A Two-stage Model for Slot Filling in Low-resource Settings: Domain-agnostic Non-slot Reduction and Pretrained Contextual Embeddings

Cennet Oguz, Ngoc Thang Vu
Institute for Natural Language Processing (IMS), University of Stuttgart
cennet.oguz|thang.vu@ims.uni-stuttgart.de

Abstract

Learning-based slot filling - a key component of spoken language understanding systems - typically requires a large amount of in-domain hand-labeled data for training. In this paper, we propose a novel two-stage model architecture that can be trained with only a few in-domain hand-labeled examples. The first step is designed to remove non-slot tokens (i.e., *O* labeled tokens), as they introduce noise in the input of slot filling models. This step is domain-agnostic and therefore, can be trained by exploiting out-of-domain data. The second step identifies slot names only for slot tokens by using state-of-the-art pretrained contextual embeddings such as ELMO and BERT. We show that our approach outperforms other state-of-art systems on the SNIPS benchmark dataset.

1 Introduction

Slot filling models, which predict task-specific names (e.g. artist, time) for these slots from user utterances, are a key component of spoken language understanding (SLU) systems. Deep learning approaches (Mesnil et al., 2013; Hakkani-Tür et al., 2016; Zhang and Wang, 2016; Zhu and Yu, 2018; Chen et al., 2013; Gupta et al., 2018; Bapna et al., 2017a) for SLU involve training on a large amount of annotated training data. Likewise, multi-domain studies (Hakkani-Tür et al., 2016; Liu and Lane, 2017) that rely on deep learning methods require a large amount of data for each domain. However, slot filling is a very challenging task if only a few labeled samples are available. Therefore, this paper proposes methods to address the low-resource domain issue of slot filling.

We aim at improving performance of the slot filling task in different low-resource scenarios by exploring the effective usage of a few in-domain samples with two different scenarios: (1) if data from other domains is not possible but a few samples are available in the current domain (2) if data from other domains are available and a few samples are accessible in the current domain. We exploit domain-agnostic syntactic similarities (e.g., the main verb of a sentence cannot be a slot) to learn the conceptual differences between slot and non-slot tokens in order to dismiss non-slot tokens from the input space. Therefore, using labeled data (SLOT and O labels) across domains can improve the non-slot token reduction step in the target domain and thereby the slot name prediction step. Therefore, we propose a novel two-stage model that first reduces this noise by adding a non-slot detection step and then predicts slot names. The identified non-slots are then removed from the input space of the name prediction step. Our modeling approach is inspired by (Zhai et al., 2017; Dauphin et al., 2013; Shah et al., 2019).

We suggest using a few annotated samples as training input instead of slot descriptions and slot names as in zero-shot learning studies (Bapna et al., 2017a; Lee and Jha, 2019; Shah et al., 2019). This is for two reasons: (1) The creation of slot descriptions needs qualified linguistic expertise and is thus expensive. (2) The relationship between slot names and the corresponding tokens is not constant. To give an example, the relationship between the '*genre*' slot name and '*drama*' token is hypernymic whereas the relationship between the '*artist*' slot name and '*Tarkan*' token is instance based. Hence, it may not be valid to learn only one function to represent the different relationships between names and tokens.

As a classification algorithm, we employ Rocchio classification method (Rocchio, 1971) for labeling the tokens with their domain specific name labels after reducing the non-slot tokens from the input. Rocchio classifier is a very simple classification method that separates the inputs into centroids

Proceedings of SustaiNLP: Workshop on Simple and Efficient Natural Language Processing, pages 73–82
Online, November 20, 2020. ©2020 Association for Computational Linguistics

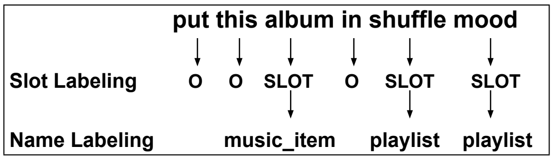

Figure 1: An example for the two-stage modeling approach for an example utterance from SNIPS.

computed as the center of mass of all vectors in the class, i.e., builds a prototype vector for each class. Decision process is simply made based on distance metrics. Because of the availability of only a small amount of data in the current domain and the semantically rich and robust presentation in contextual pretrained embeddings, we argue that Rocchio classifier is sufficient for our task. Furthermore by using this simple classification method, we show the effectiveness of the non-slot noise reduction step from the input.

