A Label-Aware BERT Attention Network for Zero-Shot Multi-Intent Detection in Spoken Language Understanding

Ting-Wei Wu, Ruolin Su, Biing-Hwang Juang
Department of Electrical and Computer Engineering
Georgia Institute of Technology
{waynewu,ruolinsu}@gatech.edu, juang@ece.gatech.edu

Abstract

With the early success of query-answer assistants such as Alexa and Siri, research attempts to expand system capabilities of handling service automation are now abundant. However, preliminary systems have quickly found the inadequacy in relying on simple classification techniques to effectively accomplish the automation task. The main challenge is that the dialogue often involves complexity in user's intents (or purposes) which are multi-proned, subject to spontaneous change, and difficult to track. Furthermore, public datasets have not considered these complications and the general semantic annotations are lacking which may result in zero-shot problem. Motivated by the above, we propose a **L**abel-**A**ware **B**ERT **A**ttention **N**etwork (LABAN) for zero-shot multi-intent detection. We first encode input utterances with BERT and construct a label embedded space by considering embedded semantics in intent labels. An input utterance is then classified based on its projection weights on each intent embedding in this embedded space. We show that it successfully extends to few/zero-shot setting where part of intent labels are unseen in training data, by also taking account of semantics in these unseen intent labels. Experimental results show that our approach is capable of detecting many unseen intent labels correctly. It also achieves the state-of-the-art performance on five multi-intent datasets in normal cases.

1 Introduction

In spoken language understanding (SLU) of task-oriented dialog systems, each utterance is often interpreted as a kind of action being performed by the speaker, which we call **speech or dialog acts** (Abbeduto, 1983). These acts may commit speakers to some course of actions, like asking or acknowledging, along with a series of distinctive semantic notions involved in a task. Usually the system forms the semantic frames by identifying intents and slots to express dialog acts. For instance, given a sample utterance, "Are there any accidents on my route to work at 10 ?", the intent detection task will first identify intents, i.e., *'Get Info Traffic'*, *'Get Location Work'* and then the slot-filling task will predict a slot such as (*time:10*). In such case, an 'intent label' for an utterance is defined as a purpose or a goal that clearly states user's act.

Dominant SLU systems have adopted several techniques to predict single intents by treating it as a multi-class classification problem (Gao et al., 2018; Goo et al., 2018; Qin et al., 2019). However, in real world scenario, many utterances may have multiple intents (Li et al., 2018b; Rastogi et al., 2019) like the above example. Multi-intent SLU often requires more sophisticated reasoning on given utterances to disambiguate different intent natures. Gangadharaiah and Narayanaswamy (2019) first explored the joint multi-intent and slot-filling task by treating multi-intents as a single context vector, but not scalable to a large number of intents. Qin et al. (2020) further proposed a state-of-the-art model to consider each intent-slot interaction via adaptive graph attention. However, these approaches cannot successfully tackle more complex multi-intent scenarios when sentences may not have explicit conjunctions.

The second challenge in SLU intent detection is **intent fluidity** variation, which we refer to the extent of naturalness when a dialogue progresses. In less stylized conversations, they usually contain a less bounded set of intents which may change with dialog context/states. Thus, usually some utterances' intents may not be seen during training and this problem deteriorates in the multi-intent scenario (Xia et al., 2020). Second, there is no rigorous definition of an intent annotation format or how many intents should be defined. Therefore, conventional models trained on one dataset with a fixed set of intent labels may possibly fail to detect a new in-domain intent. We refer it to the **zero-shot** problem. Larson et al. (2019) suggests a two-stage

Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 4884–4896
November 7–11, 2021.

process to first classify if a query is in-scope; then to assign intents. However, it cannot scale easily to unseen intents in multi-intent scenario.

To tackle the above two challenges, we found that leveraging embedded semantics in intent labels may be useful. In conventional intent classification, these systems usually classify an utterance to a label which is represented by an indexed ID like 0 (i.e. one-hot encoding). However, representing intents with indexed IDs fails to consider embedded semantics in the labels too. For instance, we can use words 'get' and 'direction' in an intent label 'get direction' to help with identifying semantically equivalent words in an utterance, i.e., I 'want direction' to SF. For a given set of intent labels within one domain, we can compare the semantic similarity between words in an utterance and words in these intents. Similarly in the zero-shot setting, even if some intents may not be visible during training, we could still compare the word semantics in these intents with a new utterance.

In this paper, we propose our new framework: **L**abel-**A**ware **B**ERT **A**ttention **N**etwork (LABAN) in Figure. 1. We first introduce BERT to capture the multi-intent natures when utterances do not have explicit conjunctions. Then, instead of treating intent labels only for indexed IDs, we use words in each intent label in training data to construct a label embedding space. After encoding an utterance and all intents in a given training set for embeddings separately, a label-aware layer will generate scores of how likely this utterance belongs to each intent. To accommodate the zero-shot case, we could additionally introduce unseen intents' embeddings too to jointly construct the embedding space. In contrast with prior works' limited predictability only on seen intents, our model unfreezes the constraint by considering semantics in intent labels to deal with new unseen labels. The code and resources are released in https://github.com/waynewu6250/LABAN. The paper has the following contributions:

1. We extend the first use of BERT into multi-intent SLU scenario with a simple but powerful label-aware approach.

2. We successfully demonstrate LABAN's effectiveness to deal with unseen multiple intents and fast harness the intent detection task by training with few data of unseen intents.

3. We compare the LABAN's performance on five extended and complex multi-intent datasets that show significant improvement over previous methods and baselines by considering the contextualized information from BERT and label semantics.

2 Related Work

Multi-intent Detection Intent detection mainly aims to classify a given utterance with its intents from user inputs. Different approaches such as convolutional-LSTM and capsule network have been proposed to solve the problem (Qian, 2017; Liu et al., 2017; Xia et al., 2018). Considering intents highly associated with slot-filling, many joint models (Goo et al., 2018; Li et al., 2018a; Qin et al., 2019; E et al., 2019; Liu et al., 2019b) utilize intent information like gradients or cross-impact networks to further reinforce the slot-filling prediction. However these methods do not consider multiple intent cases. Therefore, Rychalska et al. (2018) first adopted hierarchical structures to identify multiple user intents. Gangadharaiah and Narayanaswamy (2019) and Qin et al. (2020) further exploited interactive relations between intents and slots. Wu et al. (2021) leveraged the dialog context to better harness the joint tasks. Our model follows these models' paradigm and focuses on more complex cases: 1) Multi intents no longer exist in separate parts of the sentence which our BERT introduction can be beneficial and 2) Some testing intents are not available during training.

