Balanced Joint Adversarial Training for Robust Intent Detection and Slot Filling

Xu Cao[1,3], Deyi Xiong[2], Chongyang Shi[1*], Chao Wang[3], Yao Meng[3], Changjian Hu[3]
[1] School of Computer Science, Beijing Institute of Technology, Beijing , China
[2] College of Intelligence and Computing, Tianjin University, Tianjin, China
[3] Lenovo Research AI Lab, Beijing, China
bitcaoxu@gmail.com
dyxiong@tju.edu.cn
cy_shi@bit.edu.cn
{wangchao31, mengyao1, hucj1}@lenovo.com

Abstract

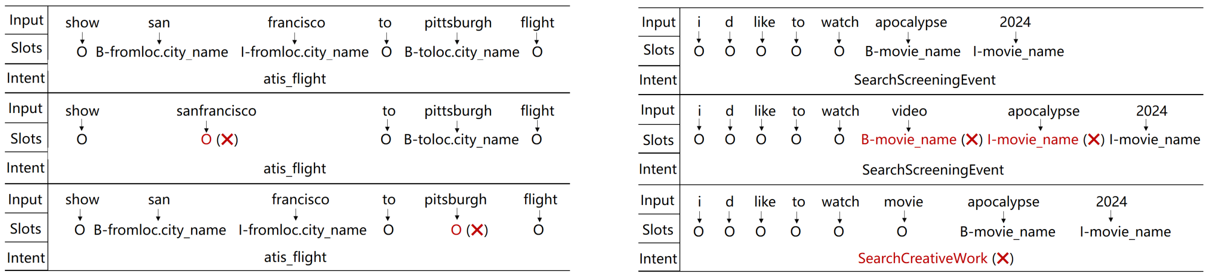
Joint intent detection and slot filling has recently achieved tremendous success in advancing the performance of utterance understanding. However, many joint models still suffer from the robustness problem, especially on noisy inputs or rare/unseen events. To address this issue, we propose a Joint Adversarial Training (JAT) model to improve the robustness of joint intent detection and slot filling, which consists of two parts: (1) automatically generating joint adversarial examples to attack the joint model, and (2) training the model to defend against the joint adversarial examples so as to robustify the model on small perturbations. As the generated joint adversarial examples have different impacts on the intent detection and slot filling loss, we further propose a Balanced Joint Adversarial Training (BJAT) model that applies a balance factor as a regularization term to the final loss function, which yields a stable training procedure. Extensive experiments and analyses on the lightweight models show that our proposed methods achieve significantly higher scores and substantially improve the robustness of both intent detection and slot filling. In addition, the combination of our BJAT with BERT-large achieves state-of-the-art results on two datasets.

1 Introduction

Intent detection and slot filling are two critical components in dialogue systems. Although these two tasks are normally considered as parallel tasks, they may inherently correlate with each other as one is helpful in defining the other. Some existing works (Liu and Lane, 2016; Vu, 2016; Zhang and Wang, 2016) indicate that sharing parameters in an encoder module to simultaneously detect the intent and fill the slots of an utterance can utilize such correlation between the two tasks. The joint modeling of them has therefore achieved tremendous success (Goo et al., 2018; Li et al., 2018; E et al., 2019) and further improved the performance of spoken language understanding (SLU) systems.

However, recent outstanding joint models still show insufficient robustness. For example, some small changes of inputs can mislead the models to give wrong prediction. As shown in Figure 1 (a), misspelled words "SanFranciso" (missing a blank) and "pitsburgh" (missing "t") are easily to fool the neural models to produce wrong slot predictions. One might think that such wrong predictions are related to the input word embeddings and using character embeddings can avoid these prediction errors. However, we find in our preliminary experiments that character embeddings cannot fully eliminate such errors as it is not easy to reconstruct the meaning of corresponding slot words based on character embeddings. In Figure 1 (b), word-level perturbations such as inserted word "video" or substituted word "movie" can lead to incorrect prediction in either slot filling or intent detection. In real world, spelling errors are easily found and word variances are pervasive as utterances are from different people with different text styles. This is challenging for many joint models of intent detection and slot filling.

[*] Corresponding author.


Input | show san francisco to pittsburgh flight
Slots | O B-fromloc.city_name I-fromloc.city_name O B-toloc.city_name O
Intent | atis_flight
Input | show sanfrancisco to pittsburgh flight
Slots | O O (✗) O B-toloc.city_name O
Intent | atis_flight
Input | show san francisco to pitsburgh flight
Slots | O B-fromloc.city_name I-fromloc.city_name O O (✗) O
Intent | atis_flight
Input | i d like to watch apocalypse 2024
Slots | O O O O O B-movie_name I-movie_name
Intent | SearchScreeningEvent
Input | i d like to watch video apocalypse 2024
Slots | O O O O O B-movie_name (✗) I-movie_name (✗) I-movie_name
Intent | SearchScreeningEvent
Input | i d like to watch movie apocalypse 2024
Slots | O O O O O O B-movie_name I-movie_name
Intent | SearchCreativeWork (✗)

(a) Spelling errors (b) Word insertion and substitution

Figure 1: Examples demonstrating that small perturbations (e.g., spelling errors in (a) and word insertion & substitution in (b)) in utterances result in wrong predictions in both intent detection and slot filling.