2 Problem Statement

2.1 Problem Definition

We partition the slot filling task in two consecutive sub-tasks which are called *Slot Labeling* and *Name Labeling*. The *Slot Labeling* task requires to predict for each token in a sentence one of classes $S = \{O, SLOT\}$ where *SLOT* corresponds to slots whereas *O* represents non-slot tokens. The *Name Labeling* task requires to predict one label from a predefined name label set $N = \{...\}$ for a set of candidate slots. This implies that candidate slots have already been identified as $SLOT$ by *Slot Labeling* task as shown in Figure 1.

While S is shared across domains, N is domain-specific. Therefore, training data can be shared across domains for the *Slot Labeling* task, but not for the *Name Labeling* task. Thus, we run into the limited data problem for *Name Labeling*.

2.2 Evaluation

We state the evaluation of the proposed systems by computing the average of the precision and recall, i.e, F1 score, over the results of *Name Labeling* task, although the system consists of two consecutive models. In order to understand the overall performance, the average F1 scores of 7 domains are computed. Additionally, the evaluation values represent the average F1 over three random data splits.

3 Model Architecture

We define our consecutive model structure as follows: given an utterance with T tokens, first we employ *Slot Labeling* model in order to identify $SLOT$ tokens while eliminating the non-slot tokens of input utterance. Consecutively, we predict the slot name of the $SLOT$ tokens which are received from the *Slot Labeling* model. The Figure 3 illustrates the overview of the consecutive model architecture with its inputs and outputs while showing the usage of contextualized word embeddings in order to represent input tokens.

3.1 Inputs

The contextualized word representation methods, e.g., ELMo (Peters et al., 2018) and BERT (Devlin et al., 2019), use a pre-trained network over the sentence in order to produce unique embeddings based-on the current context, instead of using a single, fixed vector per word like in Word2Vec (Mikolov et al., 2013) or GloVe (Pennington et al., 2014). The pre-trained models, usually an LSTM (Hochreiter and Schmidhuber, 1997) or a Transformer (Vaswani et al., 2017) can be trained for token-level classification tasks, e.g., named entity recognition, part-of-speech, or sentence-level classifications, e.g., text classification, sentiment analysis. At the same time, they can leverage the the language modeling (Peters et al., 2018; Devlin et al., 2019) by fine-tuning (Howard and Ruder, 2018) the trained objectives on domain-specific dataset as well as they can be used as feature-based models (Peters et al., 2018; Tenney et al., 2019; Brunner et al., 2020) for the down-stream tasks. In this study, we employ feature-based BERT and ELMo for the slot-filling task in low-resource domain.

BERT uses a bidirectional transformer model which is trained on a masked language modeling task. It uses WordPiece embeddings (Wu et al., 2016) which means each word of an input represented with its sub-tokens. Thus, we use the first sub-token for representing the word as it turns out in (Devlin et al., 2019). Additionally, BERT consists of multiple successive layers, i.e., 24 layers because of preferred *BERT-large-cased model*, and each layer represents different linguistic notions of syntax or semantics (Clark et al., 2019). In order to find the focused layers on local context (Tenney et al., 2019) in these linguistic notions, the attention visualization tool (Vig, 2019) is used on randomly

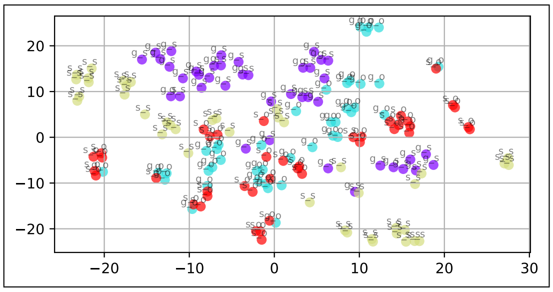

Figure 2: Two dimensional representation of ELMo vectors of randomly selected slot and non-slot token in two example domains in SNIPS dataset: GetWeather and SearchScreeningE. s_o (red) = vectors of non-slot tokens of SearchScreeningE.; s_s (green) = slot tokens of SearchScreeningE.; g_o (blue) = non-slot tokens of GetWeather; g_s (purple) = slot tokens of GetWeather.

selected samples. We select *10th*, *11th*, *12th*, and *13th* layers and concatenate hidden states of these layers in order to represent the corresponding word.

ELMo concatenates the output of two LSTM independently trained on the bidirectional language modeling task and return the hidden states for the given input sequence.

The proposed consecutive approach uses two different label sets S and N, i.e., as explained in Section 2.1, which share the same sequences per domains. We operate the contextual embeddings on given utterance with the input sequence to assign the contextual embeddings to their corresponding input tokens.