Zero-shot Learning Zero-shot learning (ZSL) aims to recognize objects whose instances may not be seen during training (Lampert, 2014). Early works usually focused in the fields of computer vision (Lampert, 2014; Al-Halah et al., 2016; Norouzi et al., 2014). They adopted a two-stage approach to first identify object's attributes and estimated class posteriors based on similarity, which often suffered from domain shift between intermediate and target tasks. Recent advances in ZSL directly learned a mapping between feature and semantic spaces (Palatucci et al., 2009; Akata et al., 2016; Frome et al., 2013) or built a common intermediate space (Zhang and Saligrama, 2015; Xian et al., 2017). Similar treatment could be applied in natural language. Chen et al. (2016) proposed CDSSM to consider cosine similarity of deep semantics from utterances and intents. Xia et al. (2018) and Liu et al. (2019a) extended ZSL in user intent detection with capsule neural networks. Si

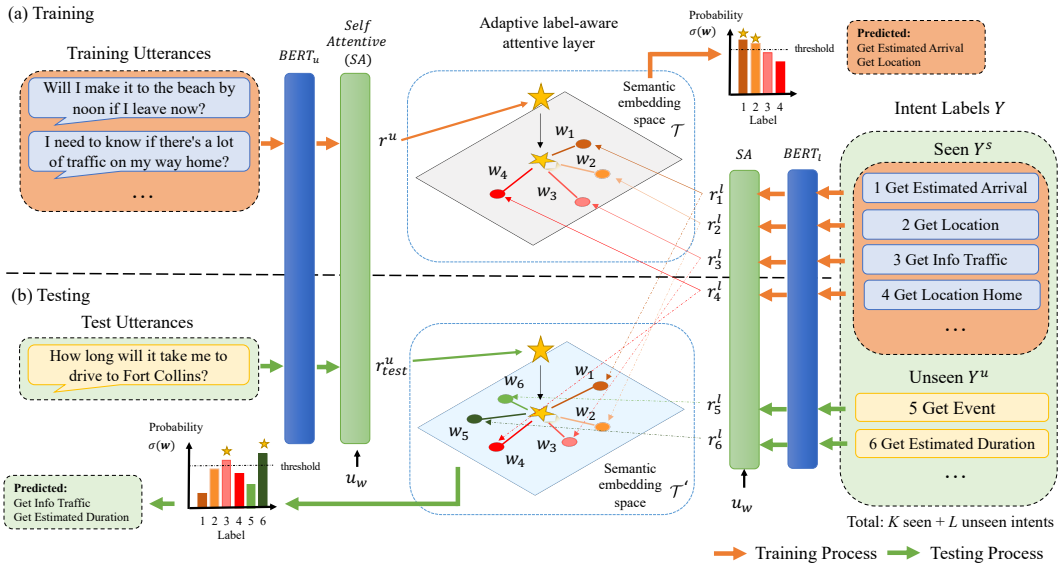

Figure 1: This figure shows the overall LABAN framework. (a) During training phase, two BERT encoders will encode both the utterance and all seen intent labels. Then the utterance embedding will be projected onto a constructed semantic embedding space $\mathcal{T}$ with projected weights as scores. (b) During testing phase, new unseen intents will also be encoded and participate in constructing $\mathcal{T}'$ to generate scores based on a new utterance.

et al. (2021) proposed disentangled intent representations for multi-task training. We follow these works and extend to multi-intent detection cases with intent semantics and pretrained models.

3 Problem Formulation

In this section, we formally state the multi-intent detection problem in the normal and zero-shot case.

Multi-Intent Detection. Given a labeled training dataset where each sample has the following format: (x, y) where x is an utterance and $y = (y_1, y_i, ..., y_K) \in \{0, 1\}^K$ is a set of multiple binary intent labels. Each y_i will belong to a set Y^s of K seen intents. We aim to classify an utterance x_{seen} in the seen intent classes Y^s.

Zero-shot Multi-Intent Detection. Given a labeled training dataset (x, y) where $y \in Y^s$, in testing we aim to classify an utterance x_{unseen} with its correct intent categories $y_{unseen} = (y_1, y_i, ..., y_{K+L}) \in \{0, 1\}^{K+L}$ from the seen and unseen intent classes $Y = Y^s \cup Y^u$. Y^u will be a set of L unseen intents which is given along with Y^s as domain ontology during testing, but not visible in training.

4 Approach

4.1 Utterance encoder

BERT is a multi-layer transformer-based encoder containing multi-head self-attention layers (Devlin et al., 2019). Models fine-tuned on BERT have achieved several benchmark results in many natural language tasks (Sun et al., 2020). Therefore, we first adopt one BERT $BERT_u$ to encode an input utterance $x = (w_1, ..., w_{T_u})$. Here, we will pad it up to a max sequence length T_u.

$$h^u = BERT_u(x) \tag{1}$$

where $h^u \in \mathbb{R}^{T_u \times H}$ is the token-level representations of x and H is the hidden size of BERT. Then, we adopt two methods to further encode them into a sentence embedding $r^u \in \mathbb{R}^H$. First, we could take the hidden state h_1^u from the first time step of [CLS] as $r^u = h_1^u$ (BERT-finetune). Or to better consider the individual word importance to the overall sentence embedding, we follow the work in Lin et al. (2017) to use a self-attentive network.

$$\bar{h}_t^u = W h_t^u + b_w \tag{2}$$

$$\alpha_t = \frac{e^{\bar{h}_t^{u\,T} u_w}}{\sum_{t'} e^{\bar{h}_{t'}^{u\,T} u_w}} \tag{3}$$

$$r^u = \sum_t \alpha_t h_t^u \tag{4}$$

where each h_t^u in h^u are fed into an affine transformation (W, b_w) and output $\bar{h}_t^u$. Then $\{\alpha_t\}$ represents the similarity scores between each h_t^u and K heads of learnable context vectors u_w as the global sentence views; for each head, we can get

a sentence representation r_h^u. Finally we will concatenate all heads for the final representation r^u.

4.2 Adaptive label-aware attentive layer

Inspired by few-shot learning works (Snell et al., 2017; Reimers and Gurevych, 2019), instead of classifying utterance into a predefined set of intents, we instead leverage the linear approximation idea (del Pino and Galaz, 1995) to help us determine the intents of an utterance. The linear approximation problem states that let $\mathcal{S}$ be a Hilbert space and $\mathcal{T}$ be a subspace of $\mathcal{S}$, given a vector $z \in \mathcal{S}$, we would like to find the closest point $\hat{z} \in \mathcal{T}$ to z. It turns out that the solution of $\hat{z} = \sum_{k=1}^{N} \beta_k v_k$ will be a linear combination of a basis $v_1, ..., v_N$ for $\mathcal{T}$ of N dimension. $\beta = \mathbf{G}^{-1}\mathbf{b}$ where an element in the Gram matrix $G_{k,n} = \langle v_n, v_k \rangle$ and $b_n = \langle z, v_n \rangle$.

