question-answer concatenation. For each example, we compute a partial Silhouette score (using ground truth intents as cluster labels) and average it per intent. Silhouette score, designed originally for evaluating the clustering quality, takes into account the mean intra-cluster distance and the mean nearest-cluster distance for each example. We plot 2D t-SNE mappings of the embeddings, Silhouette score per intent [8], and intent sizes in Figures 3 and 4 to visualize the datasets and the initial difficulty of the clustering task on general domain pre-trained models.

B Training schemes

B.1 Deep Adaptive Clustering (DAC)

It was introduced in (Chang et al., 2017) for the Computer Vision domain but is easily extended to text. Originally, output representation was interpreted as a probability distribution over unique classes, i.e., they used L_2 normalized features with positive elements. We relaxed this condition and trained real-valued representation for any clustering algorithm. The representation size doesn't have to match a unique number of classes in the dataset (unknown in real scenarios). For a pair of examples i, j the loss function $\mathcal{L}_{ij}$ is

$$\mathcal{L}_{ij} = -R_{ij} \log S_{ij} - (1 - R_{ij}) \log(1 - S_{ij}), \quad (2)$$

Dataset	Three sample intent labels
Banking77	1. Cash withdrawal charge 2. Getting spare card 3. Request refund
CLINC150	1. Transactions 2. Next song 3. International fees
Purchase	1. I have a technical problem. 2. When will my Smart! be active? 3. How to withdraw from the auction?
Delivery	1. I didn't pick up my parcel and I'm asking for a refund. 2. How to withdraw from the contract? 3. I want to use Buyers Protection Program.
Retail	1. When will the sale of the offer start? 2. I have a problem with the customer service for my purchase. 3. Is the product prepackaged?

Table 6: Domain diversity of labeled datasets used for novel intent discovery experiments. Three sample intent names per datasetare given.

Dataset	***Banking77***	***CLINC150***	***Purchase***	***Delivery***	***Retail***
Representation size	256	256	32	32	64
Batch size	128	128	16	32	16
# intents	77	150	22	23	105

Table 7: Optimal representation size and batch size vs. a number of annotated intents in the datasets.

[7] Package sentence-transformers, available at https://sbert.net, is used with models all-mpnet-base-v2 or sdadas/st-polish-paraphrase-from-mpnet for English and Polish respectively.

[8] https://scikit-learn.org/

where ($R_{ij} = 1$) for positive pairs and ($R_{ij} = 0$) for negative pairs and S_{ij} is cosine similarity of representations. The pseudo-label matrix R is defined in an online fashion for every pair of examples in a batch using current model predictions i.e.

$$R_{ij} = \begin{cases} 1, & \text{if } S_{ij} \geq u(\lambda), \\ 0, & \text{if } S_{ij} < l(\lambda), \\ \text{None}, & \text{otherwise}, \end{cases} \tag{3}$$

where $u(\lambda)$ and $l(\lambda)$ are upper and lower thresholds. Pairs between the thresholds do not take part in the training. This is compensated by adding penalty term $u(\lambda) - l(\lambda)$ to the final loss. The thresholds are updated every epoch according to the formula

$$u(\lambda) = 0.95 - \lambda,$$
$$l(\lambda) = 0.455 + 0.1 \cdot \lambda,$$

where update rule for λ every epoch is $\lambda = \lambda + 1.1 \cdot 0.009$ (Chang et al., 2017). We start with $\lambda = 0$. The training ends when $u(\lambda) = l(\lambda)$. The training resembles curriculum learning: we start with confident examples with very large or low cosine similarity and then introduce more uncertainty. The penalty term also reflects our confidence since it controls the strength of gradient updates.

B.2 Constrained DAC (CDAC)

This extension of DAC to a semi-supervised scenario was introduced in (Lin et al., 2020). In unsupervised case, we only use contrastive objective with pseudo-labels. Once we have annotated examples, we define true positive and negative pairs with labels. The label matrix R has now pseudo-label part (3) and exact part

$$R_{ij} = \begin{cases} 1, & \text{if } y_i = y_j, \\ 0, & \text{if } y_i \neq y_j, \end{cases} \tag{4}$$

where y_i denotes encoded label for i-th example. Since our batch now includes annotated and unannotated examples, we need to redefine pseudo-labels. We consider three cases. Firstly, pseudo-labels can be defined only among unannotated examples. Secondly, we can allow pseudo-labels between pairs of annotated and unannotated examples. Lastly, we can define pseudo-labels for all possible pairs, including a scenario where pseudo-labels are defined among annotated pairs. We chose the second scenario.