3.2 Slot Labeling

Figure 2 shows the domain-agnostic pattern between non-slot token vectors of *GetWeather* and *SearchScreeningE.*, non-slot tokens (g_o) from *GetWeather*, and non-slot tokens (s_o) from *SearchScreeningE.* show higher similarity than slot tokens from both. The *Slot Labeling* step aims to make efficient use of the existing slot labeled dataset from current and different domains in order to exploit that domain-agnostic semantic frames for the current domain. Therefore, we employ two different *Slot Labeling* models separately according to data availability. Thus, we define two common scenarios to cope with: (1) the absence of data from different domains whereas the occurrence of few labeled samples in the current domain (2) available data from different domains as well as the presence of few labeled samples in the current domain. For the first scenario, we apply Rocchio Slot Labeling whereas Neural Slot Labeling is employed as the solution of the second one.

Rocchio Slot Labeling: It is proposed for utilizing a few available labeled samples from the current domain and show the performance of non-slot reduction without any additional samples from different domain on slot and name labeling. Utilizing only a few samples to build classification model for slot labeling, we apply a Rocchio classifier that assigns to observations the label of the class of training samples whose centroid is closest to the observation.

$$\hat{y} = \underset{Y_s \in S}{\arg\min} \|\mu_s - v_i\|, \mu_s = \frac{1}{|X|} \sum_{v_i \in X} v_i \quad (1)$$

where $X = \{v_1, v_2, ..., v_n\}$, v_i represents a slot value. Thus, the Rocchio classifier is trained to map the given slot value to the slot label by using the centroid (μ_s) of the prototypes (X) of the corresponding slot label.

Neural Slot Labeling: We use this with the purpose of using available labeled data from different domains in addition to a few labeled samples of the current domain. The availability of large amount of labeled data from different domain make use of complex architecture such as neural networks. Thus, for ELMo embeddings we use the token classification model proposed by (Peters et al., 2018) whereas for BERT embeddings we implement the token classification model proposed by (Devlin et al., 2019). Thus, for the given $X = \{w_1, w_2, ..., w_T\}$ in order to predict $Y_s = (y_1, y_2, ..., y_T)$ where T is the token number of the given input and $y_i \in S$,

ELMo embeddings are used with an LSTM+CRF which is trained by maximizing the conditional log-likelihood,

$$\hat{Y_s} = \arg\max \sum_{i=1}^{T} \ln p(y_i|w_i, X) \quad (2)$$

BERT embeddings are used with a Linear layer and a following softmax function,

$$\hat{Y_s} = \arg\max \sum_{i=1}^{T} softmax(W * w_i) \quad (3)$$

The aim of the Neural Slot model is efficiently leveraging domain agnostic features of different task-oriented domains with the networks. Because the existence of available data lets us train the networks in order to find slot/non-slot tokens.

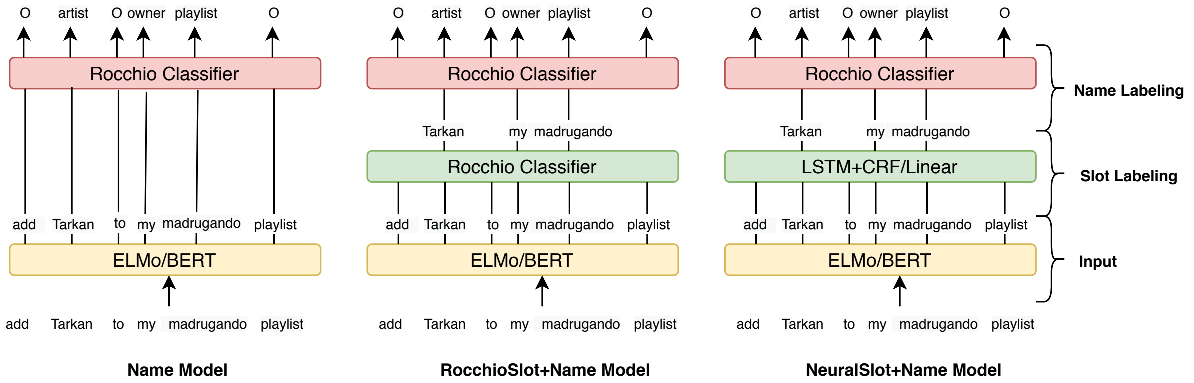

Figure 3: The overview of proposed model variations with input and output shown for an example utterance from SNIPS. *Name Model* represents the baseline without *Slot Labeling* stage and uses only a few samples from the current domain. *RocchioSlot+Name Model* employs Rocchio classifier for *Slot Labeling* stage with a few SLOT/O labeled samples from the current domain in order to reduce the non-slot tokens and then uses the same samples with Name labels to train *Name Model*. *NeuralSlot+Name Model* utilizes the out-domain SLOT/O labeled samples together with a few current domain SLOT/O labeled samples in order to train *Neural Slot Labeling* whereas it uses only the current domain samples with *Name* labels for Rocchio *Name Model*.