In this paper, we resort to adversarial training for solving this problem. Adversarial training provides a way to inject perturbed inputs (adversarial examples) into training data to regularize models for robustness. Specifically, we propose a Joint Adversarial Training (JAT) model to improve the robustness of the joint intent detection (ID) and slot filling (SF). In JAT, we first add tiny noises to the original training data on word (character) embeddings. The perturbed data created in this way are referred to as joint adversarial examples. We then train the joint ID and SF model to defend against these adversarial examples so as to make the model robust to small perturbations. By adding perturbations to word/character embeddings, we enable the joint model to learn from adversarial examples previously unseen to deal with variances and perturbations in utterances. In doing so, we hope that small changes in utterances cannot fool the new joint model trained with adversarial examples. Furthermore, as the loss function of the joint model is the combination of the ID loss and SF loss, the two losses are sensitive to adversarial examples in different degrees, which may cause the joint training to be unstable. To overcome this, we further propose a Balanced Joint Adversarial Training (BJAT) model. Particularly, we add a balanced factor to the joint loss function as a regularization term to obtain the impact equilibrium that adversarial examples impose on the ID and SF loss. BJAT shows a more stable training process and achieves better performance than JAT.

To summarize, the contributions of this work are threefold as follows:

- We propose a joint adversarial training model to the joint intent detection and slot filling, which aims to improve the robustness of the joint model in handling small perturbations in utterances. To the best of our knowledge, this is the first attempt to adapt adversarial training to the joint ID and SF.
- We further propose a balanced joint adversarial training model that helps stabilize the joint adversarial training procedure.
- We conduct experiments on the widely-used datatsets to validate the effectiveness the proposed JAT and BJAT. Experiment results and analyses demonstrate that the proposed methods can substantially improve the robustness of the joint ID and SF on noisy inputs, rare words and intent types. And the BJAT built upon BERT-large achieves SOTA performance in the datasets.

2 Related Work

2.1 Joint Training

The joint prediction of intent and slot labels has achieved higher performance due to information sharing between the two tasks than the parallel modeling of them. (Liu and Lane, 2016) explore the sequence-to-sequence model (Guo et al., 2014) for the joint task and achieve decent results. Many works follow this idea and make further research. Models based on gating mechanism (Goo et al., 2018; Li et al., 2018; E et al., 2019) have been proposed for dynamically modeling the relationship between slot filling and intent detection. (Wang et al., 2018) propose a dual model which contains two correlated bidirectional LSTMs with cross-impact features. (Chen et al., 2019) utilize large pre-trained language models to learn better representations. Their work directly constructs the relationship between intent and slots from feature interaction. In this work, we also use the joint training setting, but focus on the robustness problem of the joint model in dealing with noises and variances.

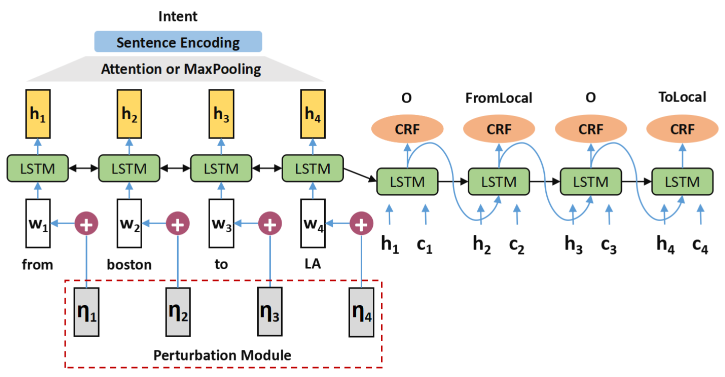


Figure 2: Illustration of the proposed model for joint intent detection and slot filling.

2.2 Adversarial Training

Adversarial examples (Goodfellow et al., 2015) have been originally introduced in the context of image classification to fool neural models with unperceivable perturbations on images. After that, adversarial training (AT) that uses automatically generated adversarial examples to strengthen the robustness of neural models has become a popular research topic. (Miyato et al., 2017) adapt AT to text classification by adding perturbations on word embeddings. Following that, many researchers (Wu et al., 2017; Yasunaga et al., 2018; Bekoulis et al., 2018; Ju et al., 2019) have attempted to explore the potential benefits that AT can bring to NLP tasks. (Yasunaga et al., 2018) use AT on POS tagging and find that AT not only improves overall tagging accuracy, but also helps the model to learn cleaner word representations. The same benefits have also emerged in the text classification task with AT (Miyato et al., 2017). Joint intent detection and slot filling is a multi-task learning problem. It is more challenging to develop an AT algorithm that naturally fits in multi-task learning. Different from previous works on adapting AT to single NLP tasks, we propose a balanced AT technique for joint models, which achieve significant improvements over the strong baselines.

3 Methodology

In this section, we elaborate the proposed joint adversarial training model for the joint ID and SF and its balanced variant. We first introduce the joint model that we use as our baseline for intent detection and slot filling.

