TRANSFERRING SLU MODELS IN NOVEL DOMAINS

Anonymous authors
Paper under double-blind review

ABSTRACT

Spoken language understanding (SLU) is a critical component in building dialogue systems. When building models for novel natural language domains, a major challenge is the lack of data in the new domains, no matter whether the data is annotated or not. Recognizing and annotating "intent" and "slot" of natural languages is a time-consuming process. Therefore, spoken language understanding in low resource domains remains a crucial problem to address. In this paper, we address this problem by proposing a transfer-learning method, whereby an SLU model is transferred to a novel but data-poor domain via a deep neural network framework. We also introduce meta-learning in our work to bridge the semantic relations between seen and unseen data, allowing new intents to be recognized and new slots to be filled with much lower new training effort. We show the performance improvement with extensive experimental results for spoken language understanding in low resource domains. We show that our method can also handle novel intent recognition and slot-filling tasks. Our methodology provides a feasible solution for alleviating data shortages in spoken language understanding.

1 INTRODUCTION

The recent surge of artificial intelligence motivates the technical and applicable exploration of novel human-computer interactions. Spoken dialogue systems are widely studied and used in various mobile devices. Well-known commercial applications driven by dialogue systems include intelligent personal assistants and intelligent robots for customer services. As more and more products and scenarios integrate dialogue systems in intelligent services, the ability for model adaptation in dialogue systems is critically needed. Many state-of-the art dialogue systems follow a learning pipeline that includes components such as spoken language understanding (SLU), dialogue management (DM) and spoken language generation or retrieval (SLG). Since both dialogue management and spoken language generation systems rely on knowledge or information learned from spoken language understanding, research and industry community have paid much attention to the SLU area. In this paper, we focus on the problem of data shortage for SLU in a new domain.

Spoken language understanding typically defines and represents user utterances in terms of semantic frames comprised of domains, intents and slots (Tur & Mori, 2011). A spoken language understanding task involves classifying the domain, detecting the intent of user utterance and identifying token sequences corresponding to slot values in the semantic frame. A learning model is often designed to handle the above subtasks separately and sometimes simultaneously. Deep learning models are the cutting-edge techniques for achieving impressive performance in spoken language understanding compared with conventional machine learning models.

However, deep neural-network models often require the collection and manual annotation of data, which is a time and labor intensive process. These problems pose as a major obstacle to building accurate learning models in new domains. To address this problem, we propose to perform domain adaption of SLU models by transferring the SLU knowledge and model parameters learned in an auxiliary domain where a model is already built. A related problem is that only few examples of some output intent and slot classes are available during the training phrase in a new domain, because training sets for a new domain are small. To deal with this limitation, we propose a novel few shot learning model in a meta learning paradigm, we call it the Transfer Semantic Similarity Model (TSSM). The TSSM achieves few shot learning by exploiting the semantic embeddings of both seen and unseen intent and slots.

In the related work section, we argue that the current spoken language understanding tasks often do not take the relationship between intents and slots into consideration, even when these works classify the intents and slots in a novel SLU domain simultaneously in a single model. We propose a "intent2slot" mechanism which consider such relationship and show that the TSSM can indeed achieve better performance in natural language understanding.

In summary, the challenges of spoken language understanding include:

- Expanding SLUs to novel domains is necessary as the development of artificial intelligent products extends to more areas in our society. But it is difficult to get sufficient data or spoken language sentence in novel domains.
- Without sufficient labeled data, deep learning methods are difficult to train and thus it is hard to obtain satisfactory performance in a new domain.
- Unseen intents and slots exist because small annotated training set might not cover all intents and slots. Thus, the generalization ability of the models are crucial for dealing with unseen classes.

Our contributions are summarized as follows:

- We design a novel and efficient SLU model, the TSSM. It incorporates transfer learning and meta learning for SLU tasks in low resource domains.
- The proposed TSSM can improve model performance with only a small amount of training data in new domains and it can handle unseen classes easily. The experiments illustrate that our proposed methods outperform the state-of-the-art baselines in many experimental domains.
- Our work can treat intent and slot detection as a structured multi-objective optimization problem, which further increases the accuracy of learning tasks in new domains.

2 Related Work

Spoken language understanding is usually defined as a supervised learning problem, involving conventional machine learning models (Young, 2002; Hahn et al., 2011; Wang et al., 2005) and deep learning models (Mesnil et al., 2015; Kurata et al., 2016; Sarikaya et al., 2011) on massive amount of annotated training data. Although unsupervised learning and semi-supervised learning based approaches have been proposed as well (Chen et al., 2013; 2015b;a), deep learning approaches were shown to outperform most others. Besides the requirement of a large amount of annotated data being available, the intents and slots are usually predefined. Thus they are inflexible to expand to new domains. Some researchers have paid attention to alleviate this limitation. An example is Korpusik et al. , which work retrains its models to cover new intents and associated slots while redesigning a semantic schema (Korpusik et al., 2014).

Considering that models for different domains are usually trained independently, some attempts have been made to learn shared latent semantic representations or shared model parameters for multiple domains via transfer learning. A survey for transfer learning with applications is Pan & Yang (2010). In general, the goal of transfer learning is to improve performance in low data domains and reduce model re-training efforts by bridging the knowledge from data-rich source tasks to new target tasks. Transfer learning in deep learning setting has been applied to recommendation systems and computer vision domains. In natural language processing (NLP), researchers have studied cross-domain NLP problems but have mainly focused in cross-lingual problem settings (Mou et al., 2016; Buys & Botha, 2016; Kim et al., 2017a;b). In Hakkani-Tr et al. (2016), multi-task and transfer learning approaches are proposed to learn shared implicit feature representation across various domains for spoken language understanding. Another study Jaech et al. (2016) follows the similar idea in cross-domain NLP and showed significant performance improvement. Researchers in Amazon have shown in a recent study that DNN-based natural language engines can re-use the existing models through transfer learning (Goyal et al., 2018).

