DialogVCS: Robust Natural Language Understanding in Dialogue System Upgrade

Zefan Cai[1,2*], **Xin Zheng**[3,5*], **Tianyu Liu**[4*†], **Xu Wang**[4], **Haoran Meng**[1,2], **Jiaqi Han**[4], **Gang Yuan**[4], **Binghuai Lin**[4], **Yunbo Cao**[4] **and Baobao Chang**[1†]
[1]National Key Laboratory for Multimedia Information Processing, Peking University
[2]School of Software and Microelectronics, Peking University, China
[3]Institute of Software, Chinese Academy of Sciences, China [4]Tencent Cloud AI
[5]University of Chinese Academy of Sciences, China
zefncai@gmail.com; zhengxin2020@iscas.ac.cn; rogertyliu@tencent.com;

Abstract

In the constant updates of the product dialogue systems, we need to retrain the natural language understanding (NLU) model as new data from the real users would be merged into the existing data accumulated in the last updates. Within the newly added data, new intents would emerge and might have semantic entanglement with the existing intents, e.g. new intents that are semantically too specific or generic are actually a subset or superset of some existing intents in the semantic space, thus impairing the robustness of the NLU model. As the first attempt to solve this problem, we setup a new benchmark consisting of 4 Dialogue Version Control dataSets (DialogVCS). We formulate the intent detection with imperfect data in the system update as a multi-label classification task with positive but unlabeled intents, which asks the models to recognize all the proper intents, including the ones with semantic entanglement, in the inference. We also propose comprehensive baseline models and conduct in-depth analyses for the benchmark, showing that the semantically entangled intents can be effectively recognized with an automatic workflow. Our code and dataset are available at https://github.com/pkunlp-icler/DialogVCS.

1 Introduction

With the rapid growth of the business market for the task-oriented chatbots, the service providers would constantly upgrade their dialogue systems in order to be adaptable to the changing user requirements. Within the system update, the workflow of updating the existing natural language understanding (NLU) model is to collect a new training corpus by accumulating emerging data and merging them into the existing training data in the last iteration, followed by retraining with the updated corpus. Throughout

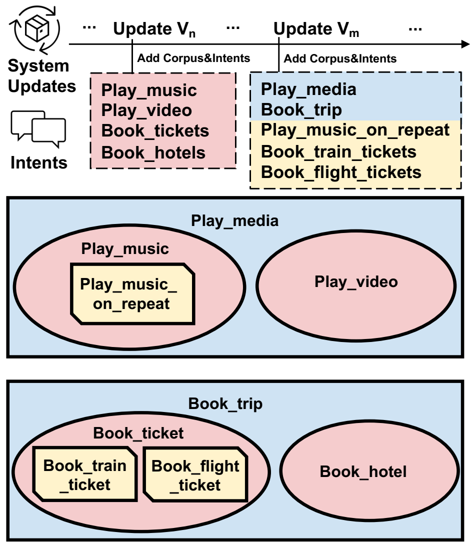


Figure 1: A motivating example for DialogVCS. In the m-th system update, the intents colored in pink are the existing labels while the intents colored in blue and yellow are the emerging ones. The emerging intents might be overlapped, e.g. being excessively specific (yellow) or generic (blue), with the existing ones in the semantic space.

the model update, new intents would emerge as more and more real-world user queries arrive.

The prior research on NLU focused on the utterance understanding with a well-defined intent[1] ontology, with the assumption that the entire intents are semantically separable and organized in the proper granularity[2]. However, the emerging intents from NLU model update might be incompatible with the existing intent ontology and thus violate the assumption regarding the properties of being semantically non-overlapping and maintain-

*Equal contribution.
†Corresponding authors.

[1]"intent" refers to the underlying goal or purpose of a user's request or query in a dialogue. This is a commonly used concept in task-oriented dialogue datasets including MultiWOZ, CrossWOZ, SNIPS, and ATIS.

[2]A well-designed NLU ontology should adequately split the entire user semantic space into the non-overlapping intents with appropriate granularity, i.e. each intent should not be excessively generic or specific in terms of semantics.

Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pages 5431–5452
June 16-21, 2024

ing well-designed granularity, e.g. the emerging intents '*play_music_on_repeat*' and '*play_media*' are semantically too specific or generic with respect to the existing intent '*play_music*'. We categorize the semantic overlapping problem between the emerging and the existing intents among the system upgrade into two categories, namely *version conflict* and *merge friction*, in which the version conflicts signify the emerging intents are too semantically specific and thus covered by the existing intents while the merge frictions are just the opposite. We argue that the semantic overlapping problem between emerging and existing intents occurs frequently in the dialogue system updates as the careful human modification for each emerging intent would be prohibitive due to the limited labor budgets and the imminent product delivery deadlines. The defective data would even propagate and accumulate through consecutive upgrades, and thus largely impair the robustness of the NLU models.

We formulate the problem as a multi-label classification task with positive but unlabeled intents[3]. As the first step towards solving this problem, we setup a benchmark consisting of 4 dialogue version control datasets (DialogVCS) to simulate the semantically overlapped intents. We employ a fully automatic workflow to create the ATIS-VCS, SNIPS-VCS, MultiWOZ-VCS, CrossWOZ-VCS datasets from 4 canonical NLU datasets, including ATIS (Hemphill et al., 1990), SNIPS (Coucke et al., 2018), MultiWOZ (Zang et al., 2020) and CrossWOZ (Zhu et al., 2020), by splitting the original intents according to the pivot entities or intentions. By leveraging existing high-quality datasets, it provides a distinct advantage in terms of quality assurance. On the other hand, manual annotation on real scenario data could be challenging to maintain consistent quality. The most critical challenge of DialogVCS is the discrepancy between training and inference, i.e. for each training instance, only one intent is provided as the target label[4], while in the testing phase, the models are expected to output all the ground-truth labels. Thus we setup multiple baselines concerning with positive but unlabeled (PU) learning for the proposed benchmark and find that the baseline models are capable of detecting semantically overlapped intents automatically.

We summarize our contributions below: **1)** We model the version conflicts and merge frictions of NLU models in the industrial dialogue system update as a multi-label classification task with positive but unlabeled intents, making it accessible to the research community. **2)** We propose 4 dialogue version control datasets by simulating the semantic overlapping problem on the ATIS, SNIPS, MultiWOZ, and CrossWOZ datasets. **3)** We setup various baselines for the proposed benchmark and show that the semantically overlapping intents can be effectively detected with an automatic workflow.

2 Task Overview

Background on system updates In the product conversational AI platforms with NLU functionalities (Ram et al., 2018; Hoy, 2018; Meng et al., 2022; Zheng et al., 2022; Liang et al., 2022) based on cloud computing, service providers would offer accessible ways, i.e. easy-to-use user interfaces, low-code application programming interfaces (APIs), for users (programmers or operators) to customize their task-oriented dialogue systems. As one of the core components in the task-oriented chatbots, the dialogue platform would provide common query understanding skills, such as weather and traffic inquiry, music and video playing, and food delivery, as the default native skills to ramp up the initial product delivery. The native skills would be updated periodically as more and more customer data comes from real-world users. After deploying the very first version of their chatbots with selected native skills, the users would constantly add new functionalities or modify existing ones following the continuous integration/delivery (CI/CD) routines (Duvall et al., 2007; Shahin et al., 2017). Except for the native skills, users would also customize user skills by adding their own training corpus[5] to the platform. In a nutshell, the natural language understanding (NLU) module of the task-oriented chatbots might be updated due to *the upgrades of the native skills* or *the adaptations to the customized user skills*.

Formulations To better signify the two aforementioned challenges, suppose at first we have two intents i_1 and i_2, the version conflict would occur when the new intents i_1^{v1}, i_1^{v2} emerges where the

[3]We focus on the intent detection rather than slot filling, as empirically we observe over 95% of bad cases associated with NLU model update are at the intent level in a commercial dialogue platform with a considerable market share.

[4]Note that we assume all the labeled intents in the training instances are factually correct, i.e. no dataset noise (false annotations) occurs.

[5]Most AI platforms would help the users reduce the labor cost of data annotation with automatic data augmentation, few-shot learning capability, etc.

superscripts $v1$ and $v2$ imply that i_1^{v1} and i_1^{v2} are different labels with respect to i_1 but semantically identical; the merge friction would occur as the new intent $i_1\&i_2$ appears where the ampersand emphasizes the new intent is different but semantically affiliated to i_1 and i_2. Note that i_1, i_1^{v1} and $i_1\&i_2$ are just the notations of the given intents rather than the real intent names, which means we can not know the relations among these intents a priori.

3 Dataset Collection

3.1 Raw Data Collection

We collect data from two single-turn dialog datasets ATIS (Hemphill et al., 1990) and SNIPS (Coucke et al., 2018), and two multi-turn dialog datasets MultiWOZ 2.1 (Eric et al., 2019) and CrossWOZ (Zhu et al., 2020). ATIS is a classic dataset on the flight inquiry, while SNIPS was collected from the real-world voice assistant and covers broader domains. MultiWoZ is a task-oriented dataset with seven domains: taxi, restaurant, hotel, attraction, train, police, and hospital, but the last two domains are not in the validation or test set, so we drop them following the prior work (Lee et al., 2019; Kim et al., 2020; Moradshahi et al., 2021). CrossWOZ is a Chinese task-oriented dataset with the same domain setting as MultiWOZ's validation/test set: taxi, restaurant, hotel, attraction, and train. For these WOZ datasets, we treat each utterance as an instance, rather than the whole dialog. The statistics of the datasets are shown in Table 1.

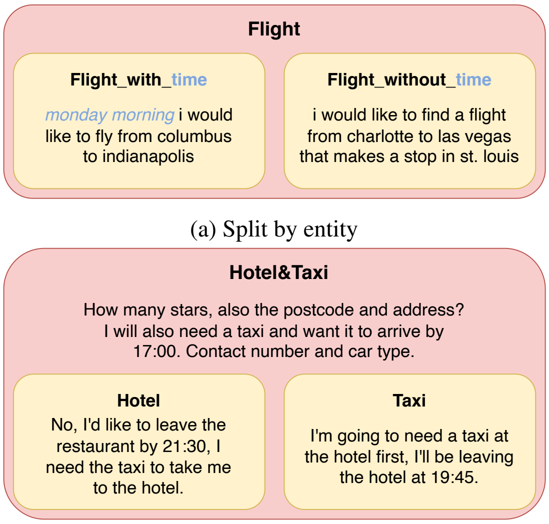
3.2 Version Conflict

We simulate the version conflict by sampling. Given an instance Ins with the original label $l = i_1$ and versions set $V = \{v_1, v_2, ..., v_k\}$, we uniformly sample the version v from V, and reset the label of the instance as $l' = i_1^v$. In the real-world applications, a specific intent might have multiple versions, but to control the difficulty of the dataset, here we assume the maximum number of versions is 2, i.e. $k = 2$. At testing time, the model shall predict both versions of the label i_1^{v1} and i_1^{v2}.