3.1 The Baseline Joint Model

Following previous works, we also use a combination of LSTM encoder-decoder with a stacked CRF to build the joint model, as shown in Figure 2 without the perturbation module. Particularly, given a sentence $s = [w_1, w_2, ..., w_n]$ (w_i is the combination of word and character embeddings), the bidirectional LSTM encoder encodes the sentence into a sequence of hidden representations $H = [h_1, h_2, .., h_n]$. The last hidden state of H is input to the LSTM decoder that learns context vectors $C = [c_1, c_2, ..., c_n]$ for decoder states with an attention network (Bengio and LeCun, 2015). The outputs of the LSTM decoder are then feed into the stacked CRF layer that predicts slot labels.Unlike previous works, we use two different methods to learn the representation of the input sentence for intent detection: an attention-based method building an attention network over H to have the overall representation of the sentence and a max-pooling over each hidden state dimension across all h_is to have the final sentence representation. We refer to the joint model with attention+CRF as AC and the joint model with max-pooling+CRF as MC.

As a multi-task learning task, the loss function of the joint model comprises two parts: intent detection loss and slot filling loss.

Intent Detection Loss Given an input sentence $s = [w_1, w_2, ..., w_n]$ (w_i is the combination of word and character embeddings), we estimate the conditional probability of the true intent as: $p(y|s; \theta_1)$ where θ_1 represents all parameters from the intent detection module.

During training, we minimize the negative log likelihood, which is the intent detection loss:

$$IL(\theta_1; s, y_{intent}) = -logp(y_{intent}|s; \theta_1) \tag{1}$$

where y_{intent} is the true intent for the given sentence.

Slot Filling Loss We pass a list of features from the forward LSTM decoder to the stacked first-order chain CRF. The final loss function for training this module is defined as follows:

$$SL(\theta_2; s, y_{slot}) = -logp(y_{slot}|s; \theta_2) \tag{2}$$

where θ_2 represents all parameters from the slot filling module (including parameters of the shared encoder and those of the decoder together with CRF), while y_{slot} denotes the ground-truth slot sequence for the given sentence.

Joint Loss The joint loss is defined as the sum of IL and SL (Liu and Lane, 2016):

$$JL(\theta; s, y) = IL + SL \tag{3}$$

where y denotes the combined representation of y_{intent} and y_{slot} and $\theta = \theta_1 \cup \theta_2$.

3.2 Balanced Joint Adversarial Training

Based on the baseline joint model, we propose the balanced joint adversarial training model as illustrated in Figure 2. In order to build this new model, we need to generate adversarial examples and train the model on these examples.

Generating Joint Adversarial Examples Adversarial training leverages continuous perturbations of inputs to robustify neural models. We therefore define perturbations at the level of dense word (with character) embeddings in a way similar to (Miyato et al., 2017)'s work. For a given sentence s, we generate perturbations bounded by a small norm ϵ, which maximize the loss function of the joint model:

$$\eta = \arg\max_{\eta' : \|\eta\| \leq \epsilon} JL\left(\hat{\theta}; s + \eta', y\right) \tag{4}$$

where $\hat{\theta}$ is a fixed copy of the current value of model parameters θ with no gradient, while y denotes the combination of target intent and slot labels. Unfortunately, the exact maximization with respect to η is intractable for neural networks (Miyato et al., 2017; Liu et al., 2017). In order to solve this, (Goodfellow et al., 2015) propose a first-order approximation method called Fast Gradient Method, which can obtain an approximate worst-case perturbation of norm ϵ, by a single gradient computation:

$$\eta = \epsilon g / \|g\|_2, \text{ where } g = \nabla_s JL\left(\hat{\theta}; s, y\right) \tag{5}$$

Note that η is an approximation of the worst-case perturbation in the direction that significantly increases the joint loss JL. We add η to the embedding (word and character) layer to generate a joint adversarial example s_{jadv}, which can be expressed as follows:

$$s_{jadv} = s + \eta \tag{6}$$

Training against Joint Adversarial Examples When we train the joint model on these generated joint adversarial examples, the new loss functions for intent detection and slot filling can be defined as:

$$IL_{adv} = IL(\theta; s_{jadv}, y_{intent}) \tag{7}$$

$$SL_{adv} = SL(\theta; s_{jadv}, y_{slot}) \tag{8}$$

The joint adversarial loss is the sum of these two losses as:

$$JL_{adv} = IL_{adv} + SL_{adv} \tag{9}$$

During the training process, we generate joint adversarial examples against the current model, and minimize the loss on both the clean examples and joint adversarial examples. Therefore, the loss function for the joint adversarial training model is defined as:

$$L = JL + JL_{adv} \tag{10}$$

If the norms of word embeddings are not bounded, the model can simply make perturbations insignificant by learning to magnify the norms of word embeddings. To prevent this problem, the word embeddings in the JAT model are normalized so that each entry has a mean of 0 and a variance of 1. To ensure a fair comparison, we also normalize input embeddings in our baseline model in the same way.