When certain classes do not show up in the training data, one can exploit meta learning to use few-shot learning to help with knowledge transfer Fei-Fei et al. (2006); Palatucci et al. (2009); Vinyals et al. (2016); Ravi & Larochelle (2018). For example, Vinyals et al. (2016) learns a network that

maps a small labeled support set and unlabeled examples to its labels, saving the effort in fine-tuning to adapt to new class types. Besides image object recognition tasks, in SLU some experiments have been conducted. The objective of Ferreira et al. (2015) is to predict the semantic tag sequences of a user query without using any target-domain user utterances and thus in-context semantic tags. The research studies build a statistical model to predict the SLU for unseen data in Yazdani & Henderson (2015); Chen et al. (2016). An Action Matching algorithm is proposed by Zhao & Eskénazi (2018) to learn a cross-domain embedding space that models the semantics of dialog responses, which, in turn, lets a neural dialog generation model generalize to new domains. Recent publication by Google (Bapna et al., 2017) explores semi-supervised slot-filling based on deep learning that can use the slot labels without the need for any labeled or unlabeled data in domain examples or the need for any explicit schema alignment, to quickly bootstrap the model in a new domain.

3 Our Proposed Approach for Model Transfer

In this section, we introduce our transfer learning approach to address the cross-domain SLU problem. We consider a source domain D_s and a target domain D_t. Typically, we have a large annotated dataset for D_s, while only a few labeled data for D_t. The tasks of interest in both D_s and D_t are of the same types, namely they are both slot filling and intent detection. However, D_s and D_t have different intent label spaces $\mathbb{T}_s$ and $\mathbb{T}_t$, because the target domain might have new intent values (slot faces the same situation).

The architecture of our model is illustrated in Fig. 1. The model contains: (1) a word hashing layer to convert one-hot word representations into embedding vectors, (2) a bi-directional Long Short Term Memory networks(LSTMs) (Hochreiter & Schmidhuber, 1997) layer to capture features from word embeddings, (3) a intent specific bi-LSTMs layer combined with an semantic network to detect intent, (4) a slot specific bi-LSTMs layer powered with an attention component and semantic network to conduct slot filling. We first pre-train this model in a source domain with large amount of annotated data. We wish to allow the model to extend to new slot names, slot values and intent names in the target domain with little annotated data.

The number of possible intents/slots grows rapidly as we get into new domains with new intents and slots. One natural approach to this task is to train one binary classifier for each possible label, to decide whether or not to include it in the output. However, this would require training a large number of classifiers. It also would be impossible to generalize to target domains that include intents and slots that do not show up in the training set, since there will not be any parameter sharing among label classifiers. Instead of encoding slot names and intent names as discrete features in the target domain, we encode them using a continuous representation. To be specific, we learn a representation of slot names, slot values and intent names from their constituent words using semantic networks. We then check to see if these representations match the representations of utterances. In the following, we explain in details how this representation works in our transfer learning algorithm.

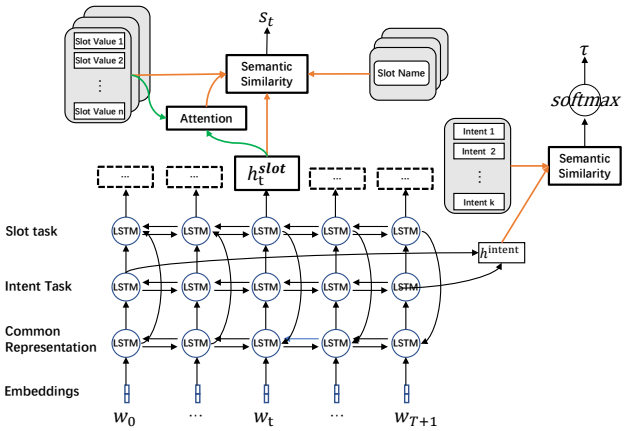

Figure 1: TSSM archiecture for joint SLU. The orange lines are inputs for semantic similarity network and the green lines represent the inputs for attention network.

Let $\boldsymbol{w} = (w_0, w_1, w_2, \ldots, w_{T+1})$ represent the input word sequence with w_0 and w_{T+1} being the beginning-of-sequence ($\langle bos \rangle$) and end-of-sequence ($\langle eos \rangle$) tokens, respectively. T is the number of words. Let $\boldsymbol{s} = (s_0, s_1, \ldots, s_T)$ be the slot label sequence, where s_0 is a padded slot label that maps to the beginning-of-sequence token $\langle bos \rangle$ and $s_i \in \mathbb{S}$. Let $\boldsymbol{\tau} \in \mathbb{T}$ be the intent class. $\mathbb{S}$ and $\mathbb{T}$ are the slot and intent label space, respectively. Let $\boldsymbol{sn} = (\boldsymbol{sn}_1, \boldsymbol{sn}_2, \ldots, \boldsymbol{sn}_m)$ represent the m slot names. $\boldsymbol{sn}_i = (w_1, \ldots, w_{m_i})$ is a vector of the words for the i-th slot name, where m_i is its number of words. Each slot s_i has a list of slot values $\boldsymbol{sv_i} = (\boldsymbol{sv}_{i,1}, \boldsymbol{sv}_{i,2}, \ldots, \boldsymbol{sv}_{i,n_i})$, where n_i is the number of slot values for the i-th slot. A slot value is allowed to be a phrase like 'fried chicken' for slot 'Food'. Let $\boldsymbol{sv}_{i,j} = (w_1, \ldots, w_{n_{i,j}})$ be the words of the j-th slot value for i-th slot and $n_{i,j}$ is the number of words for this slot value.

3.1 Base Networks

We use a Pretrained DNN model as our baseline model (Goyal et al., 2018). The model has three layers of bi-LSTMs. The first layer is the common bi-LSTMs layer, which learns task related representations from input utterances. The learned representations are taken as input to the upper two bi-LSTMs layers, which are optimized separately for the slot-filling task and intent-detection task.