3.3 Merge Friction

For merge friction, the label-splitting strategies on composite intents are different regarding single-intent and multi-intent datasets.

Split Single Intent For ATIS and SNIPS, where each instance is annotated with one single intent i and several related entities E or slots, we could



(a) Split by entity

(b) Split by intention

Figure 2: The examples of intent splitting while simulating the merge friction issue. (2a) For single-intent datasets, i.e. ATIS and SNIPS, we split the intent "$Fight$" into two sub-intents "$Flight\_with\_time$" and "$Flight\_without\_time$" by the pivot entity "$time$". (2b) For multi-intent datasets, i.e. MultiWOZ and CrossWOZ, we split the composite intent "$Hotel\&Taxi$" into two atomic intents "$Hotel$" and "$Taxi$".

split the single intent i into two separate sub-intent $i_1 = i_{with\_entity\_j}$ and $i_2 = i_{without\_entity\_j}$, the classification rule of which is whether this instance contains the $entity\_j$ or not, and the original intent i becomes compositional $i_1\&i_2$. For example, as shown in figure 2a, given an utterance "i would like to find a flight from charlotte to las vegas that makes a stop in st. louis" with the intent $Flight$, since it does not contain any time entity, the sub-intent shall be $Flight\_without\_time$; on the other hand, given an utterance "monday morning i would like to fly from columbus to indianapolis" with the same intent, since it contains time entity "monday morning", the sub-intent shall be $Flight\_with\_time$. For training data, we randomly relabel the instance by sub-intent i_1, i_2 or full-intent $i_1\&i_2$. While testing, the model shall predict both the fine-grain and coarse-grain labels. The split intents are shown in Table A3.

Split Multi Intent Unlike the previous situation, for MultiWOZ and CrossWOZ each instance might contain multiple intents, which makes splitting intent easier. We reconsider the deduplicated multi-intents as a new compositional label $i_1\&i_2$, and naturally its atomic labels are i_1 and i_2. An example is shown in Figure 2b, each of the three instances could be labeled as any of the three labels, whether compositional label $Hotel\&Taxi$, or atomic labels $Hotel$ and $Taxi$. For training data, we randomly relabel the instance by one of the atomic intents i_1 and i_2, or the compositional intent $i_1\&i_2$. While testing, the model should predict all the ground-truth labels. The split intent is at Table A4.

4 Methods

We highlight the technical challenges of DialogVCS: **1)** The discrepancy between training and testing due to the positive but unlabeled (PU) setting; **2)** The risk of pivoting the model training with false negative labels; **3)** The extreme 0-1 class imbalance of multi-label classification. We propose multiple baselines towards these challenges.

4.1 Basic Classifier

Considering the proposed task as a multi-label classification task, we apply a linear classification at the head of the output of pre-trained language model (PLM). we use a PLM to get the representations for every token x in sentence: $[\mathbf{h}_1, \mathbf{h}_2, ..., \mathbf{h}_n] = PLM([\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n])$ where h_i is the representation for token x_i. Then, we use linear transformation and $Sigmoid$ activation function at the output representation of $[CLS]$ to get output distribution for intents: $\mathbf{y} = Sigmoid(W\mathbf{h}_1)$, where W is trainable parameter. In practice, we use threshold 0.5 for the output of $Sigmoid$ to determine the final binary output for each intent.

4.2 Method against False Negative Labels

In order to alleviate the negative effect of false negative labels, which introduce noise in training, and make the model perform poorly, we propose Negative Sample method to reduce the negative effects of the inaccurate negative samples. For each sample s in training set D_{train}, instead of directly using the labels given by dataset, we construct new labels by using the positive label and randomly sample $\theta * |L|$ negative samples, where $theta$ is a proportion and $|L|$ is the number of labels of the dataset. We use the model output as the labels other than the positive label and the sampled negative labels, meaning that we do not optimize all labels other than positive and negative labels. And then we use BCE Loss (Creswell et al., 2017) for optimization.

4.3 Method for Imbalanced Binary Classification

If we consider the proposed task as intent binary classification, the distribution of positive and negative sample for each class is extremely imbalanced. Targeting at the unbalance of positive and negative sample for each intent, we propose a method based on Focal Loss with label smoothing, which puts more emphasis on positive samples. Specifically, we add a label something on the original target l:

$$l^{LS} = l(1-\beta) + \frac{\beta}{|L|} \tag{1}$$

where $|L|$ denotes the number of intent classes. β is the label smoothing parameter. $\beta/\mathbf{K}$ is the soft label, which represents the number of intent labels. l is a vector where the positive labels equal 1 and the negative labels equal to 0 and p^{LS} is the modified targets, which represents a list of ground truth labels.

We introduce Focal Loss (Lin et al., 2017) to alleviate the above problems. For notational convenience, we define $\mathbf{p}_t$ as below:

$$\mathbf{p}_t = \begin{cases} p & \text{if } y = 1 \\ 1-p & \text{otherwise,} \end{cases} \tag{2}$$

To address class imbalance, we introduce a weighting factor $\alpha_t \in [0, 1]$ for class 1 and $1-\alpha$ for class -1. As the extreme class imbalance encountered during the training of classifier overwhelms the cross entropy loss and major negative samples, the easily classified negative samples comprise the majority of the loss and dominate the gradient. As α balances the importance of positive and negative samples, we add another factor $(1-\mathbf{p}_t)^\gamma$ to differentiate between easy and hard samples and focus training on hard negatives:

$$FL(\mathbf{p}_t) = -\alpha_t(1-\mathbf{p}_t)^\gamma \log(\mathbf{p}_t). \tag{3}$$

where α and γ are hyper parameters. Considering the proposed task as binary classification, there are 2 hyper-parameters α_{pos} and α_{neg} for α_t

4.4 Method for Imbalanced Multi-Label Label Classification

Another method that we are interested in exploring is to apply Cross Entropy Loss into multi-label classification instead of modeling the proposed task as binary classification. Cross Entropy Loss maximize the difference between the score of target class and the score of other classes:

$$L_{CE} = \log \left(1 + \sum_{i=1, i \neq t}^{n} e^{s_i - s_t}\right) \tag{4}$$

where $[\mathbf{s}_1, \cdots, \mathbf{s}_{t-1}, \mathbf{s}_{t+1}, \cdots, \mathbf{s}_n]$ is the output score of non-target classes and $\mathbf{s}_t$ is the output score of target class. As an extension to apply CE Loss at multi-label classification, we still want to maximize the difference between the score of target classes and the score of other classes, so we propose a multi-label CE Loss:

$$\begin{aligned} L_{mlCE} &= \log \left(1 + \sum_{i \in \Omega_{\text{neg}}, j \in \Omega_{\text{pos}}} e^{\mathrm{s_i} - \mathrm{s_j}}\right) \\ &= \log \left(1 + \sum_{i \in \Omega_{\text{neg}}} e^{s_i} \sum_{j \in \Omega_{\text{pos}}} e^{-s_j}\right) \end{aligned} \tag{5}$$

where Ω_{neg} denotes negative classes and Ω_{pos} denotes positive classes. The optimized goal of L_{mlCE} is to make $s_i < s_j$.

In our proposed task, the number of output classes is unfixed, so we need a threshold to determine which class to be positive. We introduce an additional threshold score $\mathbf{x}_0$ and optimize to make $\mathbf{s}_j > \mathbf{s}_0$ and $\mathbf{s}_i < \mathbf{s}_0$ into Equation 5:

$$\begin{aligned} L_{mlCE} =& \log \left(e^{s0} + \sum_{i \in \Omega_{neg}} e^{s_i}\right) \\ &+ \log \left(e^{-s_0} + \sum_{j \in \Omega_{pos}} e^{-s_j}\right) \end{aligned} \tag{6}$$

Equation 6 is the extension of Softmax and Cross Entropy to multi-label classification task. Instead of turning multi-label classification into multiple binary problem, it transforms it to a two-by-two minimization of scores of target classes with non-target classes, leading to alleviation of class imbalance. As we use threshold 0.5 for the output of $Sigmoid$ to determine the final binary output for each intent, we set s_0 to be 0.

4.5 Method of In-Context-Learning

Large Language Models (LLMs) (Sanh et al., 2021; Ouyang et al., 2022; Zhang et al., 2022) have demonstrated impressive few-shot generalization abilities. We are also interested in investigating generation-based methods and incorporating label semantics as inputs for generative models. For each dataset, we provide one data sample for each label. We also provide a task description and all the available label options and query the generative model to output one or more labels that match the input.

5 Experiments

5.1 Datasets and Evaluation Metrics

We show the dataset statistics in Table 1. To compare the baseline models, we adopt the standard precision(P), recall(R), F1-score(F1) for evaluation. The above metrics consider the task as a binary classification task for all intents, ignoring the multi-label classification nature of the task. So we present the exact match ratio (EM) metrics for further evaluation. More details are shown in Appendix A.3.

5.2 Experiment Settings

For a fair comparison, we use BERT-base-uncased (Devlin et al., 2019) as the text encoder for all methods. We introduce a naive baseline by applying a basic multi-label classifier (Section 4.1). Another baseline is to train the classifier exposure to all ground-truth labels, which indicate the upper bound of other models as all other models are trained with partially positive labels.

We implement all the experiments with Hugginface Transformers (Wolf et al., 2020). We specify the model_ids we used in the model repository in Table A2. All the hyperparameters used in our proposed methods are presented in Table A1.

5.3 Experiment Results

Main Results As shown in Table 2, due to the discrepancy between the label distribution in the training and testing, fine-tuning the classifier by the naive method of 'Basic Classifier' as Sec. 4.1 to DialogVCS with the naive BCE Loss yields low performance, especially under the metric of EM, indicating the challenges of DialogVCS. The proposed baselines significantly alleviate the negative effect of inaccurate negative labels. Among the three methods, Multi-Label Focal Loss as Sec. 4.4 generally outperforms other methods to be a robust method for partial positive labels. More detailed analysis and discussion are in Appendix A.7.

For new intents that have no semantic overlapping with the original intents, we train them directly as new samples without considering version conflicts or merge frictions. Since these new intents do

Dataset	Intent Statistics					Dataset Count		
	VC-N	VC-R(%)	MF-N	MF-R(%)	Total	Train	Valid	Test
ATIS-VCS	50	75.8	10	15.2	66	4455	496	876
SNIPS-VCS	24	77.4	6	19.4	31	13084	700	700
MultiWOZ-VCS	14	63.6	8	36.4	22	42342	4229	4238
CrossWOZ-VCS	10	58.8	7	41.2	17	55189	7325	7305

Table 1: The statistics of the proposed datasets. We list the label number of the intents which involve the version conflict (VC-N) or the merge friction (MF-N) issues, the correlated ratio of concerning training instances in the training set (VC-R and MF-R), as well as the dataset split for training, validation and testing.