Adding Balance Factor As described above, the joint adversarial examples are generated to maximize the joint loss of intent detection and slot filling. The two parts IL (intent loss) and SL (slot filling loss) of the joint loss often have a large difference in their values and sometimes even in an order of magnitude. This may make the loss of intent detection and slot filling suffer from very unbalanced impacts from the joint adversarial examples. In other words, the learned perturbations may bring a big increase to the loss of the slot filling task but a very tiny change to the loss of the intent detection task. This unbalanced problem can cause oscillations in the learning of the JAT model, which can be seen at the last subsection of Analysis section in Figure 5.

To address this problem, we further propose a balanced joint adversarial training model, which introduces a balance factor α to balance the impacts on the training of the two tasks imposed by joint adversarial examples:

$$\alpha = \left| \frac{IL_{adv} - IL}{IL + \beta} - \frac{SL_{adv} - SL}{SL + \beta} \right| \tag{11}$$

There are two parts in the balance factor, each of which measures how much the adversarial training affects the corresponding task. β is a smoothing constant parameter to prevent division by zero.

In BJAT, we add α to the final loss, which is to be minimized during training:

$$BL = JL + JL_{adv} + \alpha \tag{12}$$

The balance factor α can be also considered as a regularization term. When α is optimized to the minimum (i.e., zero) along with the loss L, the degree of impact from perturbations to JL and IL will be same.

4 Experiments

We conducted intensive experiments on two benchmark datasets to validate the effectiveness of the proposed two adversarial training models for joint intent detection and slot filling.

4.1 Datasets

We used two datasets: The ATIS dataset (Tür et al., 2010) is widely used in SLU research. This dataset contains audio recordings of people making flight reservations. The training set consists of 4478 sentences, while the test set 893 sentences, with a total of 18 intent classes and 127 slot labels. We used another 500 sentences as the development set. The SNIPS dataset is collected from the Snips (Coucke et al., 2018) personal voice assistant. The training set contains 13,084 sentences. Both the test and development set contain 700 sentences. There are 72 slot labels and 7 intent types in the SNIPS dataset.

4.2 Training & Evaluation Details

Baselines We used two lightweight models (AC & MC) and BERTs (DistilBERT, BERT-base and BERT-large) as our baseline models.

Model Settings In AC & MC, the word and character embeddings were initialized according a random uniform from -0.1 to 0.1. Based on the amount of data, the word embedding dimension was set to 64 for ATIS and 128 for SNIPS, the dimension of character embedding was 30 for ATIS and 50 for SNIPS. Similarly, we set the dimension size of hidden LSTM states to be 64 for ATIS and 128 for SNIPS. And we set the norm of a perturbation ϵ to 0.08 for all the experiments. In BJAT, the smoothing constant β of the balanced factor was set as 1e-8. In addition, we used distilbert-base-uncased, bert-base-uncased and bert-large-uncased. In BERTs, the norm of a perturbation ϵ was set to 0.008 for all the experiments. And the perturbations are added to the final layer.

Optimization In AC & MC, the model parameters were trained by Adam optimizer with batch size 16 and learning rate 0.001. We also used a gradient clipping of 5.0. The number of epochs was set to 80 on both ATIS and SNIPS. In BERTs, the basic settings were the same as (Chen et al., 2019).

Method	ATIS			SNIPS		
	ICA	SF1	SOA	ICA	SF1	SOA
AC	0.951	0.939	0.805	0.966	0.897	0.758
AC+JAT	0.956	0.946	0.829	0.963	0.917	0.801
AC+BJAT	**0.967**	**0.953**	**0.846**	0.975	**0.937**	**0.841**
MC	0.955	0.934	0.796	0.971	0.902	0.777
MC+JAT	0.960	0.947	0.830	0.974	0.921	0.812
MC+BJAT	0.965	0.951	0.842	**0.977**	0.931	0.833

Table 1: Comparison with JAT, BJAT and the baseline models on the ATIS and SNIPS datasets.

Method	ATIS			SNIPS		
	ICA	SF1	SOA	ICA	SF1	SOA
Seq2Seq (Liu and Lane, 2016)	0.911	0.942	0.789	0.967	0.878	0.741
Self-attention (Li et al., 2018)	0.957	0.938	0.822	0.967	0.899	0.793
Slot-Gated (Goo et al., 2018)	0.936	0.948	0.822	0.970	0.888	0.755
SF-ID (E et al., 2019)	0.977	0.958	0.869	0.974	0.922	0.805
DistilBERT (Sanh et al., 2019)	0.976	0.955	0.877	0.985	0.964	0.918
BERT-base (Chen et al., 2019)	0.979	0.960	0.886	0.984	0.967	0.926
BERT-large (Devlin et al., 2019)	0.978	0.960	0.882	**0.991**	0.968	0.930
DistilBERT+BJAT	0.978	**0.961**	0.884	**0.991**	0.962	0.917
BERT-base+BJAT	0.981	0.960	0.887	**0.991**	0.969	0.930
BERT-large+BJAT	**0.982**	0.959	**0.888**	**0.991**	**0.973**	**0.932**

Table 2: Comparison to BERT and other previous state-of-the-art models.

Evaluation Metrics We used the following three metrics for evaluation:

- Intent Classification Accuracy **(ICA)**: this metric was used for the evaluation of intent detection.
- Slot F1 **(SF1)**: a standard F1 score was used to evaluate slot filling.
- Sentence Overall Accuracy **(SOA)**: this was used as a metric for the joint evaluation of the two tasks, where a test sentence is considered correct only if intent and all slot labels of the sentence are correctly predicted.