To transform the above idea into a multi-intent detection setting, we first construct an intent embedding subspace $\mathcal{T}$ with a basis $\{r_1^l, ..., r_K^l\}$ given a set of K intents Y^s. To obtain $\{r_1^l, ..., r_K^l\}$, we adopt another BERT $BERT_l$ to encode K intents. Namely, for every intent y_i in a given set Y^s, which could be expressed as a word sequence $(w_1, ..., w_{T_l})$, we similarly use another BERT $BERT_l$ with the self-attentive layer mentioned in section 4.1 to encode it into an intent embedding r_i^l. The reason to use a different BERT from $BERT_u$ is that intents often have very different syntactic structures (i.e. no subjects) compared to the utterances.

By such intent encoding, we will obtain K intent embeddings as our basis $\{r_1^l, ..., r_K^l\}$ to construct an intent embedding space $\mathcal{T}$. Then shown in Figure. 1, for an utterance r^u, we can project it onto $\mathcal{T}$ to obtain its linear approximation $\hat{r}^u = \sum_{i=1}^{K} w_i r_i^l$, where $\mathbf{w} \in \mathbb{R}^K$ could be computed as $\mathbf{w} = \sqrt{H}\mathbf{G}^{-1}\mathbf{b}$. And the Gram matrix $\mathbf{G}$ and $\mathbf{b}$ are the followings:

$$\mathbf{G} = \begin{bmatrix} \langle r_1^l, r_1^l \rangle & \cdots & \langle r_K^l, r_1^l \rangle \\ \vdots & \ddots & \vdots \\ \langle r_1^l, r_K^l \rangle & \cdots & \langle r_K^l, r_K^l \rangle \end{bmatrix} \quad (5)$$

$$\mathbf{b} = \begin{bmatrix} \langle r^u, r_1^l \rangle \\ \vdots \\ \langle r^u, r_K^l \rangle \end{bmatrix} \quad (6)$$

To note, we assume $\{r_1^l, ..., r_K^l\}$ are linearly independent since each vector represents the concept of an intent which should not be a linear combination of other intent vectors. Hence, $\mathbf{G}$ is guaranteed positive definite and will have an inverse. Here we further time a scaling factor $\sqrt{H}$ to compute $\mathbf{w}$ for empirical consideration since $\mathbf{G}^{-1}$ tends to lead overall product into small values.

After obtaining $\mathbf{w}$, these projection weights can be viewed as scores of how likely an utterance x belong to each intent y_i. We can follow Qin et al. (2020) to treat it as a multi-label classification task and generate the logits $\hat{y} = \sigma(\mathbf{w})$ by sending $\mathbf{w}$ into a sigmoid function σ. Finally we can have the intent detection objective as a binary cross entropy loss where N is number of samples:

$$\mathcal{L} := -\sum_{i=1}^{N}\sum_{j=1}^{K}(y_j^{(i)} log(\hat{y}_j^{(i)}) + (1 - y_j^{(i)}) log(1 - (\hat{y}_j^{(i)})) \quad (7)$$

During testing, after obtaining $\hat{y} \in \mathbb{R}^K$ as probabilities of the utterance belong to each intent, we can set a threshold t where $0 < t < 1.0$ as a hyperparameter to select the final predicted intents. For instance, if we have $\hat{y} = \{0.3, 0.6, 0.9, 0.1, 0.4\}$ and $t = 0.5$, the intents are predicted as $\{2, 3\}$.

4.3 Zero-shot setting

For normal multi-intent detection, after training, for a given K seen intent set Y^s, we could use the method in section 4.2 to calculate the scores of a new utterance x_{seen} with respect to each intent. Similarly, we could easily extend it into the zero-shot setting. First we will train $BERT_u, BERT_l$ with the training data of a given K seen intent set Y^s. Then, during testing, given a new L unseen intent set Y^u, we could also encode these intents into intent embeddings $\{r_1^l, ..., r_L^l\}$ with the trained $BERT_l$ too. Finally, plus the seen intent set Y^s, we could construct an extended intent subspace $\mathcal{T}'$ with a basis of $\{r_1^l, ..., r_K^l, r_{K+1}^l, ..., r_{K+L}^l\}$ and similarly generate scores for each seen and unseen intents with a new utterance x_{unseen}.

5 Experimental Setting

5.1 Datasets

We use three widely used public multi-intent single-sentence datasets: MixATIS, MixSNIPS (Qin et al., 2020; Hemphill et al., 1990; Coucke et al., 2018) and Facebook Semantic Parsing System (FSPS) dataset (Gupta et al., 2018) and two multi-intent dialogue datasets: Microsoft dialogue challenge dataset (MDC) (Li et al., 2018b) and Schema-Guided Dialogue dataset (SGD) (Rastogi et al.,

Dataset	Data Type	train/val/test	Total Labels
MixATIS	single	18k/1k/1k	17
MixSNIPS	single	45k/2.5k/2.5k	7
FSPS	single	31k/4.4k/9k	24
MDC	dialogue	45k/15k/15k	11
SGD	dialogue	198k/66k/66k	18

Table 1: Dataset statistics

2019) for our experiments. For FSPS, we focus on predicting all intents regardless of their positions for each utterance. For MDC and SGD, we treat each utterance as an individual sample with multiple user and system acts as intents for experiments. We use all datasets for normal and zero-shot multi-intent detection and include single intent detection results with ATIS (Hemphill et al., 1990) and SNIPS datasets (Coucke et al., 2018). The detailed data statistics is shown in Table. 1.

For zero-shot task, we use single sentence datasets MixATIS, MixSNIPS and FSPS for experiments. We subsample each dataset 5 times with the same train/valid/test number and report the average results of 5 random splits. In each split, we simulate the situation where training data only contain a part of intent labels and test will have all intent labels. For instance, MixATIS has totally 17 labels, we maintain K $<$ 17 possible intents seen in training set and the testing set has all 17 intents. In experiments, we set 4 possible values of K in each three datasets. For few-shot task, we add 5% and 10% testing data into the training data and predict the rest testing data performance. We also replace BERT with two variations: ALBERT, TOD-BERT as our utterance encoder for additional baselines.

5.2 Baselines

We compare the normal multi-intent detection results with three competitive baseline models:
1. **Stack-Prop** which uses two stacked encode-decoder structures for joint intent and slot filling tasks (Qin et al., 2019).
2. **Joint MID-SF** which first considers multi-intent detection task in use of BiLSTMs (Gangadharaiah and Narayanaswamy, 2019).
3. **AGIF** uses graph interactive framework to consider fine-grained information (Qin et al., 2020).