Additional modification is alternating training. Even epochs use only annotated data and no threshold penalty. Odd epochs use the whole dataset and pseudo-label matrix as well as exact. The loss in the supervised phase is additionally scaled by the $\delta \geq 1$ hyperparameter to control the weight put on annotated data.

C Metrics[9].

We choose metrics for our experiments. Three clustering metrics measure the separation of novel intents from each other:

- **Accuracy (ACC)** measures clusters purity. Cluster and ground-truth labels are matched with the Hungarian algorithm.
- **Normalized Mutual Information (NMI)** specifies the amount of uncertainty about class labels given cluster labels.
- **Adjusted Rand Index (ARI)** checks for all sample pairs whether their assigned and ground truth labels are the same.

ACC, NMI, and ARI are calculated only on examples with a novel intent as a ground truth label.

The separation of the novel from the known intents is measured by:

- **Binary F1-score**. It is a macro F1-score with a majority vote on the cluster label calculated on the whole dataset where all known intents are one class and all novel intents are the second class.

Last but not least, there is a metric that measures both the separation between novel intents and the separation of the novel from the known:

- **Macro F1-score** with majority vote on cluster label. It turns the clustering quality problem into multi-label classification.

The macro average is calculated only for novel intents. Examples with any ground truth label may be included[10].

All metrics increase with clustering quality up to 100%. We use five random seeds, which govern intent masking and weight initialization. In

[9] We publish the code for our metrics: https://github.com/allegro/ml/tree/main/publications/intent-discovery-metrics/

[10] See: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html

the main part of the paper, we report **AVG** i.e., the average of five metrics listed above (which are correlated variables) overall seeds. AVG is the primary metric used for model selection. Whenever in doubt, we confirm that the difference between AVG metrics is statistically significant with correlated T-Test with a p-value=5% threshold. Additionally, to facilitate comparison with other research, for all experiments, the five metrics are listed separately in Appendix.

D Initial fine-tuning

We start our experiments with fine-tuning representation size, batch size, and learning rate hyperparameters for the CDAC training scheme[11]. For every dataset, we optimize the hyperparameters in two steps: selecting optimal representation size via grid search over the representation sizes {16, 32, 64, 128, 256} and learning rates {1e-05, 5e-05, 1e-04} and then selecting the optimal learning rate and batch size via grid search over batch sizes {16, 32, 64, 128, 256, 512} and the same learning rates as step 1. Tab. 7 shows the relation of the selected hyperparameters to the number of intents. The selected hyperparameters are later fixed in the experiments. Additionally, to improve training stability, we perform an additional learning rate search again within values {1e-05, 5e-05, 1e-04} for every setup which uses *Conv* method separately.

E Pre-trained encoders (details)

To leverage large amounts of historical data, we compare four self-supervised encoders, and one supervised trained on conversational data. The training procedure for each encoder is described in detail below for reproducibility. The encoders are used for experiments in Sec. 4.4.

HerBERT State-of-the-art *BERT-base* language model for Polish (Mroczkowski et al., 2021) trained with Masked Language Model (MLM) objective.

AlleBERT The model is a result of further fine-tuning *HerBERT* on internal unsupervised conversational data. The single training example contains a conversation thread clipped to 512 tokens. We always clip threads to a random subsequence of whole consecutive utterances to persist in a conversational context. *AlleBERT* is trained with the MLM objective for 100k steps with the linearly decaying learning rate schedule (peak value 1e-05) and the batch size of 224. The training on four NVIDIA A100 GPUs lasted 2 days.

AlleConveRT The model is a result of further fine-tuning of the *AlleBERT* on the same data but with the mixture of two objectives, MLM loss with the ratio of 0.2 and Conversational Contrastive Loss (CCL). Following ConveRT (Henderson et al., 2020b) we leverage the structure of the conversations with alternately exchanged utterances in a metric learning setup. Positive examples are consecutive messages from a single conversation, and negatives come from answers within the training batch. To reduce the overfitting to specific utterances, we use label smoothing with the value of 0.2 (same as (Henderson et al., 2020b)). To utilize conversational data structure, we add two projection heads on top of the *AlleBERT* encoder, one for the question and answer representations[12]. *AlleConveRT* is trained for the 280k steps with the peak learning rate 1e-05 and the batch size of 448. The training on four NVIDIA A100 GPUs lasted 4 days.