3.3 Name Labeling

We assume, only a few samples for the current domain is available for training a model. Thus, the absence of a huge amount of labeled data for the current domain makes it impossible for the use of neural networks. Therefore, we utilize Rocchio classifier as presented in equation 1 to map the given slot value to the name label by using the centroid of the prototypes of the corresponding name label.

4 Dataset and Experimental Setup

4.1 Resources

We utilize the SNIPS dataset (Coucke et al., 2018) as a base dataset in our experiment. SNIPS is a SLU dataset of crowd-sourced user utterances with 39 slots and 7 intents. We split SNIPS with the purpose of creating a single-domain dataset.

We create *Prototype*- and *Test*-data in order to train the models and evaluate their performance on each domain. Four *Prototype* groups are generated from SNIPS in order to investigate the performance when the number of samples increases. To accomplish this, we randomly select 10 slot samples embedded in their input sequences (complete sentences) per label in SNIPS. With the initial sample of 10 slots per label, we increment the previous set by 5 randomly-selected slot samples up to 2 times, resulting in 10, 15, and 20 sub-sample groups (10-Prototype $\subset$ 15-Prototype $\subset$ 20-Prototype). Selected slot phases represent one sample in the label space even if the token number is greater then 1. For example, *Wind of Change* consists of three tokens, however, these three tokens represents one sample. *Test* data consists of 1000 randomly selected sentences. *Prototype* and *Test* include two annotation sets, *Name* and *Slot*.

Name Set: Provides annotation for the sentences with labels such as *artist, object_name* and *O* tags for the input sequences.

Slot Set: We convert the labels in the *Name Set* to *SLOT* tags while keeping the *O* tags the same.

Auxiliary Slot Set: We utilize the slot filling dataset of different domains in order to reduce non-slot tokens by exploiting syntactic similarities between domains. For example, the verb of a sentences does not represent a slot in any domain. Therefore, instead of trying to leverage the semantic similarity between the slot tokens in different domains, we use non-slot token similarity to reduce them from the input space. We obtain this dataset with the same process that converting *Name Set* to *Slot Set*.

4.1.1 Proposed Systems

4.2 Experimental Setup

We design our experimental settings to investigate the following research questions. The first question focuses on exploring the impact of existing annotated data from different domains on the performance of the slot/non-slot classification step.

Table 1: Name F1 scores. The results of previous studies, the baseline Name and proposed RocchioSlot+Name and NeuralSlot+Name models in different sample size. The values represent the average F1 over 3 data splits. 'Domain Avg.' represents the average values across sample sizes.

	Previous Systems			BERT Name			ELMo Name			BERT RocchioSlot+Name			ELMo RocchioSlot+Name			BERT NeuralSlot+Name			ELMo NeuralSlot+Name		
Data	CT	ZAT	CDS	10	15	20	10	15	20	10	15	20	10	15	20	10	15	20	10	15	20
AddToPlaylist	74	73	**76**	45	46	47	45	46	48	62	62	63	57	59	63	66	67	67	63	65	67
PlayMusic	56	56	58	56	56	55	59	59	59	70	70	72	65	66	66	72	73	74	72	73	**74**
BookRestaurant	63	63	63	55	56	56	58	59	59	63	64	65	64	65	66	67	68	69	70	71	**73**
GetWeather	72	71	**77**	46	47	46	49	49	49	66	66	66	64	66	67	75	75	75	73	75	76
RateBook	82	83	82	73	74	73	64	65	66	90	92	92	81	82	82	95	96	**97**	89	91	92
SearchCreativeW.	62	63	65	45	46	46	45	44	44	79	81	81	81	82	82	80	87	**89**	83	84	86
SearchScreeningE.	64	64	67	38	40	40	61	61	60	50	55	55	69	71	72	65	64	65	78	78	**81**
Domain Avg.	68.2	68.0	70.1	51.1	52.1	51.8	54.4	54.7	55.0	68.5	70.0	70.5	68.7	70.1	71.1	74.2	75.7	76.5	75.4	76.7	**78.4**

Table 2: F1 scores for BERT slot/non-slot classification.