4.3 Results

Results with Lightweight Models Table 1 shows the results. We find that both JAT and BJAT achieve substantial improvements over baseline models on all metrics, and BJAT performs better than JAT. We highlight scores with significant improvements in bold. On the ATIS dataset, AC+BJAT obtains +3.3% improvement on ICA, +1.5% improvement on SF1, and +5.1% improvement on SOA over the baseline AC. Similarly, significant improvements can be also found on the SNIPS dataset. On this dataset, MC+BJAT gains +0.6% improvement on ICA, while AC+BJAT obtains +4.5% improvement on SF1 and +10.9% improvement on SOA over the corresponding baseline. It can be seen that both JAT and BJAT can achieve larger improvements on slot filling than intent detection. This may be because slot labels are more sensitive to changes of word embeddings than intents.

Results with Pre-trained Language Models We further applied BJAT on the pre-trained language model BERT and compared with previous state-of-the-art models. Results in Table 2 demonstrate that the combination of our BJAT with BERT models achieves new state-of-the-art results on the two datasets.

Results on Noisy Data We have shown improvements gained by our models on the standard clean benchmark datasets. Here we further evaluate the robustness of JAT and BJAT models on data with artificially generated noise. We experimented with three types of perturbations on the two test sets (please see the definitions for slot/contextual word in the next section.):

- **OP1** We randomly choose 1 slot word from each sentence and add, change, or delete some characters of it.
- **OP2** We randomly choose 1 contextual word of slot words from each sentence and add, change, or delete some letters of it.
- **OP3** This is an combination of OP1 and OP2, where we randomly change 1 slot word and 1 contextual word for each sentence.

Table 3 show the ICA and SF1 results on noisy input. We also calculated the average performance drop of each model on OP1-3 against the clean data. It is clearly seen that both JAT and BJAT outperform all

(a) ICA

Method	ATIS				SNIPS			
	OP1	OP2	OP3	Drop	OP1	OP2	OP3	Drop
AC	0.907	0.947	0.893	3.72%	0.919	0.951	0.882	5.04%
AC+JAT	0.913	0.954	0.917	2.93%	0.931	0.958	0.918	2.80%
AC+BJAT	0.926	0.960	0.924	3.13%	0.944	0.966	0.933	2.77%
MC	0.916	0.948	0.907	3.29%	0.926	0.950	0.903	4.60%
MC+JAT	0.927	0.957	0.919	**2.68%**	0.939	0.962	0.924	3.29%
MC+BJAT	0.927	0.959	0.922	3.01%	0.943	0.971	0.930	2.97%
BERT	0.930	0.974	0.940	3.16%	0.975	0.968	0.948	2.13%
BERT+BJAT	0.941	0.977	0.944	2.55%	0.975	0.978	0.951	**1.62%**

(b) SF1

Method	ATIS				SNIPS			
	OP1	OP2	OP3	Drop	OP1	OP2	OP3	Drop
AC	0.829	0.834	0.772	13.56%	0.776	0.721	0.625	21.15%
AC+JAT	0.883	0.849	0.758	12.26%	0.784	0.743	0.633	21.48%
AC+BJAT	0.935	0.851	0.771	10.57%	0.847	0.783	0.679	17.85%
MC	0.833	0.823	0.725	15.02%	0.760	0.710	0.628	22.48%
MC+JAT	0.873	0.837	0.762	12.98%	0.780	0.757	0.645	21.03%
MC+BJAT	0.927	0.861	0.805	**9.12%**	0.834	0.787	0.667	18.08%
BERT	0.819	0.790	0.673	20.8%	0.908	0.806	0.728	15.8%
BERT+BJAT	0.826	0.808	0.701	18.9%	0.917	0.812	0.731	**15.2%**

Table 3: **ICA** and **SF1** results of the baseline (AC & MC & BERT), JAT and BJAT on noise data.

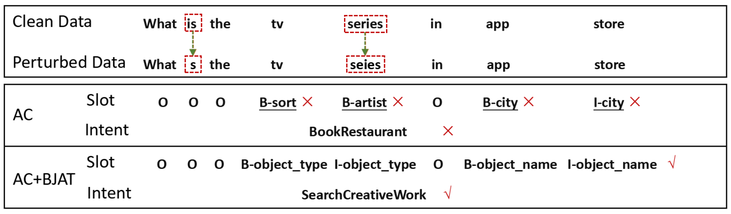

Clean Data
What is the tv series in app store
Perturbed Data
What s the tv seies in app store
AC
Slot
O O O B-sort × B-artist × O B-city × I-city ×
Intent
BookRestaurant ×
AC+BJAT
Slot
O O O B-object_type I-object_type O B-object_name I-object_name √
Intent
SearchCreativeWork √

Figure 3: An example illustrating wrong predictions of intent and slot labels on perturbed data.

baseline models across all noise types. Specifically, for ICA, BERT+BJAT drops only 2.55% on the ATIS noisy test set; BERT+BJAT drops 1.62% on the SNIPS noisy test set. For SF1, the performance drop is much more significant and our proposed methods have successfully reduced the drop to 9.12% on the ATIS noisy test set and 15.2% on the SNIPS noisy test set. In addition, we find that pre-trained language models (BERT-base-uncased in this experiment) are also sensitive to noisy data, and the performance drop due to noise is much larger on ATIS SF1 than SNIPS SF1.