We also compare zero-shot multi-intent detection results with seven competitive baselines:
1. **BERT-finetune** uses BERT as the encoder and increases the total output size of the final fully-connected layer on top of it (Devlin et al., 2019).
2. **Zero-shot LSTM** uses two LSTM encoders to encode utterances and intents; then acquires scores with dot product (Kumar et al., 2017).
3.**CDSSM** uses convolutional deep structured model to calculate cosine similarities between embeddings (Chen et al., 2016).
4. **Zero-shot BERT** uses BERT as the encoder for Zero-shot LSTM (Kumar et al., 2017) instead.
5. **CDSSM BERT** uses BERT as the encoder for CDSSM (Chen et al., 2016) instead.
6. **ALBERT-LA** uses ALBERT as encoder along with our label-aware layer (Lan et al., 2020).
7. **TOD-BERT-LA** uses TOD-BERT, a pretrained encoder for task-oriented dialogs, along with our label-aware attentive layer (Wu et al., 2020).

5.3 Experimental setting

We use the pretrained BERT with 12 hidden layers of 768 units and 12 self-attention heads. The model is trained for 50 epochs and saved with the best performance on the validation set. For zero/few-shot setting, we randomly pick a number of intents to be unseen in the training set, run experiments for 5 different splits and report the average. We set the threshold t as 0.5 for multi-label classification. We follow the metrics used in Qin et al. (2020) for intent accuracy and F1 score.

6 Main Results

6.1 Multi-intent detection

Table. 2 shows the normal multi-intent detection results on all five datasets. We can observe that LABAN outperforms the baselines substantially in the multi-intent detection especially in MixATIS and FSPS. It proves the usefulness of our fine-tuning BERT to capture more precise contextualized information for the downstream task. LABAN also considers the semantics in intent labels where the improvement enlarges when the number of intents increases, i.e. larger increase in MixATIS with 17 intents compared to MixSNIPS with only 7 intents. For datasets that do not have explicit conjunction words between the sentence like FSPS, MDC, SGD, we can observe a huge increase in accuracy in our model. Second, not only in multi-intent detection, in Table. 4, we can also see LABAN outperforms other baselines dealing with just one intent.

6.2 Zero-shot Multi-intent detection

To further justify our model's main contribution in zero-shot cases, we compare LABAN with several competitive baselines. As shown in Table. 3,

Dataset	MixATIS		MixSNIPS		FSPS		MDC		SGD	
Model	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc
Stack-Prop	0.790	0.719	0.976	0.946	0.911	0.723	0.877	0.780	0.919	0.891
Joint MID-SF	0.806	0.731	0.980	0.951	0.877	0.780	0.855	0.754	0.907	0.850
AGIF	0.812	0.758	0.985	0.961	0.914	0.749	**0.907**	0.741	0.924	0.761
LABAN	**0.958**†	**0.889**†	**0.985**	**0.963**	**0.948**†	**0.913**†	0.898	**0.814**†	**0.950**†	**0.928**†

Table 2: Normal multi-intent detection results on five datasets. We report accuracy (Acc) for all intents exact match and F1 scores based on individual intent calculation. † indicates the significant improvement of p-value < 0.05 compared to the previous state-of-the-art model AGIF.

Dataset	FSPS			MixATIS			MixSNIPS		
Model	F1-a	F1-s	F1-u	F1-a	F1-s	F1-u	F1-a	F1-s	F1-u
BERT-finetune	0.365	0.479	0.000	0.592	**0.836**	0.000	0.490	0.653	0.000
Zero-shot LSTM	0.341	**0.494**	0.029	0.533	0.728	0.055	0.475	0.546	0.264
CDSSM	0.496	0.440	0.394	0.592	0.827	0.060	0.591	**0.659**	0.432
Zero-shot BERT	0.517	0.461	0.373	0.463	0.576	0.162	0.472	0.464	0.370
CDSSM BERT	0.494	0.486	0.348	0.491	0.614	0.041	0.481	0.481	0.402
ALBERT-LA	0.391	0.425	0.228	0.595	0.739	0.362	0.567	0.574	0.466
TOD-BERT-LA	0.419	0.369	0.405	**0.702**	0.782	0.459	**0.642**	0.641	**0.559**†
BERT-LA (LABAN)	**0.544**	0.471	**0.451**†	0.696	0.808	**0.518**†	0.640	0.622	0.526

Table 3: Performance of the zero-shot multi-intent detection compared with several competitive baselines. Here we choose the train/test label ratio to be FSPS 17/24, MixATIS 14/17, MixSNIPS 5/7. F1-a, F1-s, F1-u are F1 scores evaluated on data with all/seen/unseen intent labels. † indicates the significant improvement of p-value $<$ 0.05 on F1-u results compared with CDSSM.

Model	ATIS	SNIPS
Stack-Propagation	0.969	0.980
Joint Mul ID-SF	0.954	0.972
AGIF	0.971	0.981
LABAN	**0.978**	**0.982**

Table 4: Single intent detection accuracy results on two single-intent datasets compared with baseline models.

BERT-finetune by simply enlarging the neurons for unseen intents is not capable of predicting any unseen intent utterances, causing 0.00 F1-u scores. Non-BERT approaches like Zero-shot LSTM and CDSSM using dot product or cosine similarity can show improved but limited unseen intent predictability. By leveraging pretraining power, zero-shot BERT can better associate unseen and seen intents with higher F1 score; while the performance of CDSSM BERT with more complex structures degrades with model overfitting. Finally, we discover that in all datasets (FSPS, MixATIS, MixSNIPS), with our label-aware attentive layer, three models (ALBERT-LA, TOD-BERT-LA, LABAN) with a strong pretrained power successfully outperform baselines in predicting unseen labels by associating their relations with input sequences, even if these intents are never seen in training phase.

We also observe that ALBERT has relatively inferior performance among BERT-based models, which possibly results from a light version of BERT and a different pretraining objective from the conversation-oriented version: TOD-BERT. To note, the original BERT model has slightly better F1 score for seen intents. It is reasonable since it avoids the error to predict utterances with unseen labels by searching over only the seen intents. However, without sacrificing much, models with the label-aware attentive layer could significantly boost the overall F1 scores in all three datasets.

Then we comprehensively evaluate LABAN's performance in zero/few-shot setting with different seen/unseen intent ratios in Figure. 2. We mainly have four discoveries. (1) LABAN can predict unseen intents around average half correctly. (2) When the number of seen intents decreases, F1 score reduces both for seen and unseen intent labels with model's poorer knowledge of seen intents. (3) In utterances with both seen and unseen intents, F1 score for seen intents is lower than utterances with only seen intents. The fewer seen intents are trained, the more inclined the model will predict the utterance as unseen intents frequently. (4) In the few-shot setting, with little data of unseen intents trained, both seen and unseen intent accuracy boost by a large margin especially in MixSNIPS. It indicates the fact that regardless of scarce training data with some unseen labels, LABAN could fully exploit the use of pretrained linguistic knowledge on label semantics to match the most relevant intents in current criteria.