	Rocchio Slot			Neural Slot		
Domain	10	15	20	10	15	20
AddToPlaylist	93.3	93.3	93.7	97.3	98.2	97.5
PlayMusic	93.7	94.5	94.4	98.1	98.8	98.8
BookRestaurant	92.2	92.1	91.8	98.5	99.0	99.2
GetWeather	94.3	94.3	94.2	98.9	99.0	98.8
RateBook	94.7	96.5	96.0	98.7	98.6	98.7
SearchCreativeW.	96.4	96.4	96.3	96.4	97.4	98.4
SearchScreeningE.	90.7	91.8	91.7	99.1	99.1	99.1

Table 3: F1 scores for ELMo slot/non-slot classification.

	Rocchio Slot			Neural Slot		
Domain	10	15	20	10	15	20
AddToPlaylist	91.1	91.6	91.7	94.8	95.7	96.6
PlayMusic	91.1	91.1	90.9	97.1	97.2	97.9
BookRestaurant	88.5	87.7	87.9	96.6	98.2	98.5
GetWeather	92.7	92.8	93.0	98.0	98.7	98.7
RateBook	93.1	93.4	93.5	96.8	97.4	97.8
SearchCreativeW.	96.8	96.4	96.4	97.2	97.7	98.0
SearchScreeningE.	90.7	91.1	91.3	97.8	98.0	98.4

We assume that we have sufficient training data for different domains but not for a target domain. The second question aims at exploring the effectiveness of exploiting *Prototypes* with respect to example sizes. The third question focuses on the comparison between the two contextual embeddings, ELMO and BERT.

4.2.1 Baseline

Name Model: In order to examine the effect of low-resource domain in slot tagging, we train the *Name Labeling* by using *Prototype Name Set*. Then, we test the model with *Test Name Set* that includes 1000 samples with corresponding *Name* labels *N*. We use contextual embeddings, either ELMO or BERT, as input representation, resulting in two main baseline models without using the non-slot reduction step. We employ *Name Model* to show to performance of a few samples.

RocchioSlot+Name Model: We use *Prototype Slot Set* to make use of Rocchio Slot Labeling whereas we utilize *Prototype Name Set* to train Name Model. Here, we aim to understand the efficiency of non-slot reduction with only a few current domain samples. The *Slot Set* from *Prototype* that includes only a few samples with the corresponding *Slot* labels *S* is used to train the *Rocchio Slot Labeling* model. This trained Rocchio Slot Labeling (*RocchioSlot*) model then reduces the non-slot tokens from the input of the *Name Labeling* model. The *Name Labeling* model then predicts the token labels *N*.

NeuralSlot+Name Model: The process of using this model is identical to *RocchioSlot+Name Model*. The only difference is that we add *Slot Sets* from other domains - *Auxiliary Slot Set*- and train the *Neural Slot Labeling* model in order to analyze the impact of out-domain samples on the performance of non-slot reduction and *Name Model* . An example of this would be the usage of annotated "AddToPlaylist" and "GetWeather" domains data converting the labels to *SLOT* labels for "PlayMusic" in order to train the *Neural Slot Labeling* model.

5 Results

Non-slot/slot Classification Results Table 2 and 3 show the results from the *Rocchio Slot Labeling* and *Neural Slot Labeling* models for both BERT and ELMo. According to the overall results, we strongly claim that *SLOT* label for a token is a domain-agnostic feature. Moreover, BERT embeddings show better performance then ELMo on *Slot Labeling* task in both model setups.

Two-stage Slot Name Labeling Results Table 1 shows that the proposed models outperforms the baseline across domains and sample sizes. It is apparent that the increase of samples sizes largely improves F1 score per domain. As can be seen in *Domain Avg.*, our non-slot reduction models *RocchioSlot+Name* and *NeuralSlot+Name* outperform the baseline *Name Model* with $> 20\%$. In addition, by comparing *NeuralSlot+Name* and *RocchioSlot+Name*, we see that *NeuralSlot+Name* model results in an $> 6\%$ percent increase in the average performance.

Impact of Different Contextualized Embeddings ELMo and BERT have comparable performance, with ELMo slightly better on most tasks, e.g., as expected after the study of (Tenney et al., 2019), but the Transformer scoring higher on *RateBook* and *SearchCreativeW.* consistently with all the models.

Comparisons with State-of-art Systems We compared our systems with the three following studies: (1) Zero-shot Adaptive Transfer (ZAT) (Lee and Jha, 2019) that used condition slot filling on slot descriptions with hierarchical six LSTM and CRF layers; (2) Concept Tagger (CT) (Bapna et al., 2017b) by exploiting multi-task bidirectional stacked LSTM ; (3) Cross Domain Slot filling study (CDS) (Shah et al., 2019) that used a conditional sequence tagging model by utilizing BiGRU-BiLSTM model. The results of previous studies are taken from (Shah et al., 2019)). Table 1 demonstrates that even though the previous systems use a large amount of data with the neural networks, *RocchioSlot+Name* outperforms the best performance of previous system (CDS) with up to 1% with 20 training examples, whereas the *NeuralSlot+Name* model outperforms them with up to 8.3% improvement.