Method	ATIS-VCS				SNIPS-VCS				CrossWOZ-VCS				MultiWOZ-VCS			
	P	R	F1	EM	P	R	F1	EM	P	R	F1	EM	P	R	F1	EM
Basic classfier	66.67	0.01	0.15	0.00	**99.99**	5.26	10.00	14.29	**98.06**	23.83	38.35	3.94	91.78	37.93	53.67	6.75
Neg. Sample	87.40	86.87	87.14	76.37	94.30	93.16	93.73	85.14	97.97	49.24	65.54	42.97	86.19	**87.06**	86.62	82.79
LS Focal loss	84.17	**88.81**	86.43	77.05	95.85	**95.95**	**95.90**	**92.86**	97.00	**88.37**	**92.48**	**80.34**	88.62	86.45	**87.52**	**85.85**
Multi-label CE	**91.77**	85.73	**88.65**	**79.91**	94.40	80.74	87.04	65.14	**98.06**	28.86	44.60	14.47	**94.00**	81.14	87.10	80.46
ChatGPT-ICL	49.84	52.33	51.06	0.03	82.86	0.58	0.6824	31.79	11.37	16.75	0.01	42.97	60.92	51.46	55.79	1.00
Upper Bound	**98.07**	86.80	**92.09**	**83.22**	96.73	**96.42**	**96.57**	**95.86**	96.90	**96.95**	**96.92**	**93.49**	89.33	**87.34**	**88.32**	**86.71**

Table 2: Model performance on the DialogVCS. We use BERT-base as the backbone text encoder for all the baselines. The 'Basic Classifier' and 'Upper Bound' methods signify the 'know nothing a priori' (no inductive bias of positive but unlabeled (PU) learning in the training) and 'know everything a priori' (exposure to all ground-truth labels in the training) settings, while other methods aim to recognize unlabeled intents in the regime of PU learning. For each setting except ChatGPT-ICL, we report the median scores among 5 runs using different random seeds.

not overlap semantically with the original intents, we can directly add to the training data.

We experimented with in-context learning of GPT-3.5 [6]. We provide one sample for each intent in the demonstration to form many examples (i.e., 66 intents for ATIS-VCS, 31 intents for SNIPS-VCS, 22 intents for CrossWOZ-VCS, and 17 intents for MultiWOZ-VCS). We add the requirement of completing the multi-label classification task and provide all options in the prompt, which is shown in Table B11. Then, we determine the intent of the model output by matching the options provided in the prompt with the generated text output. Following Ye et al. (2023); Qin et al. (2023), we randomly sample 100 instances in the test set for the test. The performance of GPT-3.5 on in-context learning (Kojima et al., 2022) under few-shot settings is satisfactory enough, which further demonstrates the challenging nature of the proposed benchmark.

Analysis on how to address the problem of intentional overlap in new and old data The benchmark can be seen as a unique adversarial dataset. It contains both test and training data, allowing for the analysis of model performance and trends under different levels of inconsistency control. This approach helps reveal the robustness of the model. As demonstrated in Table 5, the classifier experiences a significant drop in performance as the data becomes more inconsistent. However, a robust model should ideally not exhibit such a rapid decline in accuracy. Instead, it should generally maintain accuracy, or even approach the performance upper bound. This benchmark aims to reveal these characteristics in the tested models, contributing to the development of more robust NLU models for industrial dialogue systems. In addition, we make contributions to the method to address this problem. Our motivation for designing the method is to model the problem as a PU learning problem of multi-label classification. Next, we want the model to be able to identify semantically overlapped intents, so we apply three methods: Negative Sampling, Label-Smoothing Focal Loss, and Multi-Label Cross-Entropy.

Model Scale Up Table 3a shows the model performance on DialogVCS with different size of text encoder. We use Label-Smoothing Focal Loss method due to its high performance in Table 2. Results show that scaling up generally benefits the model performance. Transferring from BERT-Small to BERT-Base brings up to 9 points growth in the F1 score, and transferring from BERT-Base to BERT-Large brings up to 5 points growth in the

[6]https://openai.com/blog/chatgpt

F1 score. However, the performance of CrossWOZ-VCS dataset does not follow this trend, which might be caused by the insufficient training of large-size Chinese BERT models.

Size	ATIS-VCS	SNIPS-VCS	Cro-VCS	Mul-VCS
Small	77.78	90.68	**95.60**	86.26
Base	86.43	95.90	92.48	**87.52**
Large	**91.57**	**97.45**	87.66	87.34

(a) Exploration on Model Scale

Model	ATI-VCS	SNI-VCS	Cro-VCS	Mul-VCS
BERT	86.43	95.90	92.48	**87.52**
RoBERTa	91.03	**96.34**	92.41	86.62
AlBERT	84.64	88.45	84.58	85.92
DeBERTa	**91.56**	90.89	**95.93**	86.61

(b) Exploration on Model Structures

LSR	ATIS-VCS	SNIPS-VCS	Cross-VCS	Multi-VCS
0.1	77.67	**95.90**	**92.48**	86.86
0.2	**86.43**	95.05	80.89	87.02
0.4	85.13	88.58	80.59	**87.52**

(c) Exploration on Label Smoothing Rates

Table 3: The F1 scores of the the Label-Smoothing Focal Loss method with different model size (3a), different structures of the encoder (3b), and different label smoothing rates (LSR) (3c). The full tables are provided in Table A6, Table A7, and Table A8.

Model Structure We are also interested in whether the selection of text encoder is important for the task performance. Table 3b shows the model performance with different model structure for text encoder. We experiment four model structures of the text encoder, including BERT-Base, RoBERTa-Base (Liu et al., 2019), AlBERT-Base (Lan et al., 2019) and DeBERTa-Base (He et al., 2020). Results show that RoBERTa-Base and DeBERTa-Base generally outperform others.

Label Smoothing Rate for Focal Loss Our Label-Smoothing Focal Loss method consists of a dedicated label smoothing strategy. Intuitively, as the negative samples are prone to be false negative in DialogVCS, smoothing the labels in this way prevents the classifier from becoming over-confident while determining negative outputs. Table 3c shows the model performance on DialogVCS when applying Label-Smoothing Focal Loss method with different label smoothing rates (LSR). The best practise for choosing label smoothing rate depends on the number of labels of the dataset, generally speaking a dataset with larger label set requires a larger label smoothing rate. As shown in table 3c, the numbers of labels in the ATIS-VCS dataset and MultiWOZ-VCS dataset are larger than those in the SNIPS-VCS dataset and CrossWOZ dataset, thus the Label-Smoothing Focal Loss method attains better performance with a larger label smoothing rate such as 0.2 and 0.4, while the best choice of label smoothing rate for the SNIPS-VCS dataset and CrossWOZ-VCS dataset is 0.1.

NSN	ATIS-VCS	SNIPS-VCS	Cross-VCS	Multi-VCS
1	66.35	**93.73**	**65.54**	**86.62**
2	79.55	91.67	58.78	84.57
4	**87.14**	82.37	52.90	77.59
8	84.46	76.99	48.72	72.60

Table 4: The F1 scores of the Negative Sampling Method under different negative sample numbers (NSN). The full table is provided in Table A10.

Negative Sample Number There is a critical hyper-parameter for the negative sampling method — the negative sample number. As illustrated in Table 4, we try to figure out the best hyper-parameter setting in terms of the negative sample number. We observe that as the negative sample number increases, the performance decreases to a large extent for the SNIPS-VCS, CrossWOZ-VCS and MultiWOZ-VCS, with an exception that 4 negative samples work the best for the ATIS-VCS dataset.

Difficulty Control We want to explore the model performances on DialogVCS with different levels of semantic entanglement. Intuitively, we can control the difficulty level by controlling the number of conflicting labels, e.g. 'easy' and 'hard' versions of DialogVCS. The details of creating such datasets are presented in Appendix A.6. As shown in Table 5, in ATIS-VCS and SNIPS-VCS, as the number of split sub-intents decreases, the dataset becomes easier, and the performance improves. While in CrossWOZ-VCS and MultiWOZ-VCS, as the number of split atomic intents decreases, the ratio of simple intents also decreases, thus the dataset becomes harder, and the performance declines. We put more details in Table A10.

Correlation Between Labels Due to the discrepancy between training set and test set for the proposed task, the key point for model success is to capture the potential correlation between related labels, i.e., labels of i_1^{v1}, i_1^{v2}, i_2^{v1}, i_2^{v2} and $i_1\&i_2$. Figure 3 displays the co-occurrence matrix between labels based on the model output of Multi-Label Fo-

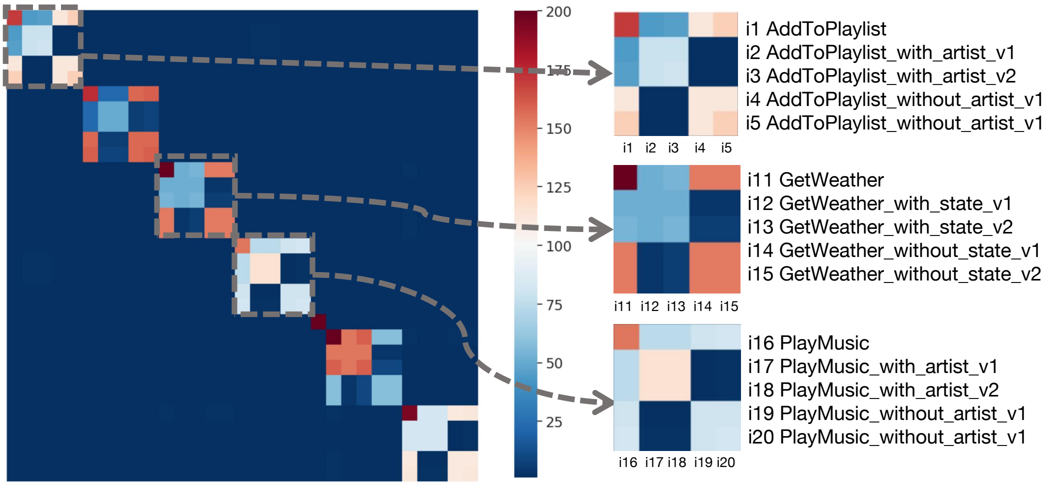


Figure 3: Display of the co-occurrence matrix between labels based on the model output of Multi-Label Focal Loss method for the test set of SNIPS-VCS. Different colors indicate different co-occurrence frequency of labels. The proposed method is able to capture the potential correlation between labels as the model output distinctly corresponds to the relationship between labels, i.e. the frequency of co-occurrence between $i_1^{v_1}$, $i_1^{v_2}$, $i_2^{v_1}$, $i_2^{v_2}$ and $i_1\&i_2$ is significantly higher than the other labels.