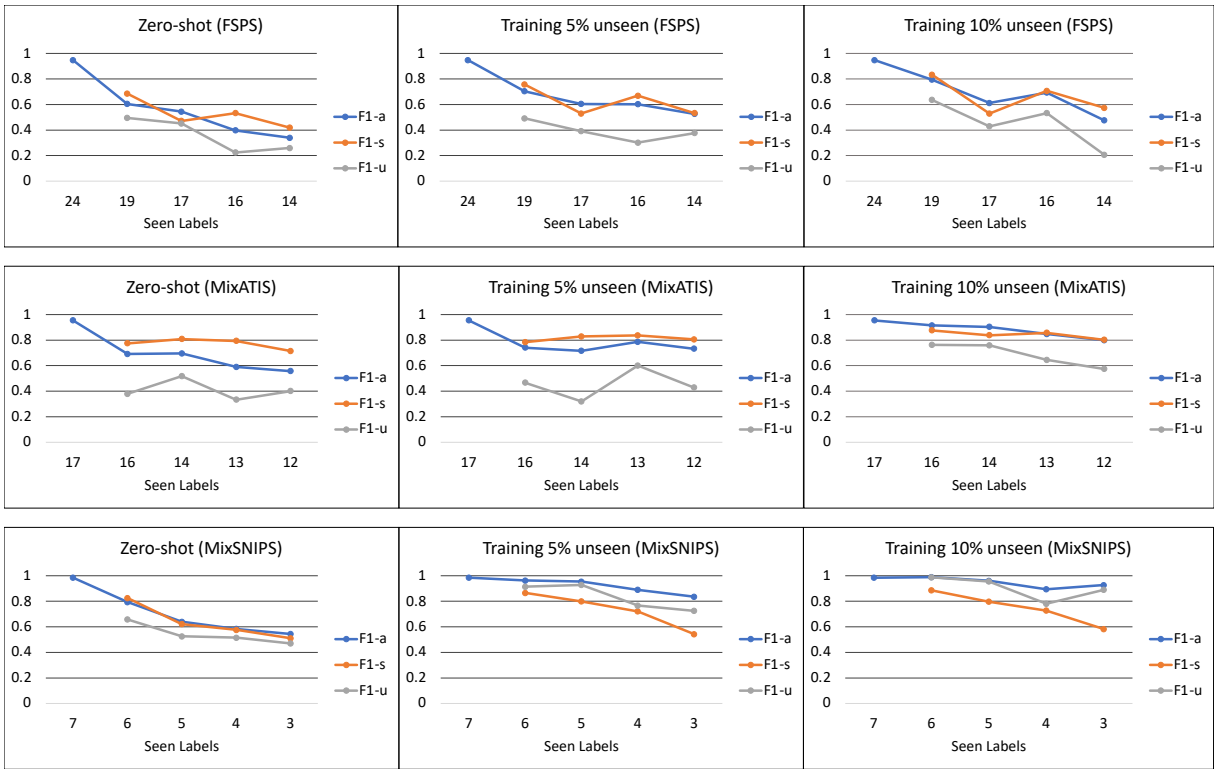

Figure 2: Zero-shot/Few-shot results of LABAN for FSPS, MixATIS, MixSNIPS datasets with varying Seen Labels, the number of seen labels during training. FSPS, MixATIS, MixSNIPS have total 24, 17, 7 intents. F1-a, F1-s, F1-u are F1 scores evaluated on data with all/seen/unseen intent labels.

Dataset	**MixATIS**		**MixSNIPS**		**FSPS**		**MDC**		**SGD**	
Model	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc
BERT-finetune	0.952	0.879	0.982	0.954	0.938	0.901	0.897	0.814	0.949	0.926
BERT-attn	**0.963**	**0.893**	0.984	0.961	0.942	0.903	0.897	**0.816**	0.950	0.927
LABAN	0.958	0.889	**0.985**	**0.963**	**0.948**	**0.913**	**0.898**	0.814	**0.950**	**0.928**

Table 5: Ablation analysis of different components in LABAN for normal multi-intent detection results on five datasets. We report accuracy (Acc) for all intents exact match and F1 scores based on individual intent calculation.

6.3 Ablation Analysis

To better understand the effectiveness of LABAN's components on multi-intent detection, we conduct the ablation analysis by reporting two different baseline variations of our model: BERT-finetune and BERT-attn. BERT-finetune refers to using the hidden state of [CLS] head from BERT without the extra label-aware layer; BERT-attn refers to adding a self-attentive layer to encode the sentence embeddings without the label-aware layer too. And finally, LABAN refers to our final model as the BERT with the self-attentive layer and adaptive label-aware attentive layer.

In experimental results shown in Table. 5, we can first observe that BERT with the additional self-attentive layer has increased performances on all five datasets, especially in MixATIS and FSPS. When the number of total intents increases, the self-attentive layer is beneficial in understanding each word importance to the overall intent prediction. After introducing the label-aware layer, we could see a further increase, especially in FSPS which contains the maximum number of intents (24). It does help LABAN to better match the utterance and different intent semantics, particularly in the case when intent options are more complicated. Although the increase seems subtle when the label sources are abundant, it can cause huge assistance of tackling unseen labels, without sacrificing much performance in normal cases.

6.4 Error Analysis

We demonstrate a few cases in Table. 6 to analyze some error cases of LABAN. For simplicity, we abbreviate each dataset as MixATIS: MA, MixSNIPS:

MI ID	Sentence	Predict labels	Real labels
MA1	At the charlotte airport, **how many** different types of aircraft are there for US air and St. Paul to Kansas city friday night.	atis_quantity (7)	atis_aircraft (4), atis_flight (12)
MS1	**Play** the album Journeyman.	play_music (0)	search_creative_work (2)
FS1	Is **traffic** always heavy **at this stretch** of highway?	get_location (3), get_info_traffic (6)	unsupported_navigation (2)
FS2	How's the **traffic** ahead?	*get_info_traffic* (6)	*get_info_traffic* (6), get_location (3)
ZS ID	**Sentence**	**Predict labels**	**Real labels**
MA2	Show me the **lowest priced fare** from Dallas to Baltimore.	*atis_airfare* (16), atis_flight (9), atis_cheapest (14)	*atis_airfare* (16)
MS2	Play **music** from 2015 and then I am giving this current novel 1 out of 6 stars.	*rate_book* (4), search_screening_event (5), book_restaurant (6)	*rate_book* (4), play_music (3)
FS3	I want to be **at my daughters** by 8am **what time** should I **leave?**	*get_location_home* (4), get_estimated_arrival (2), get_directions (16), update_directions (19)	*get_location_home* (4), get_estimated_departure (15)

Table 6: Example of multi-intent (MI) and zero-shot (ZS) prediction errors. Each example will have an id referring its dataset (MixATIS: MA, MixSNIPS: MS, FSPS: FS). *intent* indicates that it is the same both in prediction and real. And the number behind intents are the corresponding label id.