6 Qualitative Analysis

We analyzed the results on individual slots by comparing them according to contextualized embeddings and proposed models. We observed that BERT shows consistent lower results for the tokens like city, state from *BookResteurant*, and *location_name*, *object_location_type* from *SearchScreeningE* whereas it outperforms ELMo for proper name detection like *object_name* from *RateBook* and *SearchCreativeW.* domains.

The wrong predictions of Name Labeling, e.g., false-positive rates of names (e.g., object_select, cuisine, spatial_relation) for O label, draw the attention. An extreme difference between low precision and relatively high recall is observed. However, the precision results are drastically improved when RocchioSlot+ and NeuralSlot+Name models are employed. For example, *RateBook* domain's slot *object_select* has 0.41 precision with Name Model whereas the precision of it is 0.69 and 0.93 with RocchioSlot+ and NeuralSlot+Name models respectively.

On the other hand, when the *timeRange* label of *GetWeather* is reviewed, RocchioSlot+Name as well as Name Model failed. Due to leak of non-slot tokens, *timeRange* values labeled as 'O'. RocchioSlot fails for labeling the values (e.g., eleven months from now) of *timeRange* with *S*, because it is a clustering-based method and is not able to capture the sequential dependencies. NeuralSlot+Name models, however, shows significant increases. The comparison of the results from both models indicates that the wrong predictions of the 'O' label drastically reduced with NeuralSlot+Name model.

Similar proper nouns, e.g., *album* and *track*, in the same domain denote the weakness of the proposed systems. NeuralSlot+Name model is not able to distinguish similar proper nouns. For example, the highest false-negative rate for *album* is *track* while it is *album* for *track*.

7 Related Work

7.1 Low-resource Domain in NLP

Typically in NLP, the domain is meant to refer to some coherent type of dataset that related to the underlying linguistic distribution (Ramponi and Plank, 2020). When the linguistic distribution between target and source domain differ, the performance drops on the target domain. Therefore, hand-labeled samples are needed for many NLP applications even though they are expensive to create and often not available for low-resource languages or domains. Many studies have recently been proposed to tackle the low-resource issue by using different approaches such as transfer learning for domain adaptation (Daume III and Marcu, 2006; Pan and Yang, 2009), and multi-task learning (Peng and Dredze, 2017a). Here, we review the slot filling like sequence labeling studies such as part-of-speech tagging (POS) and named entity recognition (NER) within domain adaptation and multi-task learning.

The domain adaptation approach is used to transfer the domain-general feature space from source tasks as "prior knowledge" to the target task in order to overcome the hand-labeled data scarcity (Blitzer et al., 2006; Daume III and Marcu, 2006; Ramponi and Plank, 2020). For POS tagging, Jiang and Zhai (2007) propose a supervised instance weighting technique with or without labeled instances in target domain, whereas Kann et al. (2018) use character-level and subword-level supervision. However, Han and Eisenstein (2019) demonstrate unsupervised multi-task learning with the domain-adaptive fine-tuning method by utilizing contextualized word embeddings for the new domains. Similarly, NER is a sequence labeling task that is often addressed by domain adaptation and multi-task learning because of the low-resource domain. But, most of the NER tasks consist of different label spaces. Jia et al. (2019) use cross-domain language modeling for performing cross-task knowledge transfer by extracting knowledge of domain differences from raw text, while Peng and Dredze (2017b) utilize multi-task learning approach for shared representations in multiple tasks simultaneously to have better generalize for domain adaptation.

As examined here, most existing work in NLP considers the low-resource issue as a problem of shared feature spaces. The main consideration is always augmenting the most similar feature intersection of source and target domains and use this feature space to improve the low-resource target domain (Daumé III, 2009; Ruder and Plank, 2017; Ramponi and Plank, 2020).

7.2 Low-resource Domain in Slot Filling

In a broader sense, two ways of training model have often been applied to slot filling in low-resource domain scenario: (1) use a multi-task learning method (Jaech et al., 2016a; Bingel and Søgaard, 2017) (2) train a model that performs well across domains using domain adaptation or transfer learning techniques e.g., based on external memory (Peng and Yao, 2015), ranking loss (Vu et al., 2016), encoder (Kurata et al., 2016), attention (Zhu and Yu, 2017), multi-task modeling (Jaech et al., 2016b), adversarial training (Kim et al., 2017), pointer networks (Zhai et al., 2017) have recently been proposed. These methods, however, still require a substantial amount of data for adaptation. Additionally, Louvan and Magnini (2018) propose to joint learning with NER as an auxiliary task through a multi-task learning setup and show improvement in slot filling with low-resource scenarios.