Figure 3 shows an example with slot labels and intents predicted by the AC baseline and AC+BJAT. In this example, word "is" and "series" are misspelled as "s" and "seies" in the noisy input. Such spelling errors do not change the basic meaning of the input. However, the baseline AC model is completely fooled by these two errors to wrongly predict the intent as *BookRestaurant* and "app store" as a city name. In contrast, the proposed BJAT correctly predict both the intent and slot labels despite of spelling perturbations.

5 Analysis

In this section, we further take a deep look into data to investigate how the proposed adversarial training models improve intent detection and slot filling.

5.1 Slot Filling on Rare and Unseen Words

In the last section, we have conducted experiments to examine the robustness of the JAT and BJAT on noisy data with artificially generated perturbations. Now our interest is to evaluate the robustness of the two models on real data where perturbations occur in the way of unknown or rare words. Specifically, we want to study the robustness of the proposed models on slot filling over unseen/rare words.

Word Level If a word occurs less than 10 times in the training data, it is a rare word. We categorized all words in the test sets into four types: UNK (unseen words), U neighbor (the preceding and succeeding word of a UNK word), Rare (rare words) and R neighbor (the preceding and succeeding word of a rare word). We distinguish rare/unseen words from their neighboring words because we want to see the differences in the slot filling accuracy on these words. Table 4 display the results of models on the two datasets, and the percentages of these four types of words in all words. It can be seen that the percentages of these words on the ATIS dataset are much smaller than those on the SNIPS dataset.We can also find that the performance gap between UNK and U neighbor is larger than that between Rare and R neighbor. This is consistent with our intuition that unseen words are much more difficult to be correctly labeled even with correctly recognized neighbors than rare words. Both JAT and BJAT substantially improve the

(a) ATIS

Word	UNK	U Neighbor	Rare	R Neighbor
Percent.	0.56%	0.87%	2.99%	5.34%
AC	0.314	0.950	0.869	0.933
AC+JAT	0.549	0.963	0.905	0.947
AC+BJAT	0.510	0.963	0.927	0.955
MC	0.235	0.950	0.872	0.939
MC+JAT	0.333	0.963	0.916	0.943
MC+BJAT	0.549	0.963	0.927	0.951

(b) SNIPS

Word	UNK	U Neighbor	Rare	R Neighbor
Percent.	5.81%	10.47%	11.08%	19.74%
AC	0.772	0.862	0.879	0.894
AC+JAT	0.808	0.886	0.912	0.928
AC+BJAT	0.848	0.928	0.932	0.946
MC	0.772	0.859	0.885	0.906
MC+JAT	0.846	0.916	0.923	0.939
MC+BJAT	0.886	0.931	0.903	0.923

Table 4: Slot prediction accuracy on rare/unseen words and their neighboring words in the test set.

Slot	ATIS		SNIPS	
	UNK	Rare	UNK	Rare
Percent.	0.90%	4.78%	10.84%	20.73%
AC	0.091	0.800	0.792	0.885
AC+JAT	0.333	0.851	0.834	0.916
AC+BJAT	0.242	0.891	0.868	0.935
MC	0.091	0.806	0.792	0.888
MC+JAT	0.121	0.874	0.873	0.932
MC+BJAT	0.333	0.897	0.899	0.904

Table 5: Slot prediction accuracy on rare/unseen slot words on the ATIS and SNIPS test set. Percent. denotes the percentage of unseen/rare slot words in all slot words.

Method	SF1
(Kim et al., 2019)	0.929
(Yoo et al., 2019)	0.893
AC+BJAT (ours)	**0.937**
MC+BJAT (ours)	0.931

Table 6: Comparison to recent data augmentation approaches on the SNIPS SF1.

performance on all these types of words. The improvements on unseen words are larger than those on rare words.

Slot Level We further distinguish two groups of words: slot words (i.e., words labeled as "B-XXX" or "I-XXX") and contextual words (i.e., words labeled as "O"). Table 5 show the results on unseen/rare slot words and their neighbors. It can be clearly seen that the prediction performance on unseen/rare slot words are also significantly improved.

Comparison to Data Augmentation Approaches Due to the higher percentages of UNK and Rare words as described in Section 5.1, the SNIPS dataset contains more uncertainty on slot filling task. Therefore, we further compared with two data augmentation approaches recently proposed on SNIPS SF1. Results in Table 6 show that our method achieves better performance. For fair comparison, we only used lightweight models (AC & MC).