Our model modify the baseline model in three aspects. The main difference is that Goyal et al. (2018) treat intent and slot as discrete symbols and train classifiers to make predictions. Such a practice limits the knowledge transfer between two domains as the classifier layers following the upper two bi-LSTMs layers need to be re-initialized when fine-tune on new domains (the output labels are different). In our model, we encode intent and slot as continuous representations by their constituent words. The classification problems are transformed to semantic similarity problems and the whole model parameters could be transferred to new domains. Section 3.2 will introduce the semantic network that encode intent name and slot name into semantic vectors.

The second difference is the usage of *gazetteers* (lists of slot values for a slot). Goyal et al. (2018) use gazetteer features as an additional input. Such features are binary indicators of the presence of an n-gram in a gazetteer. In our model, we use attention network (Section 3.3) to encode external slot values into a semantic vector that represents the slot from the value perspective, which suits our semantic framework naturally.

Finally, there are no connections between the upper two layers in Pretrained DNN (Goyal et al., 2018). However, we believe that the output of intent specific bi-LSTMs layer could benefit the slot detection task. As one of our contributions, in our network, we concatenate the output of common bi-LSTMs layer and intent bi-LSTMs layer to feed to the slot bi-LSTMs layer.

For the intent specific bi-LSTMs layer, we concatenate the last hidden state of each direction to acquire the global representation of the whole utterance $\mathbf{h}^{intent} = [\overrightarrow{\mathbf{h}}^{intent}_{T+1}, \overleftarrow{\mathbf{h}}^{intent}_{0}]$. Such an operation converts utterances with variable lengths into a fixed-length vector, with which the information throughout the entire utterance can be captured. Representation $\mathbf{h}^{intent}$ will then be used to calculate the similarity scores with representations of candidate intents. We have also tried other methods like take Mean-Pooling or Max-Pooling of bi-LSTM hidden states to acquire $\mathbf{h}^{intent}$ but get almost the same results.

3.2 Semantic Networks

We wish to build a representation-learning mechanism for the SLU task that can generalize to unseen words and labels. In order to represent words, we use word embeddings, which are a form of vector space model. In various NLP tasks (Baroni et al., 2014; Collobert & Weston, 2008; Mikolov et al., 2013), word embeddings have proven to be effective models of semantic representation of words. Because word embeddings trained on large corpora of unlabeled data reflect the similarity between words, they help the model generalize from the words in the original training domains to the words in the new target domains. For every utterance we learn how to compose the word embeddings to reflect the semantics of that utterance. Furthermore, we learn how to compose the semantic representation of each intent/slot from the embeddings of the words used to name that intent/slot. This enables us to generalize to unseen intents/slots.

For slot names, slot values and intent names, we use the same semantic network and transform it to a fixed-length representation vector. Take slot name $\boldsymbol{sn}_i = (w_1, w_2, \ldots, w_{m_i})$ as an example, the

semantic network first extracts the word embedding $E(w_i)$ for each word. $E(w_i)$ is then fed to a feed forward layer to get the non-linear semantic features. The last is a Mean-Pooling layer, which applies the mean operations over each dimension of semantic features across all words. The output is treated as the semantic representation for $\boldsymbol{sn}_i$,

$$\mathbf{r}_i^{slot_name} = \frac{1}{m_i}\sum_{j=1}^{m_i} relu\left(E(w_j)^\intercal \mathbf{W}_{slot_transfrom} + b_{slot_transfrom}\right) \tag{1}$$

where $\mathbf{W}_{slot_transfrom}$ and $b_{slot_transfrom}$ are the learned linear projection matrix and bias for slot name semantic network respectively. In the same way, we could get the semantic representation $\mathbf{r}_{i,j}^{slot_value}$ for slot value $\boldsymbol{sv}_{i,j}$ and $\mathbf{r}_i^{intent_name}$ for the intent name τ_i.

More sophisticated vector-space semantic representations of the slots/intents are an area for future work, but we believe the simple Mean-Pooling would be a proper choice here since intent names, slot names and slot values are typically very short and composed by entity names.

3.3 Slot Value attention Networks

With previously introduced semantic networks, we can formulate a semantic representation $\mathbf{r}_{i,j}^{slot_value}$ of each slot value $\boldsymbol{sv}_{i,j}$ from the i-th candidate slot s_i. The whole slot value representation list is $\left(\mathbf{r}_{i,1}^{slot_value}, \ldots, \mathbf{r}_{i,n_i}^{slot_value}\right)$.

At each decoding step t, the slot bi-LSTMs layer output state $\mathbf{h}_t^{slot}$ is connected to each slot's slot values' representations to calculate the similarity scores. Take slot s_i as an example, we compute an attention weight $\alpha_{t,i,j}$ for slot value $\boldsymbol{sv}_{i,j}$ via a bilinear operator, which reflects how relevant or important slot value $\boldsymbol{sv}_{i,j}$ is to the current slot hidden states $\mathbf{h}_t^{slot}$,

$$\alpha_{t,i,j} \propto \exp((\mathbf{r}_{i,j}^{slot_value})^\intercal \mathbf{W}_v \mathbf{h}_t^{slot}) \tag{2}$$

where $\mathbf{W}_v$ is a parameter matrix. The overall slot value representation for slot s_i would be

$$\mathbf{r}_{t,i}^{slot_values} = \sum_{j}^{n_i} \alpha_{t,i,j}\mathbf{r}_{i,j}^{slot_value}. \tag{3}$$

3.4 Training Procedure

The source domain data containing utterances and associated intents/slots is used to train the model. The idea behind this model is to learn the representation for both utterances and intents/slots such that utterances with the same intents/slots are close to each other in the continuous semantic space. We define a semantic score between an utterance and an intent/slot using the similarity between their representation embeddings $\mathbf{h}$ and $\mathbf{r}$ by $Sim(\mathbf{h}, \mathbf{r}) = \mathbf{h}^\intercal \mathbf{W}_s \mathbf{r}$. $\mathbf{W}_s$ is the corresponding similarity matrix for slot or intent task.