Another direction relies on zero-shot learning approaches, i.e., learning method with label descriptions or label names, which have recently been popular in slot filling task. Zero-shot learning (Socher et al., 2013) is a classification setup in learning systems, where the model predict samples from classes that were not seen during training at test time. Zero-shot slot filling, i,e., either relies on slot names or slot descriptions, has been influenced the studies of the domain scaling problem for slots prediction. (Bapna et al., 2017b) leverage the encoding of the slot names and descriptions within a multi-task deep learned slot filling model, to align slots across domains with shared feature extraction. Likewise, (Lee and Jha, 2019) propose a zero-shot adaptive transfer method for slot tagging that utilizes the slot description for transferring reusable concepts across domains for eliminating the need of labeled examples for transferring reusable concepts whereas (Shah et al., 2019) add the a target domain samples to slot descriptions for conveying the domain-agnostic concepts between the intents.

8 Conclusion and Future Work

We propose a novel two-stage model for slot filling in low-resource domains. Our results demonstrate the importance of non-slot token reduction on slot filling with resource constraints by using a simple classification method. Furthermore, the benefit of employing slot filling data from other different domains for non-slot reduction is demonstrated. In addition, increasing sample sizes for the *Prototypes* shows significant improvements. Base on our findings, future usage of multi-domain or limited data could be effective in improving slot filling methods from a non-slot reduction perspective. Additionally, the outcomes of the multi-domain data usage in our study contributes a new perspective in supervised domain adaptation and generalization studies.

References

Ankur Bapna, Gokhan Tür, Dilek Hakkani-Tür, and Larry Heck. 2017a. Sequential dialogue context modeling for spoken language understanding. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 103–114.

Ankur Bapna, Gokhan Tür, Dilek Hakkani-Tür, and Larry Heck. 2017b. Towards zero-shot frame semantic parsing for domain scaling. *Proc. Interspeech 2017*, pages 2476–2480.

Joachim Bingel and Anders Søgaard. 2017. Identifying beneficial task relations for multi-task learning in deep neural networks. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 164–169, Valencia, Spain. Association for Computational Linguistics.

John Blitzer, Ryan McDonald, and Fernando Pereira. 2006. Domain adaptation with structural correspondence learning. In *Proceedings of the 2006 conference on empirical methods in natural language processing*, pages 120–128. Association for Computational Linguistics.

Gino Brunner, Yang Liu, Damian Pascual Ortiz, Oliver Richter, Massimiliano Ciaramita, and Roger Wattenhofer. 2020. On identifiability in transformers.

Yun-Nung Chen, William Yang Wang, and Alexander I Rudnicky. 2013. 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*, pages 120–125. IEEE.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D Manning. 2019. What does bert look at? an analysis of bert's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286.

Alice Coucke, Alaa Saade, Adrien Ball, Théodore Bluche, Alexandre Caulier, David Leroy, Clément Doumouro, Thibault Gisselbrecht, Francesco Caltagirone, Thibaut Lavril, et al. 2018. Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces. *arXiv preprint arXiv:1805.10190*.

Hal Daumé III. 2009. Frustratingly easy domain adaptation. *arXiv preprint arXiv:0907.1815*.

Hal Daume III and Daniel Marcu. 2006. Domain adaptation for statistical classifiers. *Journal of artificial Intelligence research*, 26:101–126.

Yann N Dauphin, Gokhan Tur, Dilek Hakkani-Tur, and Larry Heck. 2013. Zero-shot learning for semantic utterance classification. *arXiv preprint arXiv:1401.0509*.

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.

Raghav Gupta, Abhinav Rastogi, and Dilek Hakkani-Tür. 2018. An efficient approach to encoding context for spoken language understanding. *Proc. Interspeech 2018*, pages 3469–3473.

Dilek Hakkani-Tür, Gökhan Tür, Asli Celikyilmaz, Yun-Nung Chen, Jianfeng Gao, Li Deng, and Ye-Yi Wang. 2016. Multi-domain joint semantic frame parsing using bi-directional rnn-lstm. In *Interspeech*, pages 715–719.

Xiaochuang Han and Jacob Eisenstein. 2019. Unsupervised domain adaptation of contextualized embeddings for sequence labeling. 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 4229–4239.

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

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339.

Aaron Jaech, Larry Heck, and Mari Ostendorf. 2016a. Domain adaptation of recurrent neural networks for natural language understanding. *Interspeech 2016*, pages 690–694.