5.2 Slot Filling on Multi-Slots Sentences

In Table 1, we find that both JAT and BJAT achieve slight improvements on ICA, but high improvements on SOA. To explain this, we made a further analysis. We introduce a new metric called sentence-level slot accuracy (SSA) to measures the percentage of sentences where all slot labels are correctly predicted. We group sentences according to the number of slots[1] in sentences. Generally speaking, the more slots there are in a sentence, the harder it is to correctly predict all slot labels for the sentence. Sentences in the test sets are classed into two groups: sentences with ≤ 3 slots and sentences with ≥ 4 slots. Figure 4 shows the SSA results on the two groups of sentences in the ATIS and SNIPS test set. We can find that both JAT and BJAT achieve large improvements on the two test sets in terms of SSA. As many sentences have a correctly predicted intent label but a wrongly predicted sentence-level slot label sequence, the large improvements in SSA naturally lead to large improvements in SOA.

The improvements on the second group (≥ 4 slots) of the ATIS test set are larger than those of the SNIPS test set. This may be because the number of slots in ATIS ranges from 0 to 13 while SNIPS 0 to 6. We also find that the performance improvements obtained by the proposed models on sentences with 4+ slots are larger than those on sentences with ≤ 3 slots in the ATIS test set. This indicates that the baseline models is weak at dealing with multi-slot filling while our models are capable of handling complex slot filling cases. Different from ATIS, on the SNIPS test set, our models achieve large improvements on

[1]A slot is defined as a span starting from a word labeled as "B-XXX" and ending at a word just before the nearest word labeled as "O".

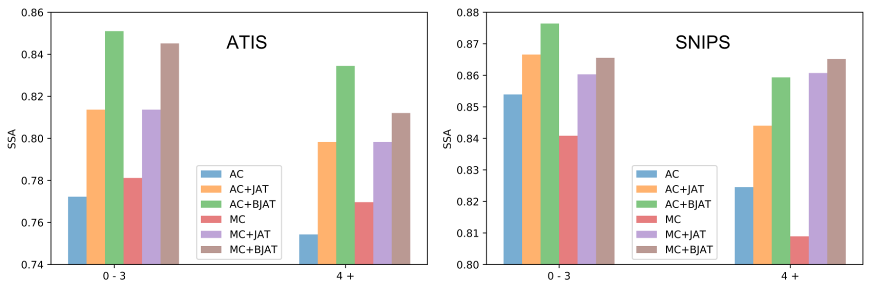


Figure 4: Results on sentences with multiple slots in the two test sets.

Intent Type	atis_flight	other intents
percent	73.9%	26.1%
AC	0.986	0.879
AC+JAT	0.989	0.888
AC+BJAT	0.987	0.932
MC	0.986	0.895
MC+JAT	0.989	0.902
MC+BJAT	0.990	0.918

Table 7: ICA results of the baseline, JAT and BJAT models.

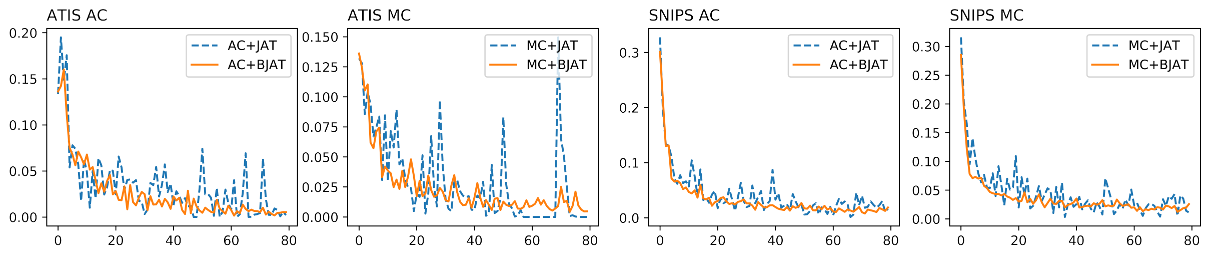


Figure 5: Learning curves of the JAT vs. BJAT model on the two datasets. The ordinate axis is calculated by $JL_{adv} - JL$. The abscissa axis denotes the number of epochs.

sentences with $\leq$ 3 slots, comparable to those on sentences with 4+ slots. We conjecture that this is because sentences in the SNIPS test set have more rare/unseen words, which makes it difficult for the baseline to predict correct slot labels even for sentences with less than 3 slots.

5.3 Unbalanced Intent Distribution

ATIS contains of 19 different intent types including multi-intent like “atis_flight#atis_airfare”. Unlike SNIPS, the distribution of intent types in ATIS are highly unbalanced, where “atis_flight” accounts for 73.9% while other 18 types of intent account for only 26.1%, distributing sparsely . To investigate the effect of JAT and BJAT on this unbalanced intent distribution problem, we evaluated ICA results on the “atis_flight” intent versus other intents, which are shown in Table 7. We observe that the ICA scores of the two baseline models are very high as the prediction on this intent type is well trained with sufficient instances. The improvements are therefore small. However, for the sparse intent types, the JAT and BJAT models achieve remarkable improvements over the baseline models in terms of ICA, indicating that the proposed models are robust on rare intent types, similar to slot filling on rare/unseen words.

5.4 Learning Curves

In Figure 5, we demonstrate the learning curves of the proposed JAT vs. BJAT model on the two datasets in terms of the difference between the joint adversarial loss on the joint adversarial examples and the joint loss on the original data along with training epochs. We observe that the generated joint adversarial examples make the training of the JAT model vibrate substantially even after sufficient training epochs (e.g., 40 epochs) as the adversarial examples have very different impacts on the ID and SF loss. By contrast, the training of the BJAT model with the proposed balance factor is much more smoothing and stable.