The posterior probability of a possible intent given an utterance is computed based on the semantic score through a softmax function

$$P(\tau = \tau_i | \mathbf{h}^{intent}, \boldsymbol{w}) = \frac{\exp\left(Sim(\mathbf{h}^{intent}, \mathbf{r}_i^{intent_name})\right)}{\sum_{\tau_j in \mathbb{T}} \exp\left(Sim(\mathbf{h}^{intent}, \mathbf{r}_j^{intent_name})\right)} \tag{4}$$

where τ_i is an intent candidate. Similarly, we can get the posterior probability of a possible slot at each time. Since we have the representations of both slot name and slot value, we first add the semantic scores from slot name and slot value and then feed to the softmax function to get the overall slot posterior probability

$$Sim(s_i, \mathbf{h}_t^{slot}) = Sim(\mathbf{h}_t^{slot}, \mathbf{r}_i^{slot_name}) + Sim(\mathbf{h}_t^{slot}, \mathbf{r}_i^{slot_values}) \tag{5}$$

$$P(s_t = s_i | \mathbf{h}_t^{slot}, \boldsymbol{w}) = \frac{\exp\left(Sim(s_i, \mathbf{h}_t^{slot})\right)}{\sum_{s_j \in \mathbb{S}} \exp\left(Sim(s_j, \mathbf{h}_t^{slot})\right)} \tag{6}$$

where s_i is an slot candidate.

For model training, we maximize the likelihood of the correctly associated intents/intents given all training utterances. The parameters $\mathbf{W}$ of the model are optimized by an joint negative log likelihood objective:

$$\mathcal{L} = -\sum_{k=1}^{l} \log P\left(\tau = \tau_i | \mathbf{h}_{\mathrm{w}_k}^{intent}, \boldsymbol{w}_k\right) - \sum_{k=1}^{l}\sum_{t=0}^{T_k} \log P\left(s_t = s_i | \mathbf{h}_{t,\mathrm{w}_k}^{slot}, \boldsymbol{w}_k\right) \tag{7}$$

where l is the total number of training utterances, $\mathbf{w}_k$ is the k-th input utterance. The model is optimized using mini-batch stochastic gradient descent(SGD) (Huang et al., 2013).

3.5 Transfer to A New Domain

We train our TSSM using labeled data from source domain D_s. This stage is called pre-training, where the networks learn to extract semantic relationships between utterance representations and intent/slot representations. The model is able to adapt to new domains, since the seen and unseen intents/slots representations are in the same semantic space, use the same shared composition semantic network with the same unsupervised word embeddings as input. It has been shown that such word embeddings can capture word similarities and hence the classifier is no longer ignorant about any new intent or slot.

After the pre-training, we fine-tune our model on the target domain D_t with less annotated data and with unseen intents/slots. The model for the new domain D_t are the same as that for D_s and are initialized by the pre-trained network parameters. We tried fine-tune different components of the model. We fix some parts and fine-tune on the other parts. After experiments on several combinations of components, we found that fine-tune the whole model gives us the best performance.

For zero-shot learning where no fine-tune data is avaiable, in order to predict new intents/slots on new utterances, the TSSM calculates semantic similarites. We transform each input utterance $\mathbf{w}$ into a vector $\mathbf{h}^{intent}$ using the base network, and then estimate its semantic similarity with all intents including seen and unseen intents. The vector representations for new intents can be generated from the trained semantic network by feeding the word embedding vectors of new intents as the input. For the utterance $\mathbf{w}$, the estimated semantic score of the i-th intent is defined as $Sim(\mathbf{h}^{intent}, \mathbf{r}_i^{intent})$ in Equation 3.4. Then predicted intent for each utterance is decided according to the estimated semantic scores. Predicting slot sequences is similar to predicting intents except that the calculation is conducted step by step at each position.

4 Experiments and Results

In this section, we first conduct extensive experiments to verify the transfer ability of the TSSM with different number of training data in the target domain. Then we verify the generalization ability of the TSSM with different number of new labels in the target domain. Finally we verify the zeros-shot learning ability of the TSSM model and verify the effectiveness of the proposed "intent2slot" mechanism.

4.1 Dataset

In this experiment, we use five popular public dialogue datasets from five domains and three datasets obtained from practical applications. The CamHotel dialogue dataset (Hotel) (Wen et al., 2015) records some dialogues about booking a hotel room in Cambridge. The CamRestaurant dialogue dataset (Restaurant) (Wen et al., 2016c;a) contains dialogue records about booking a restaurant in Cambridge. The Laptop and TV dialogue dataset (Wen et al., 2016b) are dialogues about buying Laptops and TVs. The Airline Travel Information System dataset (Atis) (Price, 1990) contains dialogues about airline travel, such as the fligh time, the destination, etc.

Weilidai, Huobijijin and Weizhongyouzhe are three financial products from WeBank. While Weilidai is very popular and its online customer service accumulated a lot of queries, Huobijijin and Weizhongyouzhe are newly developed products and have only few queries. We would like to pre-train on Weilidai queries and fine-tune on Huobijijin and Weizhongyouzhe queries to improve the online customer service's experience of the last two products.

In order to thoroughly evaluate the transfer learning ability on more diversified domains, we make five different pairs of source and target domains, as shown in table 1. In case when we have multiple source domains, all source domains will be combined into a large source domain. For each pair of the source and target domain pair, we first pretrain the model on the source domain. Then we fine-tune the model on the target domain training data, finally we evaluate the fine-tuned model on the target domain testing dataset. We use accuracy to evaluate the intent classification task and we use accuracy and F1 score to evaluate the slot filling task.

Table 1: Data Sizes of the source and target domains.