TagBERT The model is trained in two-stage fine-tuning of the first version of *HerBERT* (Rybak et al., 2020). In the first stage, we fine-tune the model on internal unsupervised conversational data. We use MLM objective and Message Threads Structural Objective (MTSO). MTSO is Sentence Structural Objective (Wang et al., 2020) tailored to the conversation domain. During training, we swap messages with respect to threads instead of swapping sentences with respect to documents. *TagBERT* is trained for 100k steps with a batch size of 640 and a peak learning rate 8e-05.

In the second stage, we fine-tune the model on the multi-label classification task. The model predicts several of the 512 classes for each thread. The noisy and highly imbalanced labels come from tags that CX consultants add to the conversation threads, roughly identifying the problem solved. The training dataset contains 2.5M messages. *TagBERT* is trained for 38k steps with a peak learning rate of 1.6e-04 and a batch size of 512. The training on sixteen NVIDIA P100 GPUs lasted 8 hours.

F Results (details)

[11] We focus on CDAC encouraged by initial good results for CDAC and high cost of fine-tuning each training scheme separately.

[12] Answers in our data come from two sources: CX consultants and sellers.

Dataset	*Purchase*					*Delivery*					*Retail*				
Method	macro F1	ACC	NMI	ARI	binary F1	macro F1	ACC	NMI	ARI	binary F1	macro F1	ACC	NMI	ARI	binary F1
Static	23	39	45	17	62	19	28	37	8	64	10	20	47	5	62
CDAC	33	50	61	30	77	27	39	50	16	72	15	32	57	10	67
Our	**75**	**83**	**88**	**78**	**92**	**49**	**64**	**72**	**56**	**81**	**19**	**42**	**65**	**31**	**70**

Table 8: Static baseline and CDAC representations compared with our framework on novel intent discovery task for real-world data. Our framework combines *TagBERT*pre-trained encoder, CDAC training scheme, and *Conv* method for using the conversation structure. Individual metrics averaged over five seeds.

Dataset	*Purchase*					*Delivery*					*Retail*				
Initialization	macro F1	ACC	NMI	ARI	binary F1	macro F1	ACC	NMI	ARI	binary F1	macro F1	ACC	NMI	ARI	binary F1
HerBERT	53	66	73	53	84	27	44	49	25	78	18	33	57	10	68
AlleBERT	54	67	74	52	86	33	46	58	32	77	17	42	65	27	70
AlleConveRT	60	74	83	67	83	46	57	64	41	**81**	**20**	**48**	**71**	**36**	**72**
TagBERT	**75**	**83**	**88**	**78**	**92**	**49**	**64**	**72**	**56**	**81**	19	42	65	31	70

Table 9: Impact of initialization for novel intent discovery. *Conv* conversation structure-aware encoder was trained with the CDAC scheme from different initialization. Individual metrics averaged over five seeds.