MS, and FSPS: FS in the table.

First, we found that some words in the utterances may obfuscate LABAN's prediction. For instance, in case MA1, LABAN may predict *'atis quantity'* based on the keyword 'how many' by comparing the sentence and label semantics. In case MS1, the 'play' keyword also induces the model to predict the intent *'play music'*, where it actually means to search and play an album list. In such sense, 'creative work' may be less relevant to 'album' for our model's sentence-label pairing.

For FSPS, we found that most errors occur when real labels are *'unsupported navigation'*, *'unsupported event'* or *'unsupported'* such as case FS1. This may be hard for the model without an external ontology to identify unsupported events (out-of-scope). Therefore, in most cases, the model will just identify *'get info traffic'* and *'get location'* as the closest intents. In FS2 case, the model fails to predict *'get location'* correctly. Without including contexts, it may be hard for the model to associate 'ahead' with *'get location'*.

Then, we show the errors in zero-shot setting. Here, the model only sees 12/17 intents in MixATIS, 3/7 intents in MixSNIPS and 14/24 intents in FSPS during training. We found two distinctive phenomena: (1) The model tends to predict more labels like in case MA2 if it is uncertain with unseen intents, resulting in lower precision. (2) We found that the model can predict seen intents well regardless of other existence of unseen intents in the same sentence. For unseen intent errors, the model tends to categorize them more into other unseen classes than seen classes, which indicates that the model has a basic knowledge of what seen intents should be. Mechanisms for explicit semantic pairing may be one of reasons and show ability of separating known and unknown classes confidently.

In case MA2, *'atis cheapest'* and *'atis airfare'* are not seen in training phase. However, the model is still capable of predicting *'atis airfare'* accurately. Moreover, 'lowest' keyword is matched with the predicted label *'atis cheapest'*, benefiting from our label-aware attentive layer. For case MS2, all of predicted and real labels are unseen during training. We found the model still accurately predicts *'rate book'* correctly based on keyword 'stars'. And the model predicts *'search screening event'* or *'search creative work'* instead *'play music'*, which actually happen frequently in other predictions. In FSPS like FS3 case, the model tends to predict lots of unseen intents without matching any of true intents. In FS3 case, it has only seen the intent *'get estimated arrival'* during training which makes it erroneously predicts the sentence to *'arrival'* rather than *'departure'*. The effect could be possibly alleviated by introducing external knowledge embeddings for keyword 'leave' related to 'departure', which human usually associates with.

6.5 Visualization

To better understand the classification results of LABAN, shown in Figure. 3, we perform TSNE visualization (van der Maaten and Hinton, 2008) on the projected embeddings $\hat{r}^u = \sum_{i=1}^{K} w_i r_i^l$ of each utterance onto the intent subspace $\mathcal{T}$. Here we also plot each intent embedding r_i^l with their intent numbers. We can observe that numerous clusters are formed with close semantic distances. And most of intent embeddings like id 0, 6, 9, 12 are close to their respective clusters. It indicates that LABAN successfully constructs an intent embedding space

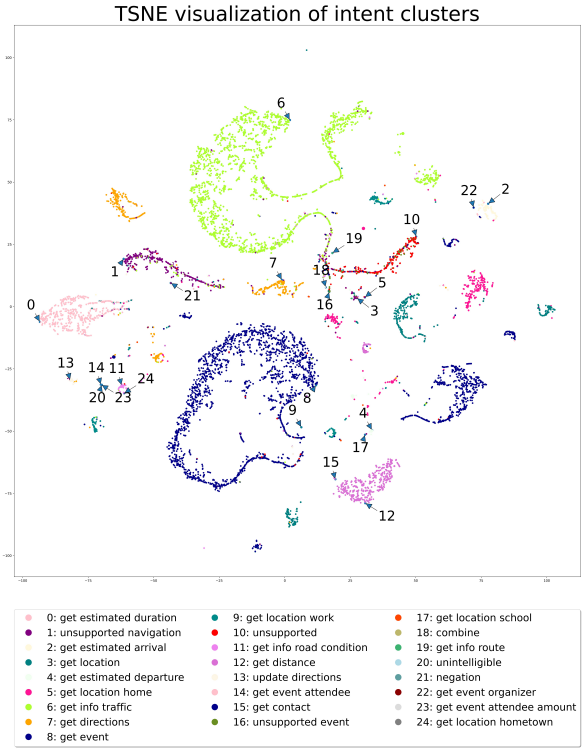

Figure 3: This figure shows the visualization of utterance embeddings with their intent labels (color) in FSPS test set. Each number i indicates an intent embedding r_i^l's location and its intent class.

that illustrates the semantic relation between each of intents and helps with classification of a projected utterance embedding. To note, since some of utterances have more than one intent, to simply the graph, we randomly pick one of intents in these utterances for visualization. Therefore, we can see some of clusters like id 8 actually have two dominant sub clusters. And some of utterances on the right sub cluster have other intents like id 3, 4, 12, 17. Hence, they may be semantically close to these intent embeddings (3, 4, 12, 17) on the graph.

7 Conclusion

In this paper, we propose the extension of fine-tuning BERT and label-aware semantic interactions into the multi-intent detection task in SLU. It successfully provides the solution to zero/few-shot setting where there are unseen labels in new utterances. By considering the label semantics, we can generate scores of how likely new utterances belong to these unseen intents. We compare the performance of our approach with previous methods and obtain significant improvements over baselines. It sheds the light that constructing a label semantic space could help the model to distinguish seen and unseen intents in utterances better. It provides the guidance in the work of improving SLU zero-shot multi-intent detection by considering dialogue contexts and external knowledge learning, or a more challenging task of detecting out-of-domain (OOD) detection where unseen intents are not available.

Acknowledgements

We are grateful for the insightful comments from the anonymous reviewers and the computing resources supported from Department of Electrical and Computer Engineering at Georgia Institute of Technology.

Ethical Consideration and Impact

The work aims to unfreeze the limitation of intent granularity defined in task-oriented dialogue training datasets, which is often ill-posed in the context of modeling precise and multiple intents in many previous works (Qin et al., 2019; Goo et al., 2018). Multi-intent detection could be applied to a wide range of applications in many industries where the scenario requires a broader understanding of user requests. For example, customer service automation often solicits clear intent identification at each utterance for flexible answer policy, where identifying single intents may increase redundant and ambiguous dialogue turns. Second, zero-shot work has long been studied to unfreeze the limitation of deep learning models requesting large amount of data. It could be applied to multiple domains where intent labels are significantly lacking and may cause time-consuming labeling. By transferring the knowledge from existing labels, the model shall be more robust in dealing with unseen labels as humans have approached new things, which will be very beneficial in dialogue system design where many of data are unlabeled.