Source Domain	Target Domain	#Src Sentence	#Tgt Sentence
Restaurant, Laptop, TV, Atis	Hotel	48566	2168
Restaurant, Hotel, Laptop, TV	Atis	45756	4978
Atis	Hotel	4978	2168
Weilidai	Huobijijin	10000	3000
Weilidai	Weizhongyouzhe	10000	2000

4.2 BASELINES

We compare the proposed TSSM with the Pretrained DNN (Goyal et al., 2018) and two popular baselines that do not use transfer learning.

1. The Pretrained DNN (Goyal et al., 2018) (DNN) shares low-level word features between the intent classifiation task and the slot filling task.
2. The Max Entropy model (MaxEntropy) is a popular baseline for intent classification task, which uses bag-of-words features. This model cannot benefit from the source domain since the source and the target domains have different intents.
3. The Conditional Random Field model (CRF) is a popular baseline for slot filling task. It leverages the pretrained word-embedding models and it can model the transformation probability among sequential labels. This model also cannot benefit from the source domain.

4.3 RESULTS UNDER DIFFERENT SIZES IN TARGET-DOMAIN DATA SETS

In order to evaluate the transfer learning performance for the proposed model, we compare the proposed model with the baselines when we have different number of target domain training data. After pretraining on the source domain, we fine-tune all models on different percentage of target domain data, and we evaluate the performance of the models on the test set. We use 50% of the target domain data as testset, and we vary the percentage of trainset used from 5% to 46%.

The results are shown in Figure 2. As a higher percentage of target domain data is used to fine-tune the model, the performance of all models improves. Generally, transfer learning models perform better than non-transfer models, which implies that transfer learning can help improve the performance in the target domain. The proposed TSSM model outperforms the baseline model significantly in almost every pair of source and target domains, and for both intent classification task and slot filling task. As the amount of target domain training data increases, the improvement of the TSSM model remain very significant, which demonstrates the TSSM model can effectively adapt to new domains. In some cases, the proposed TSSM is not better than Pre-trained DNN. This might because that our model has more parameters than the baseline DNN model and in some cases simper model performs better.

Compared with intent classification task, the slot filling task benefit more from transferring from the source domain. This is because the intent has larger domain differences compared with the slots. For example, the intent in the hotel domain contains verbs such as “inform” “inform_no_match” while the intent in the atis domain contains nouns such as “flight” “aircraft”. While the slots in all domains mostly contain nouns. This suggest that it is better to transfer from a more similar source task. The improvement of transferring from Atis to Hotel dataset is worse than the other source and target pairs. We think this is because the intent and slot difference between “Atis” and ’Hotel’ domains are bigger than the domains difference between other tasks.

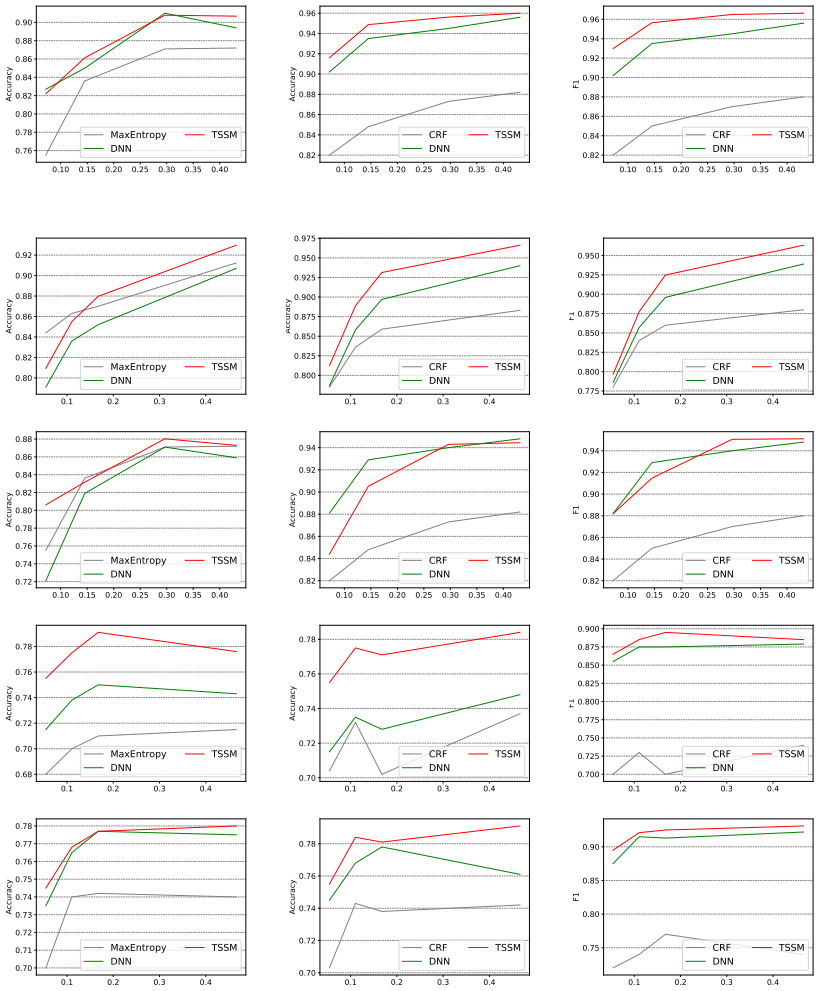

Figure 2: Performance comparison under different percentage of training data used in the target domain fine-tuning. In each figure, X-axis is the percentage of data used in the target domain, y-axis is the performance (higher is better). The three columns corresponds to three different metrics, that is the intent accuracy (left column), the slot accuracy (middle column) and the slot f1 score (right column). The three rows corresponds to different source and target domain pairs. The first row is obtained when transferring from "Restaurant, Laptop, TV, Atis" to "Hotel". The second row are the results transferring from "Restaurant, Hotel, Laptop, TV" to "Atis", and the third row are transferring from "Atis" to "Hotel". The fourth row are transferring from "Weilidai" to "Weizhongyouzhe", and the fifth row are transferring from "Weilidai" to "Huobijijin"

4.4 Results under different numbers of new labels in the target domain

In order to evaluate the generalization ability of the proposed TSSM model, we test the model on previously unseen testing labels. More specifically, we evaluate the proposed model under situations when the target domain test set have different number of new labels that is not seen in the target domain training set. We remove zero, one and two labels and their corresponding training instances from the target domain training set.