Difficulty	**ATIS-VCS**	**SNIPS-VCS**
Easy 1	96.17	98.50
Easy 2	96.46	95.93
Easy 4	93.15	96.72
Normal	86.43	95.90
Difficulty	**CrossWOZ-VCS**	**MultiWOZ-VCS**
Hard 1	76.07	84.96
Hard 2	80.89	85.41
Hard 4	80.59	86.29
Normal	92.48	87.52

Table 5: The F1 scores of the Label-Smoothing Focal Loss method with different levels of difficulty. We control the dataset difficulty by controlling the group numbers of label versions, i.e. k in "Easy k" or "Hard k" (Appendix A.6).

cal Loss method for the test set of SNIPS-VCS. Results for other datasets are at Figure A2, Figure A3 and Figure A4. The proposed method is able to capture the potential correlation between labels as the model output distinctly corresponds to the relationship between labels, i.e. the frequency of co-occurrence between $i_1^{v_1}$, $i_1^{v_2}$, $i_2^{v_1}$, $i_2^{v_2}$ and $i_1\&i_2$ is significantly higher than the other labels. We also visualize the model's prediction on different version labels in the test set of SNIPS-VCS in Appendix A.8.

6 Related Work

Robust NLU Recently, the topics concerning the NLP robustness and debiasing have attracted board attention (Liu et al., 2020b,a; Wang et al., 2021). To the best of our knowledge, this study is the first to investigate the non-robustness of NLU systems caused by overlapping and conflicting labels resulting from continuous system updates.

Multi-label classification Multi-label classification (Tsoumakas et al., 2006; Zhang and Zhou, 2013; Liu et al., 2021b; Wang et al., 2022) is a well-studied problem that allows each sample assigned multiple labels simultaneously. The label assignments can be incomplete in many real-world scenarios, especially with a large label set.

PU Learning The label incomplete problem is related to positive and unlabeled (PU) learning (Bekker and Davis, 2020). Many works focus on identifying reliable negative examples from the unlabeled dataset and utilize the estimated labels to improve the classification performances (Chaudhari and Shevade, 2012; Ienco et al., 2012; Basile et al., 2017; He et al., 2018).

7 Conclusion

The version conflicts and merge frictions of intents occur frequently due to the semantic overlapping between emerging and existing intents in the industrial dialogue system updates, but are unexplored in the research community. We take the first step to model the version conflict problem as a multi-label classification with positive but unlabeled intents, and propose a dialogue version control (DialogVCS) benchmark with extensive baselines. We find that the overlapping intents can be effectively detected with an automatic workflow. We leave the construction of real-scenario data for future works.

Acknowledgement

This work is supported by the National Science Foundation of China under Grant No.61936012 and 61876004.

Limitations

In this paper, we focused on the version conflicts of the intents in the NLU model update, without considering dataset noise or skewed intent distribution (extreme long-tail intents). In the real-world applications, other problems would appear in the same time as the version conflicts, thus largely impeding the robustness of NLU models. We call for more realistic, product-driven datasets for more in-depth analyses of the robustness of NLU models.

Ethics Statement

The raw data we used to create the dialogue version control datasets (DialogVCS) are all publicly available. We employ automatic data process to simulate the semantic overlapping problem as new intents emerge in the NLU model update, without introducing new user utterances. We guarantee that no user privacy or any other sensitive data were exposed, and no gender/ethnic biases, profanities would appear in the proposed DialogVCS benchmark. The model trained with the benchmark is used to identify the overlapping intents and would not generate any malicious content.

References

Teresa Basile, Nicola Di Mauro, Floriana Esposito, Stefano Ferilli, and Antonio Vergari. 2017. Density estimators for positive-unlabeled learning. In *International Workshop on New Frontiers in Mining Complex Patterns*, pages 49–64. Springer.

Jessa Bekker and Jesse Davis. 2020. Learning from positive and unlabeled data: A survey. *Machine Learning*, 109(4):719–760.

Sneha Chaudhari and Shirish Shevade. 2012. Learning from positive and unlabelled examples using maximum margin clustering. In *International Conference on Neural Information Processing*, pages 465–473. Springer.

Elijah Cole, Oisin Mac Aodha, Titouan Lorieul, Pietro Perona, Dan Morris, and Nebojsa Jojic. 2021. Multi-label learning from single positive labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 933–942.

Alice Coucke, Alaa Saade, Adrien Ball, Théodore Bluche, Alexandre Caulier, David Leroy, Clément Doumouro, Thibault Gisselbrecht, Francesco Caltagirone, Thibaut Lavril, Maël Primet, and Joseph Dureau. 2018. Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces. *CoRR*, abs/1805.10190.

Antonia Creswell, Kai Arulkumaran, and Anil A Bharath. 2017. On denoising autoencoders trained to minimise binary cross-entropy. *arXiv preprint arXiv:1708.08487*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Paul M Duvall, Steve Matyas, and Andrew Glover. 2007. *Continuous integration: improving software quality and reducing risk*. Pearson Education.

Mihail Eric, Rahul Goel, Shachi Paul, Adarsh Kumar, Abhishek Sethi, Peter Ku, Anuj Kumar Goyal, Sanchit Agarwal, Shuyang Gao, and Dilek Hakkani-Tur. 2019. Multiwoz 2.1: A consolidated multi-domain dialogue dataset with state corrections and state tracking baselines. *arXiv preprint arXiv:1907.01669*.

Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD'96, page 226–231. AAAI Press.

Anjie Fang, Simone Filice, Nut Limsopatham, and Oleg Rokhlenko. 2020. Using phoneme representations to build predictive models robust to asr errors. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 699–708.

Abbas Ghaddar, Phillippe Langlais, Mehdi Rezagholizadeh, and Ahmad Rashid. 2021. End-to-end self-debiasing framework for robust NLU training. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1923–1929, Online. Association for Computational Linguistics.

Yunchao Gong, Yangqing Jia, Alexander Toshev, Thomas Leung, and Sergey Ioffe. 2014. Deep convolutional ranking for multilabel image annotation. In *International Conference on Learning Representations*.

Zayd Hammoudeh and Daniel Lowd. 2020. Learning from positive and unlabeled data with arbitrary positive shift. *Advances in Neural Information Processing Systems*, 33:13088–13099.

Yufei Han, Guolei Sun, Yun Shen, and Xiangliang Zhang. 2018. Multi-label learning with highly incomplete data via collaborative embedding. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 1494–1503.

Fengxiang He, Tongliang Liu, Geoffrey I Webb, and Dacheng Tao. 2018. Instance-dependent pu learning by bayesian optimal relabeling. *arXiv preprint arXiv:1808.02180*.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*.

Charles T. Hemphill, John J. Godfrey, and George R. Doddington. 1990. The ATIS spoken language systems pilot corpus. In *Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania, June 24-27,1990*.

Matthew B Hoy. 2018. Alexa, siri, cortana, and more: an introduction to voice assistants. *Medical reference services quarterly*, 37(1):81–88.

Dino Ienco, Ruggero G Pensa, and Rosa Meo. 2012. From context to distance: Learning dissimilarity for categorical data clustering. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 6(1):1–25.

Atsushi Kanehira and Tatsuya Harada. 2016. Multi-label ranking from positive and unlabeled data. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5138–5146.

Ting Ke, Bing Yang, Ling Zhen, Junyan Tan, Yi Li, and Ling Jing. 2012. Building high-performance classifiers using positive and unlabeled examples for text classification. In *International symposium on neural networks*, pages 187–195. Springer.

Sungdong Kim, Sohee Yang, Gyuwan Kim, and Sang-Woo Lee. 2020. Efficient dialogue state tracking by selectively overwriting memory. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 567–582, Online. Association for Computational Linguistics.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *arXiv preprint arXiv:2205.11916*.

Xiangnan Kong, Zhaoming Wu, Li-Jia Li, Ruofei Zhang, Philip S Yu, Hang Wu, and Wei Fan. 2014. Large-scale multi-label learning with incomplete label assignments. In *Proceedings of the 2014 SIAM International Conference on Data Mining*, pages 920–928. SIAM.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations*.

Hwaran Lee, Jinsik Lee, and Tae-Yoon Kim. 2019. SUMBT: Slot-utterance matching for universal and scalable belief tracking. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5478–5483, Florence, Italy. Association for Computational Linguistics.

Hua Liang, Tianyu Liu, Peiyi Wang, Mengliang Rao, and Yunbo Cao. 2022. Smartsales: Sales script extraction and analysis from sales chatlog. *arXiv preprint arXiv:2204.08811*.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988.

Bing Liu, Yang Dai, Xiaoli Li, Wee Sun Lee, and Philip S Yu. 2003. Building text classifiers using positive and unlabeled examples. In *Third IEEE international conference on data mining*, pages 179–186. IEEE.

Jiexi Liu, Ryuichi Takanobu, Jiaxin Wen, Dazhen Wan, Hongguang Li, Weiran Nie, Cheng Li, Wei Peng, and Minlie Huang. 2021a. Robustness testing of language understanding in task-oriented dialog. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2467–2480, Online. Association for Computational Linguistics.

Jingzhou Liu, Wei-Cheng Chang, Yuexin Wu, and Yiming Yang. 2017. Deep learning for extreme multi-label text classification. In *Proceedings of the 40th international ACM SIGIR conference on research and development in information retrieval*, pages 115–124.

Tianyu Liu, Zheng Xin, Baobao Chang, and Zhifang Sui. 2020a. Hyponli: Exploring the artificial patterns of hypothesis-only bias in natural language inference. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 6852–6860.

Tianyu Liu, Zheng Xin, Xiaoan Ding, Baobao Chang, and Zhifang Sui. 2020b. An empirical study on model-agnostic debiasing strategies for robust natural language inference. In *Proceedings of the 24th Conference on Computational Natural Language Learning*, pages 596–608.

Weiwei Liu, Haobo Wang, Xiaobo Shen, and Ivor W Tsang. 2021b. The emerging trends of multi-label learning. *IEEE transactions on pattern analysis and machine intelligence*, 44(11):7955–7974.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Haoran Meng, Zheng Xin, Tianyu Liu, Zizhen Wang, He Feng, Binghuai Lin, Xuemin Zhao, Yunbo Cao, and Zhifang Sui. 2022. DialogUSR: Complex dialogue utterance splitting and reformulation for multiple intent detection. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 3214–3229, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Mehrad Moradshahi, Victoria Tsai, Giovanni Campagna, and Monica S. Lam. 2021. Contextual semantic parsing for multilingual task-oriented dialogues. *CoRR*, abs/2111.02574.

Yaroslav Nechaev, Weitong Ruan, and Imre Kiss. 2021. Towards nlu model robustness to asr errors at scale.

Mateusz Ochal, Massimiliano Patacchiola, Jose Vazquez, Amos Storkey, and Sen Wang. 2023. Few-shot learning with class imbalance. *IEEE Transactions on Artificial Intelligence*, 4(5):1348–1358.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *arXiv preprint arXiv:2203.02155*.