In ethical aspect, naturalness of dialog structure heavily defines the scope of intent detection and usually changes during the dialog state transition. How to capture adequate intents from user is somehow critical in SLU and the following tasks like dialog state tracking. Wrong interpretation of intents may offend users and cause unsatisfactory answers. And we should also avoid predicting sensitive labels regarding user privacy. In such sense, we mainly test our model in all public released datasets which have been widely justified as unbiased in multiple domains and are not sensitive in revealing specific user information.

Overall, we see great opportunities for research applying LABAN to investigate interactions between utterance and their latent intents. It gives good intuition how the model understands the underlying human acts and improves the transparency in decision-critical applications. To mitigate the risks associated with our model, we aim to anonymize user sensitive information in training data and focus on extracting domain-agnostic knowledge for better generalization and interpretability.

References

Leonard Abbeduto. 1983. Linguistic communication and speech acts. kent bach, robert m. harnish. cambridge: M.i.t. press, 1979, pp. xvii 327. *Applied Psycholinguistics*, 4(4):397–407.

Zeynep Akata, Florent Perronnin, Zaid Harchaoui, and Cordelia Schmid. 2016. Label-embedding for image classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(7):1425–1438.

Ziad Al-Halah, Makarand Tapaswi, and Rainer Stiefelhagen. 2016. Recovering the missing link: Predicting class-attribute associations for unsupervised zero-shot learning. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 5975–5984. IEEE Computer Society.

Yun-Nung Chen, Dilek Hakkani-Tür, and Xiaodong He. 2016. Zero-shot learning of intent embeddings for expansion by convolutional deep structured semantic models. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2016, Shanghai, China, March 20-25, 2016*, pages 6045–6049. IEEE.

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.

Guido E. del Pino and Hector Galaz. 1995. Statistical applications of the inverse gram matrix: A revisitation. *Brazilian Journal of Probability and Statistics*, 9(2):177–196.

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.

Haihong E, Peiqing Niu, Zhongfu Chen, and Meina Song. 2019. A novel bi-directional interrelated model for joint intent detection and slot filling. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5467–5471, Florence, Italy. Association for Computational Linguistics.

Andrea Frome, Gregory S. Corrado, Jonathon Shlens, Samy Bengio, Jeffrey Dean, Marc'Aurelio Ranzato, and Tomás Mikolov. 2013. Devise: A deep visual-semantic embedding model. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, pages 2121–2129.

Rashmi Gangadharaiah and Balakrishnan Narayanaswamy. 2019. Joint multiple intent detection and slot labeling for goal-oriented dialog. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 564–569, Minneapolis, Minnesota. Association for Computational Linguistics.

Jianfeng Gao, Michel Galley, and Lihong Li. 2018. Neural approaches to conversational AI. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR 2018, Ann Arbor, MI, USA, July 08-12, 2018*, pages 1371–1374. ACM.

Chih-Wen Goo, Guang Gao, Yun-Kai Hsu, Chih-Li Huo, Tsung-Chieh Chen, Keng-Wei Hsu, and Yun-Nung Chen. 2018. Slot-gated modeling for joint slot filling and intent prediction. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 753–757, New Orleans, Louisiana. Association for Computational Linguistics.

Sonal Gupta, Rushin Shah, Mrinal Mohit, Anuj Kumar, and Mike Lewis. 2018. Semantic parsing for task oriented dialog using hierarchical representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2787–2792, Brussels, Belgium. Association for Computational Linguistics.

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*.

Anjishnu Kumar, Pavankumar Reddy Muddireddy, Markus Dreyer, and Björn Hoffmeister. 2017. Zero-shot learning across heterogeneous overlapping domains. In *INTERSPEECH*.

Christoph H Lampert. 2014. Attribute-based classification for zero-shot visual object categorization.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A lite BERT for self-supervised learning of language representations. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Stefan Larson, Anish Mahendran, Joseph J. Peper, Christopher Clarke, Andrew Lee, Parker Hill, Jonathan K. Kummerfeld, Kevin Leach, Michael A. Laurenzano, Lingjia Tang, and Jason Mars. 2019. An evaluation dataset for intent classification and out-of-scope prediction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1311–1316, Hong Kong, China. Association for Computational Linguistics.

Changliang Li, Liang Li, and Ji Qi. 2018a. A self-attentive model with gate mechanism for spoken language understanding. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3824–3833, Brussels, Belgium. Association for Computational Linguistics.

Xiujun Li, Sarah Panda, Jingjing Liu, and Jianfeng Gao. 2018b. Microsoft dialogue challenge: Building end-to-end task-completion dialogue systems. *arXiv preprint arXiv:1807.11125*.

Zhouhan Lin, Minwei Feng, Cícero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. 2017. A structured self-attentive sentence embedding. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Han Liu, Xiaotong Zhang, Lu Fan, Xuandi Fu, Qimai Li, Xiao-Ming Wu, and Albert Y.S. Lam. 2019a. Reconstructing capsule networks for zero-shot intent classification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4799–4809, Hong Kong, China. Association for Computational Linguistics.

Ting Liu, Xiao DING, Yue QIAN, and Yiheng CHEN. 2017. Identification method of user's travel consumption intention in chatting robot. *SCIENTIA SINICA Informationis*, 47:997.

Yijin Liu, Fandong Meng, Jinchao Zhang, Jie Zhou, Yufeng Chen, and Jinan Xu. 2019b. CM-net: A novel collaborative memory network for spoken language understanding. 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 1051–1060, Hong Kong, China. Association for Computational Linguistics.

Mohammad Norouzi, Tomás Mikolov, Samy Bengio, Yoram Singer, Jonathon Shlens, Andrea Frome, Greg Corrado, and Jeffrey Dean. 2014. Zero-shot learning by convex combination of semantic embeddings. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*.

Mark Palatucci, Dean Pomerleau, Geoffrey E. Hinton, and Tom M. Mitchell. 2009. Zero-shot learning with semantic output codes. In *Advances in Neural Information Processing Systems 22: 23rd Annual Conference on Neural Information Processing Systems 2009. Proceedings of a meeting held 7-10 December 2009, Vancouver, British Columbia, Canada*, pages 1410–1418. Curran Associates, Inc.

Yue Qian. 2017. Research on the identification method of users' travel consumption intent in chat robot.

Libo Qin, Wanxiang Che, Yangming Li, Haoyang Wen, and Ting Liu. 2019. A stack-propagation framework with token-level intent detection for spoken language understanding. 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 2078–2087, Hong Kong, China. Association for Computational Linguistics.