6 Conclusion

We have presented a joint adversarial training framework to improve the robustness of the joint intent detection and slot filling. To balance the impacts of the joint adversarial examples on the intent detection loss and slot filling loss, we further propose a balanced joint adversarial training model with a balance factor as a regularization term. Experiment results on the ATIS and SNIPS datasets demonstrate the capability of our approaches in improving both the performance and robustness. Our in-depth analyses further disclose that both JAT and BJAT can (1) boost slot filling accuracy for rare/unseen words, (2) deal with noisy inputs such as spelling errors to some extent, (3) perform well on complex sentences with multiple slots, and (4) successfully handle the problem of unbalanced intent distribution. Our experiments and analyses also suggest that the balanced JAT is better than JAT in both the performance and training stability.

Acknowledgements

This work is supported by the National Key R&D Program of China (2019YFB1406302), National Natural Science Foundation of China (No. 61502033, 61472034, 61772071, 61272361 and 61672098) and the Fundamental Research Funds for the Central Universities.

References

Giannis Bekoulis, Johannes Deleu, Thomas Demeester, and Chris Develder. 2018. Adversarial training for multi-context joint entity and relation extraction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2830–2836, Brussels, Belgium, October-November. Association for Computational Linguistics.

Yoshua Bengio and Yann LeCun, editors. 2015. *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings.*

Qian Chen, Zhu Zhuo, and Wen Wang. 2019. BERT for joint intent classification and slot filling. *CoRR*, abs/1902.10909.

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

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. 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 Anna Korhonen, David R. Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28-August 2, 2019, Volume 1: Long Papers*, pages 5467–5471. Association for Computational Linguistics.

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, June. Association for Computational Linguistics.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and harnessing adversarial examples. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Daniel Guo, Gökhan Tür, Wen-tau Yih, and Geoffrey Zweig. 2014. Joint semantic utterance classification and slot filling with recursive neural networks. In *2014 IEEE Spoken Language Technology Workshop, SLT 2014, South Lake Tahoe, NV, USA, December 7-10, 2014*, pages 554–559.

Ying Ju, Fubang Zhao, Shijie Chen, Bowen Zheng, Xuefeng Yang, and Yunfeng Liu. 2019. Technical report on conversational question answering. *CoRR*, abs/1909.10772.

Hwa-Yeon Kim, Yoon-Hyung Roh, and Young-Kil Kim. 2019. Data augmentation by data noising for open-vocabulary slots in spoken language understanding. In Sudipta Kar, Farah Nadeem, Laura Burdick, Greg Durrett, and Na-Rae Han, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 3-5, 2019, Student Research Workshop*, pages 97–102. Association for Computational Linguistics.

Changliang Li, Liang Li, and Ji Qi. 2018. 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, October-November. Association for Computational Linguistics.

Bing Liu and Ian Lane. 2016. Attention-based recurrent neural network models for joint intent detection and slot filling. In Nelson Morgan, editor, *Interspeech 2016, 17th Annual Conference of the International Speech Communication Association, San Francisco, CA, USA, September 8-12, 2016*, pages 685–689. ISCA.

Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. 2017. Delving into transferable adversarial examples and black-box attacks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*.

Takeru Miyato, Andrew M. Dai, and Ian J. Goodfellow. 2017. Adversarial training methods for semi-supervised text classification. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. *CoRR*, abs/1910.01108.

Gökhan Tür, Dilek Hakkani-Tür, and Larry P. Heck. 2010. What is left to be understood in atis? In *2010 IEEE Spoken Language Technology Workshop, SLT 2010, Berkeley, California, USA, December 12-15, 2010*, pages 19–24.

Ngoc Thang Vu. 2016. Sequential convolutional neural networks for slot filling in spoken language understanding. In Nelson Morgan, editor, *Interspeech 2016, 17th Annual Conference of the International Speech Communication Association, San Francisco, CA, USA, September 8-12, 2016*, pages 3250–3254. ISCA.

Yu Wang, Yilin Shen, and Hongxia Jin. 2018. A bi-model based RNN semantic frame parsing model for intent detection and slot filling. 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 309–314, New Orleans, Louisiana, June. Association for Computational Linguistics.

Yi Wu, David Bamman, and Stuart Russell. 2017. Adversarial training for relation extraction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1778–1783, Copenhagen, Denmark, September. Association for Computational Linguistics.

Michihiro Yasunaga, Jungo Kasai, and Dragomir Radev. 2018. Robust multilingual part-of-speech tagging via adversarial training. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 976–986, New Orleans, Louisiana, June. Association for Computational Linguistics.

Kang Min Yoo, Youhyun Shin, and Sang-goo Lee. 2019. Data augmentation for spoken language understanding via joint variational generation. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 7402–7409. AAAI Press.

Xiaodong Zhang and Houfeng Wang. 2016. A joint model of intent determination and slot filling for spoken language understanding. In Subbarao Kambhampati, editor, *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, pages 2993–2999. IJCAI/AAAI Press.