Aaron Jaech, Larry Heck, and Mari Ostendorf. 2016b. Domain adaptation of recurrent neural networks for natural language understanding. *arXiv preprint arXiv:1604.00117*.

Chen Jia, Xiaobo Liang, and Yue Zhang. 2019. Cross-domain NER using cross-domain language modeling. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2464–2474, Florence, Italy. Association for Computational Linguistics.

Jing Jiang and ChengXiang Zhai. 2007. Instance weighting for domain adaptation in NLP. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 264–271, Prague, Czech Republic. Association for Computational Linguistics.

Katharina Kann, Johannes Bjerva, Isabelle Augenstein, Barbara Plank, and Anders Søgaard. 2018. Character-level supervision for low-resource pos tagging. In *Proceedings of the Workshop on Deep Learning Approaches for Low-Resource NLP*, pages 1–11.

Young-Bum Kim, Karl Stratos, and Dongchan Kim. 2017. Adversarial adaptation of synthetic or stale data. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1297–1307.

Gakuto Kurata, Bing Xiang, Bowen Zhou, and Mo Yu. 2016. Leveraging sentence-level information with encoder lstm for semantic slot filling. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2077–2083.

Sungjin Lee and Rahul Jha. 2019. Zero-shot adaptive transfer for conversational language understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6642–6649.

Bing Liu and Ian Lane. 2017. Multi-domain adversarial learning for slot filling in spoken language understanding. *arXiv preprint arXiv:1711.11310*.

Samuel Louvan and Bernardo Magnini. 2018. Exploring named entity recognition as an auxiliary task for slot filling in conversational language understanding. In *Proceedings of the 2018 EMNLP Workshop SCAI: The 2nd International Workshop on Search-Oriented Conversational AI*, pages 74–80.

Grégoire Mesnil, Xiaodong He, Li Deng, and Yoshua Bengio. 2013. Investigation of recurrent-neural-network architectures and learning methods for spoken language understanding. In *Interspeech*, pages 3771–3775.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Sinno Jialin Pan and Qiang Yang. 2009. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359.

Baolin Peng and Kaisheng Yao. 2015. Recurrent neural networks with external memory for language understanding. *arXiv preprint arXiv:1506.00195*.

Nanyun Peng and Mark Dredze. 2017a. Multi-task domain adaptation for sequence tagging. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 91–100.

Nanyun Peng and Mark Dredze. 2017b. Multi-task domain adaptation for sequence tagging. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 91–100, Vancouver, Canada. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. 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 2227–2237.

Alan Ramponi and Barbara Plank. 2020. Neural unsupervised domain adaptation in nlp—a survey. *arXiv preprint arXiv:2006.00632*.

Joseph Rocchio. 1971. Relevance feedback in information retrieval. *The Smart retrieval system-experiments in automatic document processing*, pages 313–323.

Sebastian Ruder and Barbara Plank. 2017. Learning to select data for transfer learning with bayesian optimization. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 372–382.

Darsh J Shah, Raghav Gupta, Amir A Fayazi, and Dilek Hakkani-Tur. 2019. Robust zero-shot cross-domain slot filling with example values. *arXiv preprint arXiv:1906.06870*.

Richard Socher, Milind Ganjoo, Christopher D Manning, and Andrew Ng. 2013. Zero-shot learning through cross-modal transfer. In *Advances in neural information processing systems*, pages 935–943.

Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R Thomas McCoy, Najoung Kim, Benjamin Van Durme, Samuel R Bowman, Dipanjan Das, et al. 2019. What do you learn from context? probing for sentence structure in contextualized word representations. *arXiv preprint arXiv:1905.06316*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Jesse Vig. 2019. A multiscale visualization of attention in the transformer model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 37–42.

Ngoc Thang Vu, Pankaj Gupta, Heike Adel, and Hinrich Schütze. 2016. Bi-directional recurrent neural network with ranking loss for spoken language understanding. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6060–6064. IEEE.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Feifei Zhai, Saloni Potdar, Bing Xiang, and Bowen Zhou. 2017. Neural models for sequence chunking. In *Thirty-First AAAI Conference on Artificial Intelligence*.

Xiaodong Zhang and Houfeng Wang. 2016. A joint model of intent determination and slot filling for spoken language understanding. In *IJCAI*, volume 16, pages 2993–2999.

Su Zhu and Kai Yu. 2017. Encoder-decoder with focus-mechanism for sequence labelling based spoken language understanding. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5675–5679. IEEE.

Su Zhu and Kai Yu. 2018. Concept transfer learning for adaptive language understanding. In *Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue*, pages 391–399.