Libo Qin, Xiao Xu, Wanxiang Che, and Ting Liu. 2020. AGIF: An adaptive graph-interactive framework for joint multiple intent detection and slot filling. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1807–1816, Online. Association for Computational Linguistics.

Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, Raghav Gupta, and Pranav Khaitan. 2019. Towards scalable multi-domain conversational agents: The schema-guided dialogue dataset. *arXiv preprint arXiv:1909.05855*.

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.

B. Rychalska, H. Glabska, and A. Wroblewska. 2018. Multi-intent hierarchical natural language understanding for chatbots. In *2018 Fifth International Conference on Social Networks Analysis, Management and Security (SNAMS)*, pages 256–259.

Qingyi Si, Yuanxin Liu, Peng Fu, Zheng Lin, Jiangnan Li, and Weiping Wang. 2021. Learning class-transductive intent representations for zero-shot intent detection. In *IJCAI*.

Jake Snell, Kevin Swersky, and Richard S. Zemel. 2017. Prototypical networks for few-shot learning.

In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 4077–4087.

Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2020. How to fine-tune bert for text classification?

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

Chien-Sheng Wu, Steven C.H. Hoi, Richard Socher, and Caiming Xiong. 2020. TOD-BERT: Pre-trained natural language understanding for task-oriented dialogue. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 917–929, Online. Association for Computational Linguistics.

Ting-Wei Wu, Ruolin Su, and Biing-Hwang Juang. 2021. A Context-Aware Hierarchical BERT Fusion Network for Multi-Turn Dialog Act Detection. In *Proc. Interspeech 2021*, pages 1239–1243.

Congying Xia, Caiming Xiong, Philip Yu, and Richard Socher. 2020. Composed variational natural language generation for few-shot intents. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3379–3388, Online. Association for Computational Linguistics.

Congying Xia, Chenwei Zhang, Xiaohui Yan, Yi Chang, and Philip Yu. 2018. Zero-shot user intent detection via capsule neural networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3090–3099, Brussels, Belgium. Association for Computational Linguistics.

Yongqin Xian, Bernt Schiele, and Zeynep Akata. 2017. Zero-shot learning - the good, the bad and the ugly. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 3077–3086. IEEE Computer Society.

Ziming Zhang and Venkatesh Saligrama. 2015. Zero-shot learning via semantic similarity embedding. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, pages 4166–4174. IEEE Computer Society.

A Appendix

A.1 Linear Approximation in a Hilbert Space

Let $\mathcal{S}$ be a Hilbert space with inner product $\langle \cdot, \cdot \rangle$ and induced norm $|| \cdot ||$, and let $\mathcal{T}$ be a subspace of $\mathcal{S}$. Given a vector $z \in \mathcal{S}$, we would like to find the closest point $\hat{z} \in \mathcal{T}$ to z. Namely, we would like to solve the following optimization program:

$$\min_{x \in \mathcal{T}} ||z - x|| \tag{8}$$

Given an arbitrary $z \in \mathcal{S}$, we know there exists exactly one point $\hat{z} \in \mathcal{T}$ that obeys

$$z - \hat{z} \perp \mathcal{T} \tag{9}$$

meaning $\langle z - \hat{z}, y \rangle = 0$ for all $y \in \mathcal{T}$ and this point $\hat{z}$ is the unique minimizer of Equation 8. We can further construct $\hat{z}$ as the following:

$$\hat{z} = \sum_{k=1}^{N} \beta_k v_k \tag{10}$$

where N is the dimension of $\mathcal{T}$ and $v_1, ..., v_N$ is a basis for $\mathcal{T}$. Then we can transform our problem as finding coefficients $\beta_1, ..., \beta_N \in \mathbb{C}$.

From Equation 9, we know $\langle z - \hat{z}, v_n \rangle = 0$ for $n = 1, ..., N$. This means by plugging Equation 10, β_n must obey $\langle z - \sum_{k=1}^{N} \beta_k v_k, v_n \rangle = 0$ for $n = 1, ..., N$. We can then obtain the following equation:

$$\langle z, v_n \rangle = \sum_{k=1}^{N} \beta_k \langle v_k, v_n \rangle \tag{11}$$

Since z and the $\{v_n\}$ are given, we know both the $\langle z, v_n \rangle$ and $\langle v_k, v_n \rangle$. We can write down the matrix form:

$$\mathbf{G}\beta = \mathbf{b} \tag{12}$$

where $\beta \in \mathbb{C}^N$, $b_n = \langle z, v_n \rangle$ and $G_{k,n} = \langle v_n, v_k \rangle$. Or in the complete form:

$$\mathbf{G} = \begin{bmatrix} \langle v_1, v_1 \rangle & \cdots & \langle v_N, v_1 \rangle \\ \vdots & \ddots & \vdots \\ \langle v_1, v_N \rangle & \cdots & \langle v_N, v_N \rangle \end{bmatrix} \tag{13}$$

$$\mathbf{b} = \begin{bmatrix} \langle z, v_1 \rangle \\ \vdots \\ \langle z, v_N \rangle \end{bmatrix} \tag{14}$$

We can then solve the problem by finding $\beta = \mathbf{G}^{-1}\mathbf{b}$ where $\mathbf{G}$ is guaranteed invertible since $\{v_n\}$ is linear independent.

A.2 Dataset

Here are some more detailed descriptions about datasets we used:

MixATIS (Qin et al., 2020; Hemphill et al., 1990) ATIS (Airline Travel Information System) dataset is a standard benchmark dataset in the airline domain widely used as an intent classification. MixATIS is further synthesized based on ATIS by concatenating single utterances only with the conjunction word 'and'.

MixSNIPS (Qin et al., 2020; Coucke et al., 2018) MixSNIPS dataset is collected from the SNIPS personal voice assistant and has the ratio of sentences with 1-3 intents at [0.3, 0.5, 0.2]. It also concatenates SNIPS utterances with the conjunction word 'and'.

FSPS (Gupta et al., 2018) Facebook Semantic Parsing System (FSPS) dataset is a large dataset of 44k requests annotated with a hierarchical semantic representation for task oriented dialog systems. Intents are prefixed with 'IN:' an slots with 'SL:'. Each utterance may contain one or more embedded intents and slots.

MDC (Li et al., 2018b) Microsoft dialogue challenge dataset (MDC) is a well-annotated dataset for three task-completion domains: movie-ticket booking, restaurant reservation and taxi ordering. It was first released for SLT 2018 special session and contains information of dialogue acts and slots for each utterance.

SGD (Rastogi et al., 2019) Schema-Guided Dialogue dataset (SGD) is a large dialogue dataset with over 20k annotated multi-domain, task-oriented conversations between a human and a virtual assistant. These conversations involve interactions with services and APIs spanning 20 domains. It could be used for intent prediction, slot filling, dialogue state tracking, policy imitation learning or language generation.