The results are shown in Figure 3. Here we use 10% of all target domain data as the training set, when the percentage of target domain training data increases to 20% or 30%, the results are similar. As the number of unseen label increases in the target domain testset, the performance of all model drops significantly. Transfer learning methods again out-performans none-transfer learning baselines, which demonstrate that transferring knowledge from source domain is generally beneficial to the target domain. As the number of unseen label increases, the performance drop of the TSSM model is significantly slower than the other baselines models. Compared with a softmax classification layer, the semantic similarity module works better since it can still computes semantic similarity

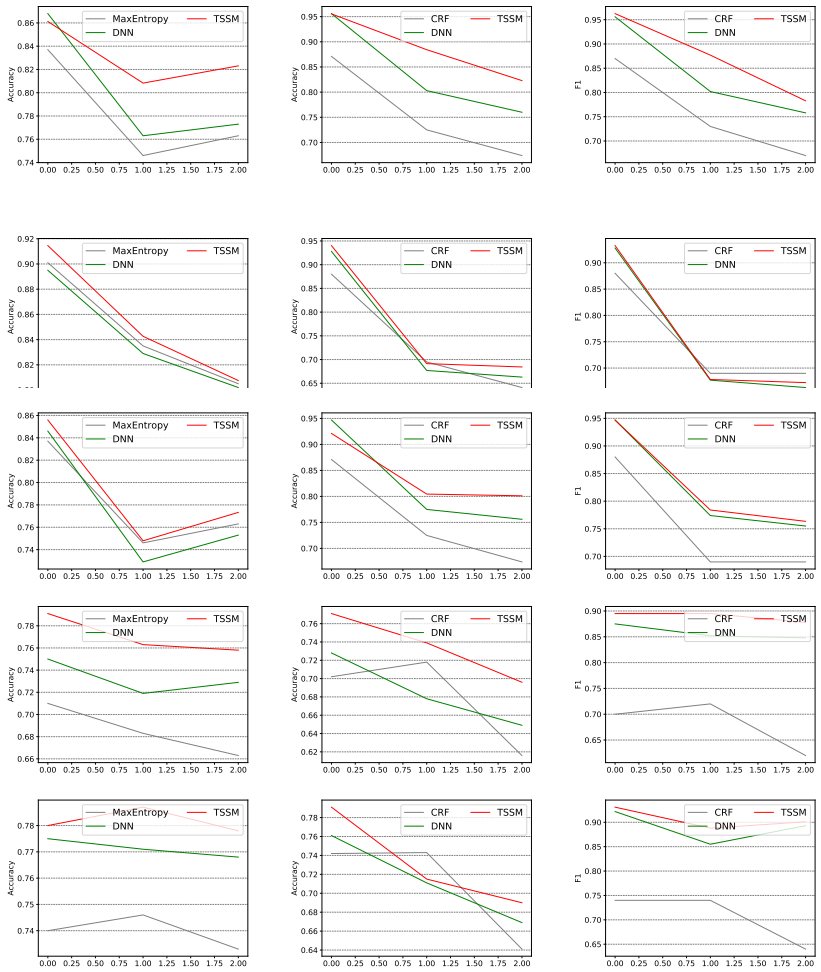

Figure 3: Performance vs the number of unseen intent/slot in the target domain testset, where 10% of the total target domain data is used. In each figure, X-axis is the number of unseen intent/slots in the target domain testset, y-axis is the performance (higher is better).

for unseen labels. With a pretrained word embeddings, the TSSM can theoretically compute semantic similarity for arbitrary label as long as the label has a word embedding. The experimental results demonstrate that the TSSM model can effectively deal with cold start labels that is not seen in the target domain training data.

4.5 Zero-shot Transfer Learning

In the extreme case, there could be no training instances in the target domain at all. In order to test the proposed TSSM model on such extreme case, we report the performances of the TSSM model when there is no taraget domain training instances. The result is shown in Table 2. The random guess performance is 0.05 for intent classification accuracy, 0.02 for slot accuracy and 0.02 for slot f1. The TSSM have an performance much better than random guess. In such case, both the Pretrained DNN, the Max Entropy model and the CRF model could not work, since all of them require target domain training instances to initialize their softmax layer.

Table 2: The zeros-shot performance of TSSM

Source and Target	Intent Accuracy	Slot Accuracy	Slot F1
nohotel2hotel	0.787	0.871	0.870
atis2hotel	0.478	0.756	0.759
noatis2atis	0.220	0.617	0.616

4.6 ABLATION EXPERIMENT

One of the improvements we have made is to use the intent bi-LSTMs output states as input features to the slot bi-LSTMs, we call this mechanism "intent2slot". In order to evaluate whether the "intent2slot" mechanism can actually improve the performance of the model, we compare the proposed TSSM with its ablation variant without "intent2slot", and the result is shown in Table 3 As shown in the result, the TSSM outperforms the TSSM-abl without "intent2slot" in most cases. The results demonstrate that the "intent2slot" mechanism helps to improve intent classification performance. The parameters in intent bi-LSTMs layer might have obtained additional supervision from the upper slot filling task. The improvements for transferring from Restaurant, Laptop, TV, Atis to Hotel is not obvious. The reason needs further experiments.

Table 3: The TSSM with/without "intent2slot"

Source and Target	Intent Accuracy		Slot Accuracy		Slot F1	
	TSSM-abl	TSSM	TSSM-abl	TSSM	TSSM-abl	TSSM
nohotel2hotel	0.949	**0.952**	0.924	0.926	0.938	0.935
atis2hotel	0.945	**0.952**	0.937	0.938	0.945	0.945
noatis2atis	0.807	**0.840**	0.891	0.896	0.899	0.888

5 CONCLUSIONS

In this paper, we introduce a deep semantic similarity network to transfer from domains with sufficient labeled data to low resource domains. The experiments illustrate that our proposed methods outperform the state-of-the-art baselines in several experimental settings.