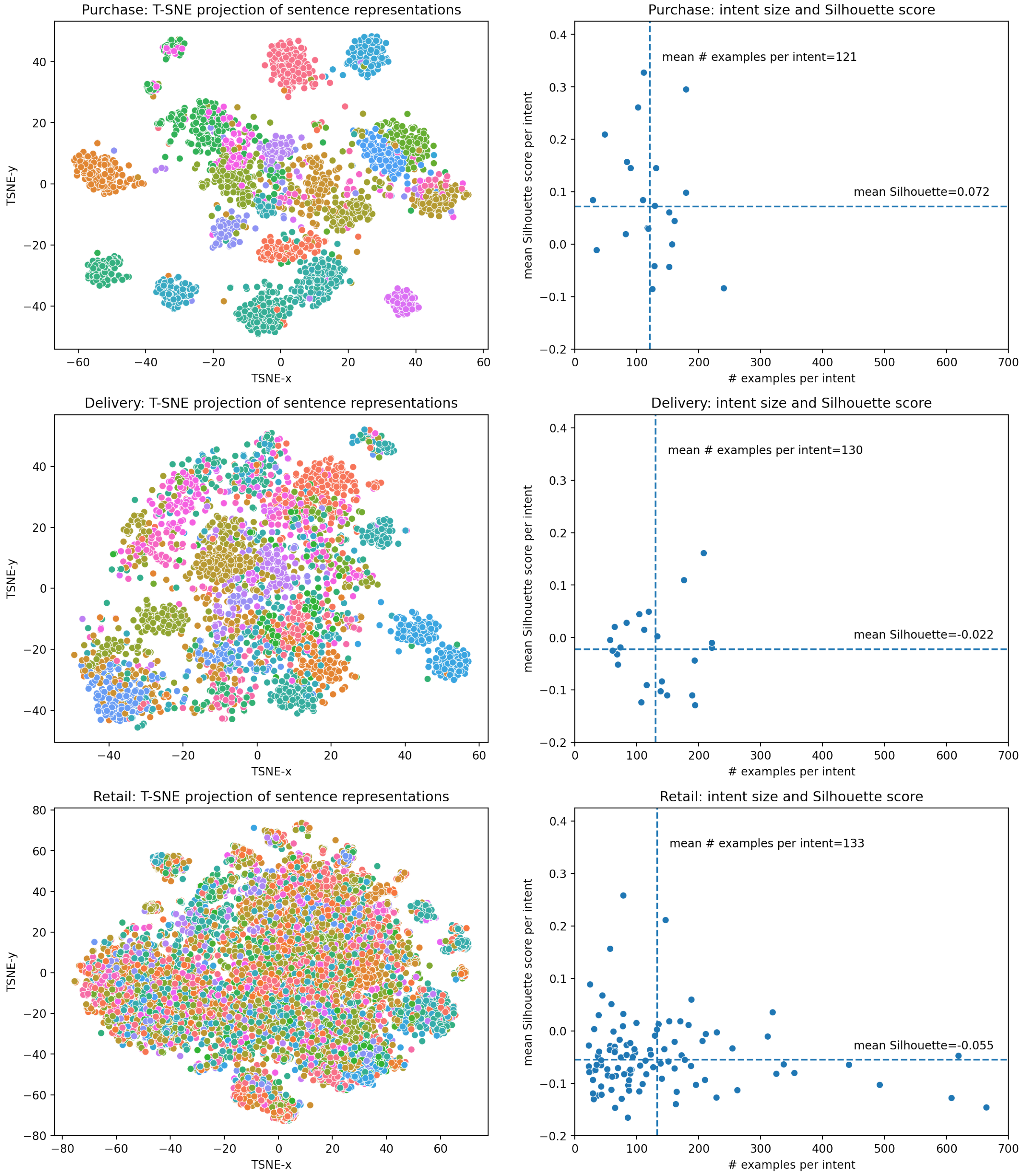

Figure 3: Internal dataset visualization. On the left we visualize t-SNE mapping of sentence representations to 2 dimensions. Different colors indicate different intent labels, each point corresponds to a single example in the dataset. On the right there is a scatter plot of intent sizes and Silhouette score per intent. Each point corresponds to one intent in the dataset. Silhouette score values are in the range from -1 to 1. 1 indicates perfect clustering, and 0 indicates overlapping clusters. The visualizations show the initial difficulty of the clustering task on general domain pre-trained models.