Chengwei Qin, Aston Zhang, Zhuosheng Zhang, Jiaao Chen, Michihiro Yasunaga, and Diyi Yang. 2023. Is chatgpt a general-purpose natural language processing task solver? *arXiv preprint arXiv:2302.06476*.

Ashwin Ram, Rohit Prasad, Chandra Khatri, Anu Venkatesh, Raefer Gabriel, Qing Liu, Jeff Nunn, Behnam Hedayatnia, Ming Cheng, Ashish Nagar, et al. 2018. Conversational ai: The science behind the alexa prize. *arXiv preprint arXiv:1801.03604*.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, et al. 2021. Multitask prompted training enables zero-shot task generalization. *arXiv preprint arXiv:2110.08207*.

Mojtaba Shahin, Muhammad Ali Babar, and Liming Zhu. 2017. Continuous integration, delivery and deployment: a systematic review on approaches, tools, challenges and practices. *IEEE Access*, 5:3909–3943.

Yu-Yin Sun, Yin Zhang, and Zhi-Hua Zhou. 2010. Multi-label learning with weak label. In *Twenty-fourth AAAI conference on artificial intelligence*.

Grigorios Tsoumakas, Ioannis Katakis, and Ioannis Vlahavas. 2006. A review of multi-label classification methods. In *Proceedings of the 2nd ADBIS workshop on data mining and knowledge discovery (ADMKD 2006)*, pages 99–109.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605.

Peiyi Wang, Runxin Xun, Tianyu Liu, Damai Dai, Baobao Chang, and Zhifang Sui. 2021. Behind the scenes: An exploration of trigger biases problem in few-shot event classification. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 1969–1978.

Zihan Wang, Peiyi Wang, Tianyu Liu, Binghuai Lin, Yunbo Cao, Zhifang Sui, and Houfeng Wang. 2022. HPT: Hierarchy-aware prompt tuning for hierarchical text classification. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3740–3751, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Yixing Xu, Chang Xu, Chao Xu, and Dacheng Tao. 2017. Multi-positive and unlabeled learning. In *IJCAI*, pages 3182–3188.

Junjie Ye, Xuanting Chen, Nuo Xu, Can Zu, Zekai Shao, Shichun Liu, Yuhan Cui, Zeyang Zhou, Chao Gong, Yang Shen, et al. 2023. A comprehensive capability analysis of gpt-3 and gpt-3.5 series models. *arXiv preprint arXiv:2303.10420*.

Xiaoxue Zang, Abhinav Rastogi, Srinivas Sunkara, Raghav Gupta, Jianguo Zhang, and Jindong Chen. 2020. MultiWOZ 2.2 : A dialogue dataset with additional annotation corrections and state tracking baselines. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, pages 109–117, Online. Association for Computational Linguistics.

Min-Ling Zhang and Kun Zhang. 2010. Multi-label learning by exploiting label dependency. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 999–1008.

Min-Ling Zhang and Zhi-Hua Zhou. 2013. A review on multi-label learning algorithms. *IEEE transactions on knowledge and data engineering*, 26(8):1819–1837.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. 2022. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.

Xinliang Frederick Zhang. 2021. Towards more robust natural language understanding. *arXiv preprint arXiv:2112.02992*.

Xin Zheng, Tianyu Liu, Haoran Meng, Xu Wang, Yufan Jiang, Mengliang Rao, Binghuai Lin, Zhifang Sui, and Yunbo Cao. 2022. Dialogqae: N-to-n question answer pair extraction from customer service chatlog. *arXiv preprint arXiv:2212.07112*.

Feng Zhu, Hongsheng Li, Wanli Ouyang, Nenghai Yu, and Xiaogang Wang. 2017a. Learning spatial regularization with image-level supervisions for multi-label image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5513–5522.

Qi Zhu, Kaili Huang, Zheng Zhang, Xiaoyan Zhu, and Minlie Huang. 2020. CrossWOZ: A large-scale Chinese cross-domain task-oriented dialogue dataset. *Transactions of the Association for Computational Linguistics*, 8:281–295.

Yue Zhu, James T Kwok, and Zhi-Hua Zhou. 2017b. Multi-label learning with global and local label correlation. *IEEE Transactions on Knowledge and Data Engineering*, 30(6):1081–1094.

A Appendix

A.1 Related Work

Robust NLU In the recent years, the topics concerning the NLP robustness and debiasing have attracted board attention. (Liu et al., 2020b,a; Wang et al., 2021) For NLU models, Nechaev et al. (2021) studied data-efficient techniques to make NLU models robust to ASR errors, including data augmentation, adversarial training, and a confidence-aware layer. Fang et al. (2020) proposed novel phonetic-aware text representations which represent ASR transcriptions at the phoneme level, aiming to capture pronunciation similarities. Besides ASR, there are other factors that affect the robustness of the NLU systems. Liu et al. (2021a) analyzed different factors affecting the robustness of NLU models including language variety, speech characteristics, and noise perturbation. Ghaddar et al. (2021) proposed a debiasing framework to slove out-of-distribution (OOD) problem in NLU. Zhang (2021) discussed three robustness problems, namely poor generalization across domains, inherently ambiguous training samples, and unreliable datasets. To the best of our knowledge, this study is the first to investigate the non-robustness of NLU systems caused by overlapping and conflicting labels resulting from continuous system updates.

Multi-label classification Multi-label classification (Tsoumakas et al., 2006; Zhang and Zhou, 2013; Liu et al., 2021b; Wang et al., 2022) is a well-studied problem that allows each sample assigned multiple labels simultaneously. The simplest solution is converting the multi-label problem into multiple independent binary classifications (one for each label) (Liu et al., 2017). But different labels are generally correlated with each other, instead of being independent. Some methods are proposed to exploit label correlations in multi-label classification (Zhang and Zhang, 2010; Sun et al., 2010; Kong et al., 2014; Zhu et al., 2017b). Additionally, there are some studies treating the task as a ranking problem, trying to rank all positive labels higher than other labels for each sample (Gong et al., 2014; Kanehira and Harada, 2016). All of these works assume that each instance in training data is fully assigned without any missing labels. However, the label assignments can be incomplete in many real-world scenarios, especially with a large label set.

PU Learning The label incomplete problem is related to positive and unlabeled (PU) learning (Bekker and Davis, 2020). PU learning aims to train a classifier from a set of positive samples and an additional set of unlabeled samples. Many works focus on identifying reliable negative examples from the unlabeled dataset and utilize the estimated labels to improve the classification performances (Chaudhari and Shevade, 2012; Ienco et al., 2012; Basile et al., 2017; He et al., 2018). Biased PU learning methods treat the unlabeled samples as negative samples with noise, and use higher penalties on misclassified positive samples to accommodate noise (Liu et al., 2003; Ke et al., 2012). Most studies on PU learning concentrate on binary classification problems which are not sufficient to cover the wide range of real-world applications. Xu et al. (2017) proposed a one-step method that directly enables a multi-class model to be trained using the given multi-class PU data. Furthermore, there are relatively few studies that explore PU learning for multi-label tasks (Sun et al., 2010; Kong et al., 2014; Kanehira and Harada, 2016; Han et al., 2018). Cole et al. (2021) addressed the hardest multi-label version in which there is only a single positive label available for each sample in training time, and the model needs to predict all proper labels at test time.

A.2 Hyper Parameters

We list the detailed hyperparameters in Table A1. All experiments are run on a NVIDIA-A40. In Table A2, we list the models used in this paper and their mapping with the hugginface model_ids. We use a NVIDIA-A40 for 80 hours to get all the reported results.

A.3 Metrics

We show the dataset statistics in Table 1. To compare the baseline models, we adopt the standard precision(P), recall(R), F1-score(F1) for evaluation. The above metrics consider the task as a binary classification task for all intents, ignoring the multi-label classification nature of the task. So we present the exact match ratio (EM) metrics for further evaluation.

All the above metrics are under the setting that a label is predicted as positive if its estimated probability is greater than 0.5 (Zhu et al., 2017a). Among these metrics, F1 and EM are the most representative metrics.

Name	ATIS-VCS	SNIPS-VCS	CrossWOZ-VCS	MultiWOZ-VCS
Learning Rate	2e-5	2e-5	2e-5	2e-5
Batch Size	512	512	512	512
Max Sequence Length	32	32	32	32
Sample Number in Sec.4.2	4	1	1	1
β in Eq.1	0.2	0.1	0.1	0.4
γ in Eq.3	4	4	4	4
α_{neg} in Eq.3	0.00001	0.00001	0.00001	0.00001
α_{pos} in Eq.3	0.99999	0.99999	0.99999	0.99999
s_0 in Eq.6	0	0	0	0

Table A1: All hyper parameters used in Table 2.

Model_name	Hugginface_ModelID
BERT-small (English)	bert-small
BERT-base (English)	bert-base-uncased
BERT-large (English)	bert-large-uncased
RoBERTa-base (English)	roberta-base
ALBERT-base (English)	albert-base-v2
DeBERTa-base (English)	deberta-base
BERT-small (Chinese)	bert-tiny
BERT-base (Chinese)	bert-base-chinese
BERT-large (Chinese)	bert-large-chinese
RoBERTa-base (Chinese)	chinese-roberta-wwm-ext
ALBERT-base (Chinese)	albert-base-chinese -cluecorpussmall
DeBERTa-base (Chinese)	deberta-base-chinese

Table A2: The model mapping between model names and hugginface model ids used in this paper.

Intent	Split Entity
flight	time
abbreviation	fare_basis_code
aircraft	loc
airfare	cost_relative
airline	airline_code
capacity	aircraft_code
city	airline_name
flight_no	airline_name
flight_time	depart
ground_service	airport_name

(a) ATIS-VSC

Intent	Split Entity
AddToPlaylist	artist
BookRestaurant	restaurant_name
GetWeather	state
PlayMusic	artist
SearchCreativeWork	object_type
SearchScreeningEvent	object_type

(b) SNIPS-VSC

Table A3: Split intent of ATIS (A3a) and SNIPS (A3b)

$$\begin{aligned} \mathrm{P} &= \frac{\sum_i N_i^c}{\sum_i N_i^p}, \quad \mathrm{F1} = \frac{2 \times \mathrm{P} \times \mathrm{R}}{\mathrm{P} + \mathrm{R}}, \\ \mathrm{R} &= \frac{\sum_i N_i^c}{\sum_i N_i^g}, \quad \mathrm{EM} = \frac{1}{m} \sum_{j=1}^{m} I\left(p_j == l_j\right) \end{aligned} \tag{7}$$

where N_i^c is the number of intents that are correctly predicted to be true for the i-th label, N_i^p is the number of intents predicted to be true for the i-th label, N_i^g is the number of ground truth intents for the i-th label, m is the number of instances in test dataset D_{test}, p_j is the model output of all intent labels for sample s_j, l_j is the ground truth intent labels for sample s_j and $I()$ is an indicator function, which will output 1 when the distribution of p_j is equivalent to l_j.