REFERENCES

Ankur Bapna, Gokhan Tur, Dilek Hakkani-Tur, and Larry Heck. Towards zero shot frame semantic parsing for domain scaling. In *Interspeech 2017*, 2017.

Marco Baroni, Georgiana Dinu, and Germán Kruszewski. Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pp. 238–247, 2014.

Jan Buys and Jan A. Botha. Cross-lingual morphological tagging for low-resource languages. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1954–1964. Association for Computational Linguistics, 2016. doi: 10.18653/v1/P16-1184. URL http://www.aclweb.org/anthology/P16-1184.

Y. Chen, W. Y. Wang, and A. I. Rudnicky. Unsupervised induction and filling of semantic slots for spoken dialogue systems using frame-semantic parsing. In *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*, pp. 120–125, Dec 2013. doi: 10.1109/ASRU.2013.6707716.

Y. Chen, D. Hakkani-Tr, and X. He. 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)*, pp. 6045–6049, March 2016. doi: 10.1109/ICASSP.2016.7472838.

Yun-Nung Chen, William Yang Wang, Anatole Gershman, and Alexander Rudnicky. Matrix factorization with knowledge graph propagation for unsupervised spoken language understanding. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 483–494. Association for Computational Linguistics, 2015a. doi: 10.3115/v1/P15-1047. URL http://www.aclweb.org/anthology/P15-1047.

Yun-Nung Chen, William Yang Wang, and Alexander Rudnicky. Jointly modeling inter-slot relations by random walk on knowledge graphs for unsupervised spoken language understanding. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for*

Computational Linguistics: Human Language Technologies, pp. 619–629. Association for Computational Linguistics, 2015b. doi: 10.3115/v1/N15-1064. URL http://www.aclweb.org/anthology/N15-1064.

Ronan Collobert and Jason Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pp. 160–167. ACM, 2008.

Li Fei-Fei, R. Fergus, and P. Perona. One-shot learning of object categories. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(4):594–611, April 2006. ISSN 0162-8828. doi: 10.1109/TPAMI.2006.79.

E. Ferreira, B. Jabaian, and F. Lefvre. Online adaptative zero-shot learning spoken language understanding using word-embedding. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5321–5325, April 2015. doi: 10.1109/ICASSP.2015.7178987.

Anuj Kumar Goyal, Angeliki Metallinou, and Spyros Matsoukas. Fast and scalable expansion of natural language understanding functionality for intelligent agents. *CoRR*, abs/1805.01542, 2018. URL http://arxiv.org/abs/1805.01542.

S. Hahn, M. Dinarelli, C. Raymond, F. Lefevre, P. Lehnen, R. De Mori, A. Moschitti, H. Ney, and G. Riccardi. Comparing stochastic approaches to spoken language understanding in multiple languages. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(6):1569–1583, 2011. doi: 10.

Dilek Hakkani-Tr, Gokhan Tur, Asli Celikyilmaz, Yun-Nung Vivian Chen, Jianfeng Gao, Li Deng, and Ye-Yi Wang. Multi-domain joint semantic frame parsing using bi-directional rnn-lstm. ISCA, June 2016. URL https://www.microsoft.com/en-us/research/publication/multijoint/.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8): 1735–1780, 1997.

Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*, pp. 2333–2338. ACM, 2013.

Aaron Jaech, Larry P. Heck, and Mari Ostendorf. Domain adaptation of recurrent neural networks for natural language understanding. *CoRR*, abs/1604.00117, 2016. URL http://arxiv.org/abs/1604.00117.

Joo-Kyung Kim, Young-Bum Kim, Ruhi Sarikaya, and Eric Fosler-Lussier. Cross-lingual transfer learning for pos tagging without cross-lingual resources. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 2832–2838. Association for Computational Linguistics, 2017a. URL http://aclweb.org/anthology/D17-1302.

Young-Bum Kim, Karl Stratos, and Dongchan Kim. Domain attention with an ensemble of experts. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 643–653. Association for Computational Linguistics, 2017b. doi: 10.18653/v1/P17-1060. URL http://www.aclweb.org/anthology/P17-1060.

M. Korpusik, N. Schmidt, J. Drexler, S. Cyphers, and J. Glass. Data collection and language understanding of food descriptions. In *2014 IEEE Spoken Language Technology Workshop (SLT)*, pp. 560–565, Dec 2014. doi: 10.1109/SLT.2014.7078635.

Gakuto Kurata, Bing Xiang, Bowen Zhou, and Mo Yu. Leveraging sentence-level information with encoder LSTM for natural language understanding. *CoRR*, abs/1601.01530, 2016. URL http://arxiv.org/abs/1601.01530.

G. Mesnil, Y. Dauphin, K. Yao, Y. Bengio, L. Deng, D. Hakkani-Tur, X. He, L. Heck, G. Tur, D. Yu, and G. Zweig. Using recurrent neural networks for slot filling in spoken language understanding. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(3):530–539, March 2015. ISSN 2329-9290. doi: 10.1109/TASLP.2014.2383614.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 746–751, 2013.

Lili Mou, Zhao Meng, Rui Yan, Ge Li, Yan Xu, Lu Zhang, and Zhi Jin. How transferable are neural networks in nlp applications? In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 479–489. Association for Computational Linguistics, 2016. doi: 10.18653/v1/D16-1046. URL `http://www.aclweb.org/anthology/D16-1046`.

Mark Palatucci, Dean Pomerleau, Geoffrey Hinton, and Tom M. Mitchell. Zero-shot learning with semantic output codes. In *Proceedings of the 22Nd International Conference on Neural Information Processing Systems*, NIPS'09, pp. 1410–1418, USA, 2009. Curran Associates Inc. ISBN 978-1-61567-911-9. URL `http://dl.acm.org/citation.cfm?id=2984093.2984252`.