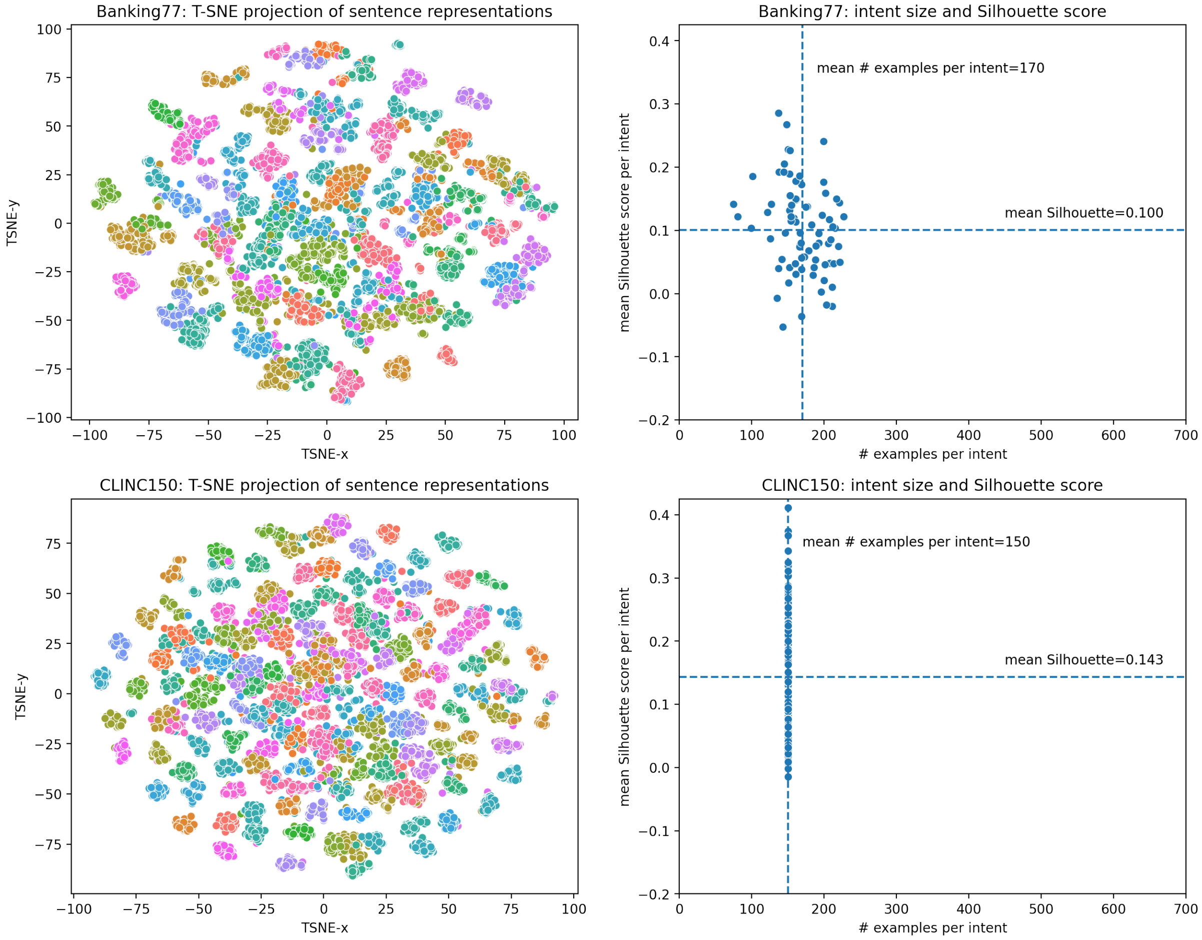

Figure 4: Public dataset visualization. On the left we visualize t-SNE mapping of sentence representations to 2 dimensions. Different colors indicate different intent labels, each point corresponds to a single example in the dataset. On the right there is a scatter plot of intent sizes and Silhouette score per intent. Each point corresponds to one intent in the dataset. Silhouette score values are in the range from -1 to 1. 1 indicates perfect clustering, and 0 indicates overlapping clusters. The visualizations show the initial difficulty of the clustering task on general domain pre-trained models.

	Dataset	***Banking77***	***CLINC150***	***Purchase***	***Delivery***	***Retail***
Static	macro F1	30	44	21	17	11
	ACC	33	49	36	30	22
	NMI	55	75	41	36	48
	ARI	23	36	14	9	6
	binary F1	68	75	65	63	62
DAC	macro F1	42	55	13	12	10
	ACC	45	58	22	20	17
	NMI	64	81	30	28	46
	ARI	35	49	0	1	3
	binary F1	73	80	55	59	61
Supervised	macro F1	55	64	22	19	11
	ACC	60	68	36	32	26
	NMI	76	86	46	38	45
	ARI	51	61	12	9	4
	binary F1	83	87	75	70	64
CDAC	macro F1	51	58	34	30	18
	ACC	54	66	54	42	35
	NMI	74	86	67	51	61
	ARI	47	59	36	17	14
	binary F1	82	83	74	72	68

Table 10: Impact of training schemes for novel intent discovery. Models use *BERT-base* (English datasets) or *AlleBERT* (Polish datasets) encoder and question input only. Individual metrics averaged over five seeds.

	Dataset	***Purchase***	***Delivery***	***Retail***
Q	macro F1	30	34	18
	ACC	54	42	35
	NMI	67	51	61
	ARI	36	17	14
	binary F1	74	72	68
A	macro F1	27	23	12
	ACC	55	35	24
	NMI	64	44	49
	ARI	42	15	6
	binary F1	70	71	61
QA concat.	macro F1	27	31	17
	ACC	59	45	41
	NMI	71	55	64
	ARI	48	26	26
	binary F1	71	80	69
QA two head.	macro F1	38	32	19
	ACC	56	44	36
	NMI	68	54	62
	ARI	44	28	16
	binary F1	75	76	69
Conv	macro F1	54	33	17
	ACC	67	46	42
	NMI	74	58	65
	ARI	52	32	27
	binary F1	86	77	70

Table 11: Impact of conversational structure for novel intent discovery. Models use *AlleBERT* initialization, CDAC training scheme, and various inputs, i.e., question *Q*, answer *A*, or both fields (QA) in three model variants; *QA concatenation*, *QA two heads*, and *Conv*. Individual metrics averaged over five seeds