A.4 Split Intent in Proposed Datasets

For single-intent datasets ATIS and SNIPS, we split the intent into two sub-intents by critical entity, which is listed in Table A3. For multi-intent datasets MultiWOZ and CrossWOZ, we split the composite intent into several atomic intents, which is listed in Table A4.

A.5 Extended Experiment Results

We list the full experiment scores of the analyses on model scale up, model structure, label smoothing for Label-Smoothing Focal Loss, negative sample number in Table A6, A7, A8, A9, respectively.

A.6 Difficulty Control

We introduce version conflict and merge friction to every possible label, but in practice, we may not see version labels in such a high proportion. To better simulate the actual scenario and also have better control over the difficulty of the datasets, we limit the number of version labels to 1, 2, and 4. For ATIS-VCS and SNIPS-VCS, more version labels would be more difficult, since intent splitting creates sub-intents that need to check both the original intent and the critical entity. For example, checking the sub-intent "$Flight\_with\_time$" requires more computation than full-intent "$Flight$".However, for MultiWOZ-VCS and CrossWOZ-VCS, more version labels would not be more difficult, because

Composite Intent	Atomic Intent
attraction&hotel	attraction,hotel
attraction&restaurant	attraction,restaurant
attraction&train	attraction,train
hotel&restaurant	hotel,restaurant
hotel&taxi	hotel,taxi
hotel&train	hotel,train
restaurant&taxi	restaurant,taxi
restaurant&train	restaurant,train

(a) MultiWOZ-VSC

Composite Intent	Atomic Intent
General&Inform	General,Inform
General&Inform&Request	General,Inform,Request
General&Inform&Select	General,Inform,Select
General&Request	General,Request
Inform&Request	Inform,Request
Inform&Request&Select	Inform,Request,Select
Inform&Select	Inform,Select

(b) CrossWOZ-VSC

Table A4: Split intent of MultiWOZ (A4a) and Cross-WOZ (A4b)

Dataset	Difficulty	VC-N	MF-N	Total
ATIS_1	Easy 1	4	1	20
ATIS_2	Easy 2	8	2	24
ATIS_4	Easy 4	16	4	32
ATIS	Normal	50	10	66
SNIPS_1	Easy 1	4	1	11
SNIPS_2	Easy 2	8	2	15
SNIPS_4	Easy 4	16	4	23
SNIPS	Normal	24	6	31
MultiWOZ_1	Hard 1	4	1	17
MultiWOZ_2	Hard 2	6	2	18
MultiWOZ_4	Hard 4	10	4	20
MultiWOZ	Normal	14	8	22
CrossWOZ_1	Hard 1	4	1	15
CrossWOZ_2	Hard 2	6	2	17
CrossWOZ_	Hard 4	8	4	17
CrossWOZ	Normal	10	7	17

Table A5: The number of version conflict labels (VC-N), merge friction labels (MF-N), and the total labels (Total) of the proposed datasets according to the difficulty levels. The difficulty levels are paired with the ones in Table 5. "Easy k" or "Hard k" means there are k group of version labels.

composite-intent splitting creates atomic intents that are easier to check. Fore example, checking the composite intent "$Hotel\&Taxi$" requires more computation them simply checking atomic intent "$Hotel$" or "$Taxi$". The statistics is shown in Table A5.

A.7 More Analysis

Performance variance under different datasets Generally, LS Focal loss is the most powerful method, but it performs poorly when available data is small. As presented in Table 1, four datasets used in our experiments have varied label types and instance amounts (Ochal et al., 2023). Since ATIS has 66 initial intents but only 4455 training samples, and Multi-label CE is less data-hungry, Multi-label CE slightly outperforms LS Focal Loss in this setting.

Similarly, the basic classifier has very low recall on the dataset of ATIS-VCS and SNIPS-VCS, since they have a larger number of labels than MultiWOZ-VCS and CrossWOZ-VCS. A larger number of labels in a dataset results in a harder difficulty, which is proved in the performance gap in 4 datasets. The basic classifier is trained with the data using the data that each sample is only provided with only one label. Even with a model structure that can perform multi-label classification, the basic classifier generally only outputs one intent because of the data. Intuitively, larger candidate pools (ATIS and SNIPS) will make the recall worse, because the model output intent will be less likely to hit the ground truth.

Beyond the amount of training data and labels, the length of input utterance could also affect the results. Multi-WOZ-VCS contains samples with very short sentences, so different models and sizes do not make a great difference in Multi-WOZ. This dataset does not need models with strong semantics understanding ability. For ATIS-VCS and SNIPS-VCS, sentences are long, so the task becomes harder. Also, a stronger model with more parameters has better performance.

Challenges of detecting semantic overlap The results of Table 2 show that the proposed benchmark is hard for the basic classifier. And the proposed methods of overlapping intents detection are effective. However, these methods are only effective in Precision, Recall, and F1. In real products, EM is the most important metric. The proposed methods are far from the Upper Bound in EM metrics. Thus we believe that the benchmark is challenging and more powerful methods need to be proposed.

Also, the experiment results of Table 3c and Table A10 provide a comparison of results under the different numbers of conflict labels and merge friction labels. We change the ratio of updated labels to control the degree of update. The difficulty is controlled by the ratio of updated labels. A harder degree means a larger ratio of updated labels. This result can help us see the details of before and after adding entangled intents. As we can see, a larger degree of updating entangled intents makes

Size	ATIS-VCS P	R	F1	EM	SNIPS-VCS P	R	F1	EM	CrossWOZ-VCS P	R	F1	EM	MultiWOZ-VCS P	R	F1	EM
BERT-small	66.28	94.10	77.78	47.37	85.67	96.32	90.68	76.57	93.60	97.70	95.60	90.23	82.68	90.16	86.26	81.73
BERT-base	84.17	88.81	86.43	77.05	95.85	95.95	95.90	92.86	97.00	88.37	92.48	80.34	88.62	86.45	87.52	85.85
BERT-large	87.14	96.46	91.57	79.34	97.32	97.58	97.45	96.71	97.35	79.72	87.66	68.69	88.60	86.11	87.34	85.85

Table A6: Additional study on different size of BERT including BERT-Small, BERT-Base and BERT-Large. We use Label-Smoothing Focal Loss method to get all the results. Metrics in this table are Precision, Recall, F1-Score and Exact Match Ratio.

Model	ATIS-VCS P	R	F1	EM	SNIPS-VCS P	R	F1	EM	CrossWOZ-VCS P	R	F1	EM	MultiWOZ-VCS P	R	F1	EM
BERT-base	84.17	88.81	86.43	77.05	95.85	95.95	95.90	92.86	97.00	88.37	92.48	80.34	88.62	86.45	87.52	85.85
RoBERTa-base	87.37	95.02	91.03	79.91	96.42	96.26	96.34	95.43	94.90	90.04	92.41	79.80	85.94	87.30	86.62	83.86
AlBERT-base	88.10	81.43	84.64	68.95	91.69	85.42	88.45	74.71	96.83	75.08	84.58	68.69	86.35	85.50	85.92	84.27
DeBERTa-base	90.52	92.62	91.56	85.39	96.90	85.58	90.89	75.14	96.40	95.46	95.93	88.08	88.99	84.3	86.61	80.75

Table A7: Results of four models including BERT, RoBERTa, AlBERT and DeBERTa. We Label-Smoothing Focal Loss method to get all the reported results. Metrics in this table are Precision, Recall, F1-Score and Exact Match Ratio.

LSR	ATIS-VCS P	R	F1	EM	SNIPS-VCS P	R	F1	EM	CrossWOZ-VCS P	R	F1	EM	MultiWOZ-VCS P	R	F1	EM
0.1	65.99	94.37	77.67	47.72	95.85	95.95	95.90	92.86	97.00	88.37	92.48	80.34	85.26	88.53	86.86	83.75
0.2	84.17	88.81	86.43	77.05	96.63	93.53	95.05	89.00	95.53	70.14	80.89	40.66	86.88	87.16	87.02	84.58
0.4	91.59	79.53	85.13	73.63	97.41	81.21	88.58	65.86	95.34	69.79	80.59	40.23	88.62	86.45	87.52	85.85

Table A8: Results of different label smoothing rate used in Label-Smothing Focal Loss including 0.1, 0.2, and 0.4. We use Label-Smoothing Focal Loss method to get all the reported results. Metrics in this table are Precision, Recall, F1-Score, and Exact Match Ratio.

NSN	ATIS-VCS P	R	F1	EM	SNIPS-VCS P	R	F1	EM	CrossWOZ-VCS P	R	F1	EM	MultiWOZ-VCS P	R	F1	EM
1	50.46	96.84	66.35	55.59	94.30	93.16	93.73	85.14	97.97	49.24	65.54	42.97	86.19	87.06	86.62	82.79
2	69.95	92.20	79.55	67.92	95.97	87.74	91.67	74.71	97.88	42.01	58.78	35.61	88.28	81.15	84.57	74.50
4	87.40	86.87	87.14	76.37	97.00	71.58	82.37	41.14	97.81	36.25	52.90	28.41	90.15	68.10	77.59	50.65
8	93.00	77.36	84.46	71.23	96.96	63.84	76.99	32.57	97.67	32.45	48.72	22.42	91.34	60.23	72.60	35.71

Table A9: Results of five Negative Sample number including 1, 2, 4 and 8. We use NS method to get all the reported results. Metrics in this table are Precision, Recall, F1-Score, and Exact Match Ratio.

Difficulty	ATIS-VCS P	R	F1	EM	SNIPS-VCS P	R	F1	EM
Easy 1	93.90	98.55	96.17	91.44	98.34	98.67	98.50	98.43
Easy 2	94.47	98.32	96.46	92.35	96.42	95.45	95.93	94.71
Easy 3	88.15	98.75	93.15	82.99	96.98	96.47	96.72	95.29
Normal	84.17	88.81	86.43	77.05	95.85	95.95	95.90	92.86

Difficulty	CrossWOZ-VCS P	R	F1	EM	MultiWOZ-VCS P	R	F1	EM
Hard 1	94.50	63.65	76.07	44.10	81.28	88.98	84.96	82.20
Hard 2	95.53	70.14	80.89	40.66	82.93	88.04	85.41	83.36
Hard 3	95.34	69.79	80.59	40.23	84.68	87.96	86.29	83.51
Normal	97.00	88.37	92.48	80.34	88.62	86.45	87.52	85.85

Table A10: Results of 3 difficulty including 1, 2 and 4 in the four datasets: ATIS, SNIPS, CrossWOZ and MultiWOZ. Metrics in this table are F1-Score, Exact Match Ratio and Zero One Loss. 1 is the easiest and 4 is hardest.

the model perform worse.