S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, Oct 2010. ISSN 1041-4347. doi: 10.1109/TKDE.2009.191.

Patti J Price. Evaluation of spoken language systems: The atis domain. In *Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania, June 24-27, 1990*, 1990.

Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. In *Proceedings of the 6th International Conference on Learning Representations*, ICLR'18, 2018.

R. Sarikaya, G. E. Hinton, and B. Ramabhadran. Deep belief nets for natural language call-routing. In *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5680–5683, May 2011. doi: 10.1109/ICASSP.2011.5947649.

Gokhan Tur and Renato De Mori. Spoken language understanding: Systems for extracting semantic information from speech. 2011.

Oriol Vinyals, Charles Blundell, Timothy P. Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. *CoRR*, abs/1606.04080, 2016. URL `http://arxiv.org/abs/1606.04080`.

Ye-Yi Wang, Li Deng, and Alex Acero. Spoken language understanding an introduction to the statistical framework. *IEEE Signal Processing Magazine*, 22/5:16–31, January 2005. URL `https://www.microsoft.com/en-us/research/publication/spoken-language-understanding-an-introduction-to-the-statistical-framework/`.

Tsung-Hsien Wen, Milica Gasic, Nikola Mrksic, Pei-Hao Su, David Vandyke, and Steve Young. Semantically conditioned lstm-based natural language generation for spoken dialogue systems. *arXiv preprint arXiv:1508.01745*, 2015.

Tsung-Hsien Wen, Milica Gašić, Nikola Mrkšić, Lina M. Rojas-Barahona, Pei-Hao Su, Stefan Ultes, David Vandyke, and Steve Young. Conditional generation and snapshot learning in neural dialogue systems. *arXiv preprint: 1606.03352*, June 2016a.

Tsung-Hsien Wen, Milica Gašić, Nikola Mrkšić, Lina Rojas-Barahona, Pei-Hao Su, David Vandyke, and Steve Young. Multi-domain neural network language generation for spoken dialogue systems. In *Proceedings of NAACL-HLT*, 2016b.

Tsung-Hsien Wen, David Vandyke, Nikola Mrkšić, Milica Gašić, Lina M. Rojas-Barahona, Pei-Hao Su, Stefan Ultes, and Steve Young. A network-based end-to-end trainable task-oriented dialogue system. *arXiv preprint: 1604.04562*, April 2016c.

Majid Yazdani and James Henderson. A model of zero-shot learning of spoken language understanding. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 244–249. Association for Computational Linguistics, 2015. doi: 10.18653/v1/D15-1027. URL http://www.aclweb.org/anthology/D15-1027.

Steve Young. Talking to machines (statistically speaking). 2002.

Tiancheng Zhao and Maxine Eskénazi. Zero-shot dialog generation with cross-domain latent actions. *CoRR*, abs/1805.04803, 2018. URL http://arxiv.org/abs/1805.04803.

Table 4: youzhe remove intent

	303:205:545			351:157:545			415:93:545		
	rm 0	rm 1	rm 2	rm 0	rm 1	rm 2	rm 0	rm 1	rm 2
TSSM	0.775	0.743	0.748	0.791	0.763	0.758	0.776	0.755	0.771
DNN	0.738	0.726	0.725	0.750	0.719	0.729	0.743	0.732	0.73
Max	0.87	0.82	0.829	0.831	0.822	0.829	0.842	0.824	0.829

Table 5: youzhe remove slot accuracy

	303:205:545			351:157:545			415:93:545		
	rm 0	rm 1	rm 2	rm 0	rm 1	rm 2	rm 0	rm 1	rm 2
TSSM	0.775	0.716	0.692	0.771	0.739	0.696	0.784	0.749	0.706
DNN	0.735	0.687	0.660	0.728	0.678	0.649	0.748	0.685	0.653
CRF	0.732	0.650	0.622	0.702	0.718	0.616	0.737	0.692	0.624

Table 6: youzhe remove slot f1

	303:205:545			351:157:545			415:93:545		
	rm 0	rm 1	rm 2	rm 0	rm 1	rm 2	rm 0	rm 1	rm 2
TSSM	0.885	0.882	0.832	0.895	0.895	0.878	0.885	0.862	0.852
DNN	0.875	0.852	0.830	0.875	0.852	0.848	0.879	0.835	0.839
CRF	0.73	0.69	0.62	0.70	0.72	0.62	0.74	0.69	0.62

Table 7: huobijijin remove intent

	268:183:476			334:117:476			374:77:476		
	rm 0	rm 1	rm 2	rm 0	rm 1	rm 2	rm 0	rm 1	rm 2
TSSM	0.768	0.794	0.778	0.777	0.787	0.783	0.78	0.787	0.778
DNN	0.765	0.772	0.768	0.777	0.769	0.772	0.775	0.771	0.768
Max	0.91	0.908	0.891	0.912	0.908	0.891	0.91	0.901	0.891

Table 8: huobijijin remove slot accuracy

	268:183:476			334:117:476			374:77:476		
	rm 0	rm 1	rm 2	rm 0	rm 1	rm 2	rm 0	rm 1	rm 2
TSSM	0.784	0.715	0.673	0.781	0.721	0.66	0.791	0.715	0.669
DNN	0.768	0.723	0.687	0.778	0.618	0.681	0.761	0.711	0.69
CRF	0.743	0.681	0.641	0.738	0.685	0.649	0.742	0.743	0.641

Table 9: huobijijin remove slot f1

	268:183:476			334:117:476			374:77:476		
	rm 0	rm 1	rm 2	rm 0	rm 1	rm 2	rm 0	rm 1	rm 2
TSSM	0.901	0.901	0.895	0.901	0.898	0.895	0.911	0.855	0.901
DNN	0.915	0.895	0.888	0.913	0.893	0.893	0.922	0.888	0.893
CRF	0.74	0.68	0.64	0.77	0.68	0.65	0.74	0.74	0.64