A.8 Visualization and Error Analysis

In Figure 3, Figure A2, Figure A3, and Figure A4, we present the co-occurrence matrix between predictions on the Multi-Label Focal Loss method for ATIS-VCS, SNIPS-VCS. From Figure A2, we can see that it's challenging to capture the semantic overlap of labels under diverse intents but limited training instances, as the occurrence matrix is more noisy than that of SNIPS-VCS. Similarly, compare with Figure A3 and Figure A4, we can see a clearer pattern of grasping version conflict and merge frictions under the dataset of CrossWOZ than MultiWOZ, as the performance is also slightly better in Table 2. From Figure A3, we can discover a minor bias in the model's prediction, which is the imbalance occurrence between "Request_v1" and "Request_v2". While in Figure A4, the model does not handle the compound intents "hotel&taxi" and "restaurant&train", as the correlation between "hotel&taxi" and "hotel" is weak, and the co-occurrence between "restaurant&train" and "train" is very low.

In Figure A1, we visualize the model's "behavior" on different version labels in the test set of SNIPS-VCS. Different colors represent different labels, while different shapes represent different clusters. From the figure, we can see that different versions of the same intent family are clustered together. We first use t-SNE (van der Maaten and Hinton, 2008) to reduce the co-occurrence matrix to two dimensions, then use DBSCAN (Ester et al., 1996) to cluster the labels.

B Detailed Explanation of the Setting

B.1 Definition of the Setting

Our setting is not a scenario where each sample is provided with ground truth. If that were the case, we would not encounter semantic duplications (i.e. version conflict) and semantic overlap (i.e. merge friction). The objective of the benchmark is to evaluate if a model trained with imperfect data (i.e., samples labeled only with one of the ground truth values) can achieve perfect predictions (i.e., accurately predict all ground truth values, including $l_1 \& l_2$, l_1 and l_2). The primary goal of this setup is to address the real-world issue where users introduce new labels during version upgrades without considering the correlations between existing and newly added labels. In this case, the models are trained using positive but unlabeled data and then tested using ground truth.

The objective of this setting is to ensure that the model efficiently utilizes both existing and newly added data, enabling it to perform well on both types of data while minimizing costs. The proposed benchmark primarily focuses on investigating strategies for effectively leveraging both pre-upgrade and post-upgrade data, which may contain inconsistent labels (i.e. positive but unlabeled data), and building a cost-effective model that performs well on both data.

B.2 Positive and Unlabeled Data

Regarding positive and unlabeled data (Hammoudeh and Lowd, 2020; Bekker and Davis, 2020), a common definition of positive and unlabeled data refers to the presence of unlabeled data where the positive labels are not explicitly identified as positive. In our setting, positive and unlabeled data means that not all positive labels are provided in the training set. Only one of the ground truth values is designated as positive, while all other labels are considered negative. Consequently, only the positive label can be relied upon as trustworthy, as the other negative labels may mistakenly include positive labels.

B.3 ChatGPT In-context Learning

Our template contains exemplars and candidate options. Regarding the selection of exemplars, we randomly select one single exemplar for each label. We use five random seeds to select exemplars and present the order of the exemplars. Then we will provide all candidate options, then ask ChatGPT to choose one or more than one option. We use five random seeds to select the present order of options, which is to prevent potential order bias. We report their average performance. About post-processing, we use Python split to get multiple outputs from the generated string, then we use string matching to match each output with candidates.

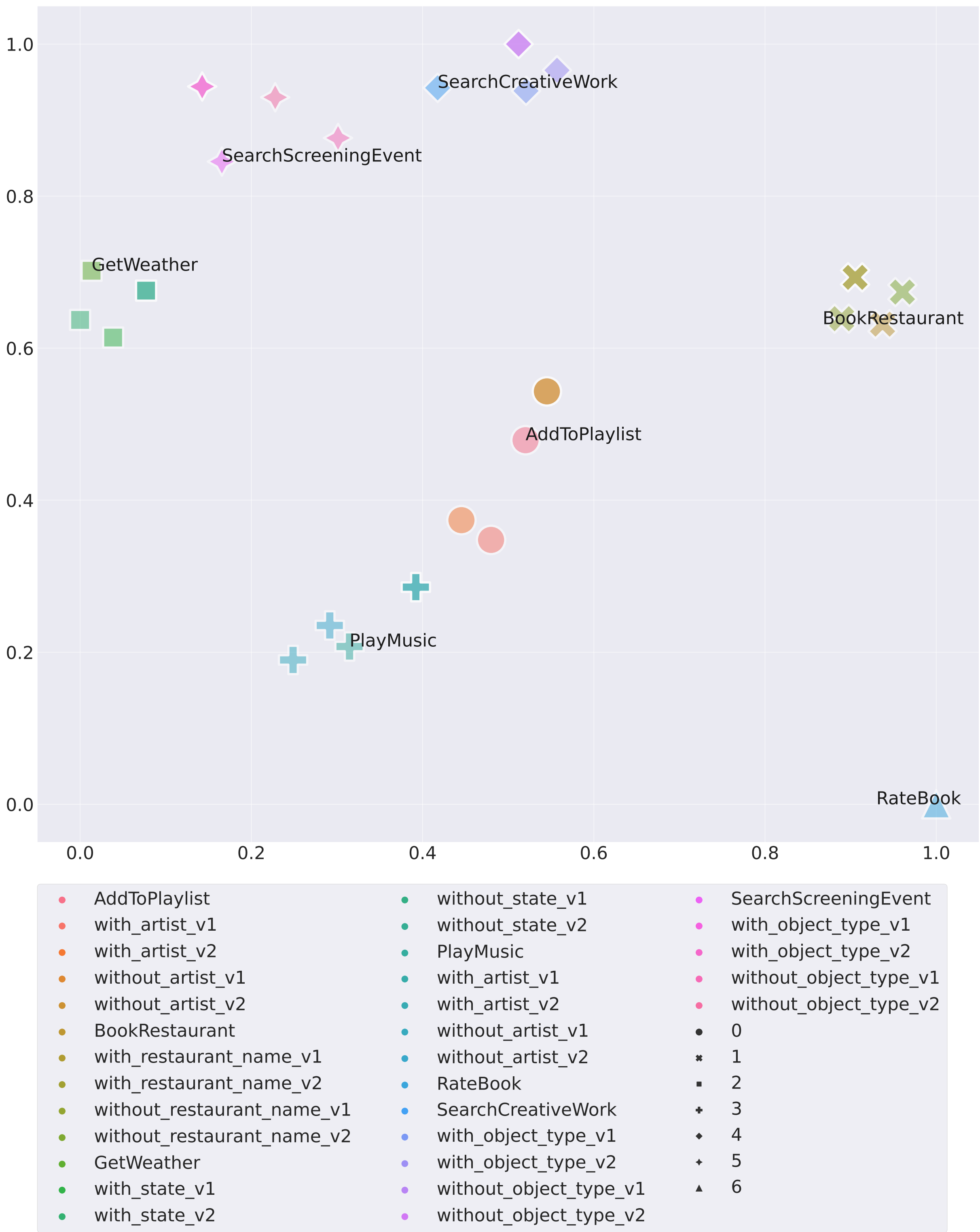


Figure A1: t-SNE dimensionality reduction and DBSCAN clustering for SNIPS. Different colors represent different intents while different shape reperesent differt clusters.

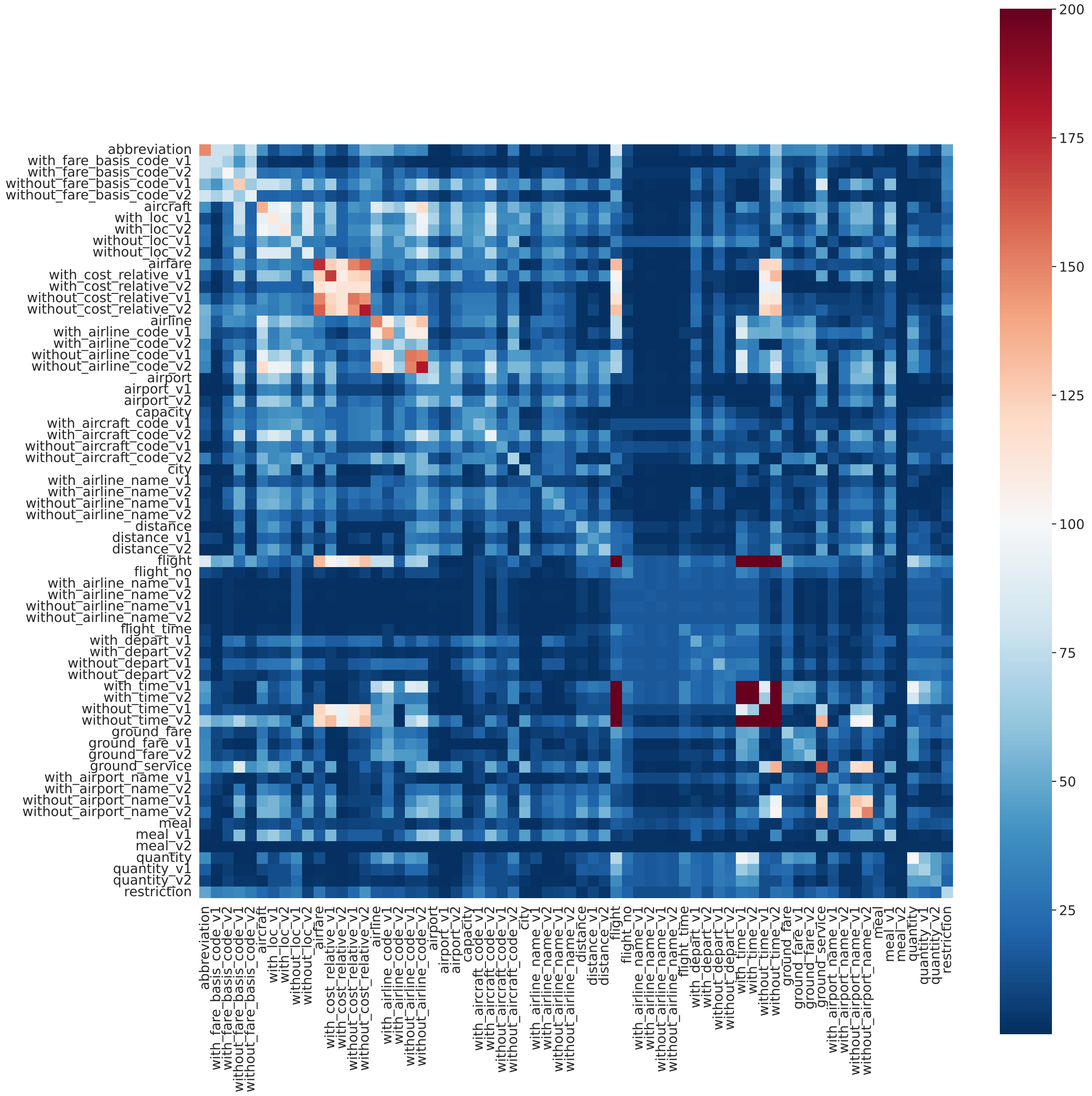


Figure A2: Display of the co-occurrence matrix between labels based on the model output of Multi-Label Focal Loss method for the test set of ATIS-VCS. Different colors indicate different co-occurrence frequency of labels.

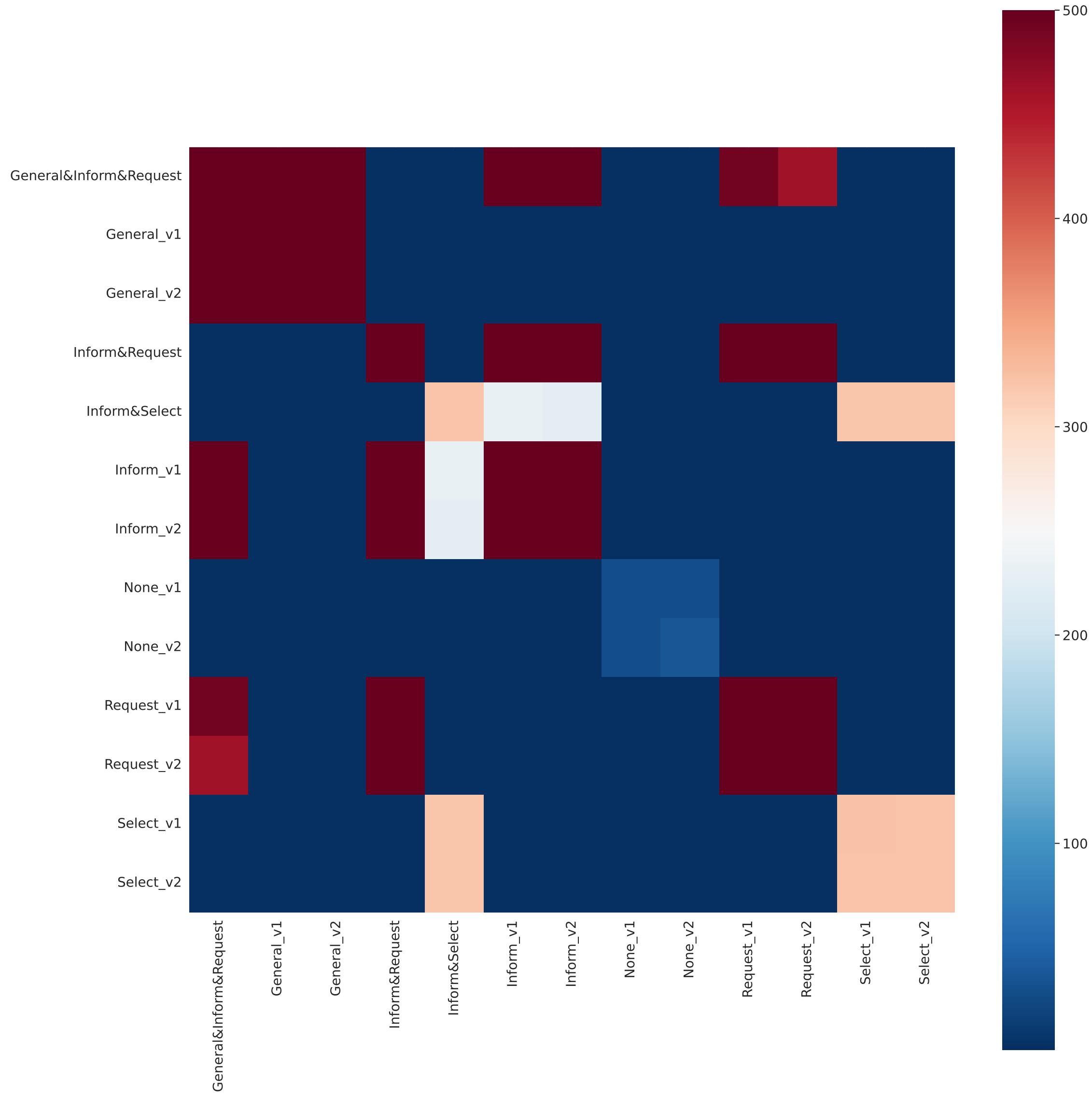


Figure A3: Display of the co-occurrence matrix between labels based on the model output of Multi-Label Focal Loss method for the test set of CrossWOZ-VCS. Different colors indicate different co-occurrence frequency of labels. For better visualization, We remove the labels that have fewer than 10 instances in the test set.

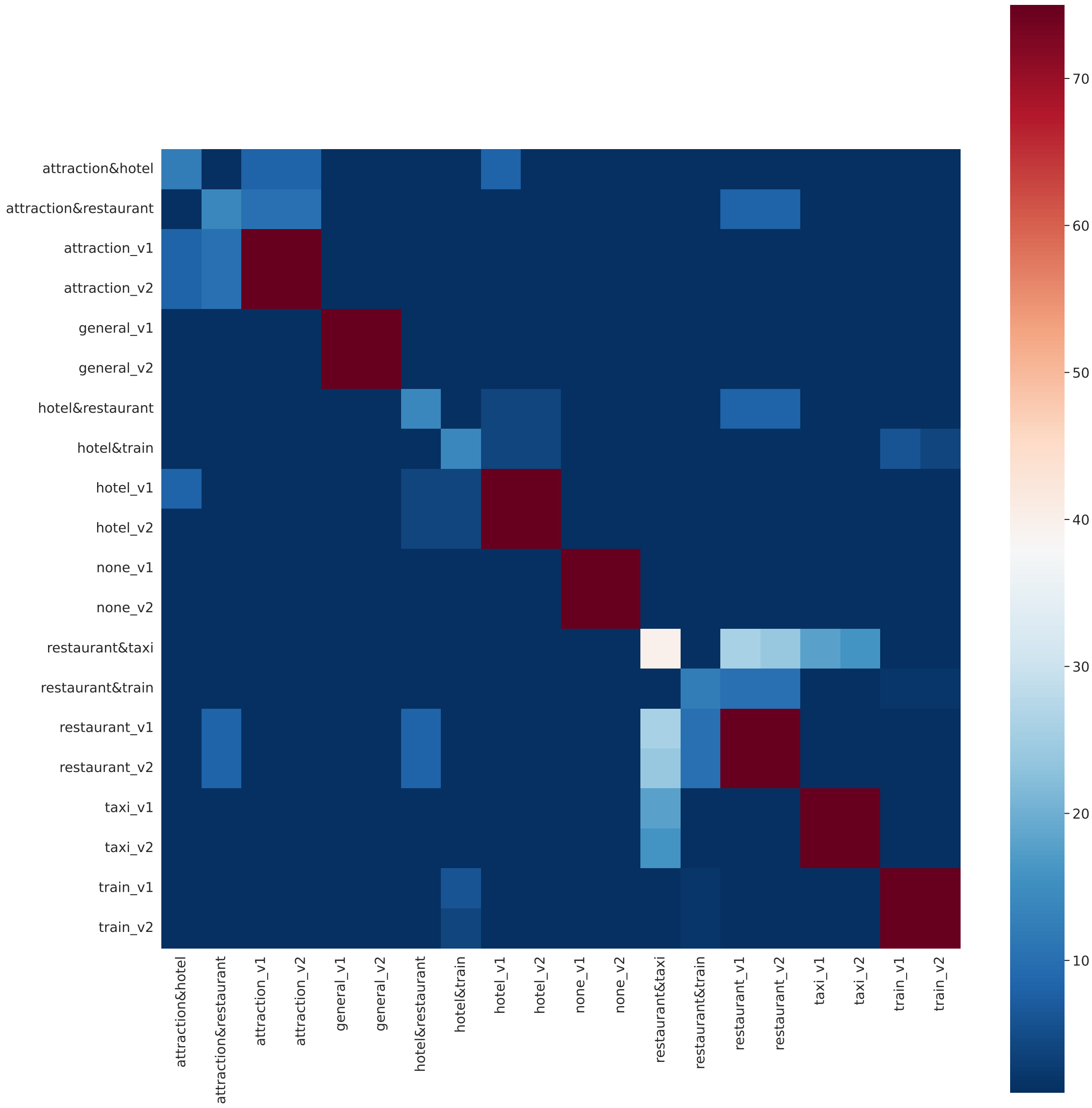


Figure A4: Display of the co-occurrence matrix between labels based on the model output of Multi-Label Focal Loss method for the test set of MultiWOZ-VCS. Different colors indicate different co-occurrence frequency of labels. For better visualization, We remove the labels that have fewer than 10 instances in the test set.

Dialogue:
what is the cost of the air taxi operation at philadelphia international airport.
Question:
What is the intent of this dialogue?
Answer: ground_fare

... (Other 65 examples for each intent)

Given a dialogue, please answer the intent of the dialogue from options:
abbreviation, abbreviation_with_fare_basis_code_v1, abbreviation_with_fare_basis_code_v2, abbreviation_without_fare_basis_code_v1, abbreviation_without_fare_basis_code_v2, aircraft, aircraft_with_loc_v1, aircraft_with_loc_v2, aircraft_without_loc_v1, aircraft_without_loc_v2, airfare, airfare_with_cost_relative_v1, airfare_with_cost_relative_v2, airfare_without_cost_relative_v1, airfare_without_cost_relative_v2, airline, airline_with_airline_code_v1, airline_with_airline_code_v2, airline_without_airline_code_v1, airline_without_airline_code_v2, airport, airport_v1, airport_v2, capacity, capacity_with_aircraft_code_v1, capacity_with_aircraft_code_v2, capacity_without_aircraft_code_v1, capacity_without_aircraft_code_v2, city, city_with_airline_name_v1, city_with_airline_name_v2, city_without_airline_name_v1, city_without_airline_name_v2, distance, distance_v1, distance_v2, flight, flight_no, flight_no_with_airline_name_v1, flight_no_with_airline_name_v2, flight_no_without_airline_name_v1, flight_no_without_airline_name_v2, flight_time, flight_time_with_depart_v1, flight_time_with_depart_v2, flight_time_without_depart_v1, flight_time_without_depart_v2, flight_with_time_v1, flight_with_time_v2, flight_without_time_v1, flight_without_time_v2, ground_fare, ground_fare_v1, ground_fare_v2, ground_service, ground_service_with_airport_name_v1, ground_service_with_airport_name_v2, ground_service_without_airport_name_v1, ground_service_without_airport_name_v2, meal, meal_v1, meal_v2, quantity, quantity_v1, quantity_v2, restriction.
You can answer 1 to 3 intents.

{Dialogue}

Table B11: Prompt template for ChatGPT for in-context learning. Our template contains exemplars and candidate options. Regarding the selection of exemplars, we randomly select one single exemplar for each label. We use five random seeds to select exemplars and present the order of the exemplars. Then we will provide all candidate options. We use five random seeds to select the present order of options.