ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 978-1-6654-0540-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICASSP43922.2022.9746890

EXPLORING TRANSFERABILITY MEASURES AND DOMAIN SELECTION IN CROSS-DOMAIN SLOT FILLING

Xin-Chun Li[⋆] *Yan-Jia Wang*[⋆] *Le Gan*[⋆] *De-Chuan Zhan*[⋆]

[⋆] State Key Laboratory for Novel Software Technology, Nanjing University

ABSTRACT

As an essential task for natural language understanding, slot filling aims to identify the contiguous spans of specific slots in an utterance. In real-world applications, the labeling costs of utterances may be expensive, and transfer learning techniques have been developed to ease this problem. However, cross-domain slot filling could significantly suffer from negative transfer due to non-targeted or zero-shot slots. Originally, this paper explores several ways to measure transferability across slot filling domains and finds that the shared slot number could serve as an efficient and effective estimator. First, this frustratingly easy measure requires no training data and is efficient to calculate. Second, it guides us heuristically select source domains that contain more shared slots with the target domain, which obtains SOTA results on Snips benchmark. Third, a dynamic transfer procedure based on this estimator clearly shows the negative transfer in cross-domain slot filling. We finally explore a source-free scene that we could only obtain black-box source models and propose to weight source domains based on prediction entropy.

***Index Terms*—** slot filling, cross-domain, transferability, shared slots, negative transfer

1. INTRODUCTION

Understanding the natural language in some real-world AI applications such as smart speakers (e.g., Amazon Alexa) is necessary. Slot filling [1, 2, 3, 4] aims to detect critical spans of words in user utterances and identify which entity it belongs to, i.e., the slot type. There are two main kinds of strategies to accomplish slot filling. The first one views slot filling as a single sequential labeling task that assigns each token a "slot-combined BIO" tag (e.g., "B-playlist", "I-playlist") [1, 2, 5, 6]. The second one decomposes slot filling into a coarse-to-fine process. First, it detects possible slot spans via sequence labeling with tags being "B", "I", or "O". Then, it predicts the specific slot types for detected spans via classification [4, 7].

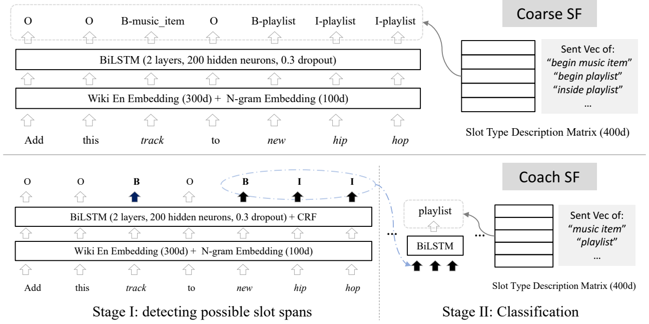

Coarse SF
O
O
B-music_item
O
B-playlist
I-playlist
I-playlist
BiLSTM (2 layers, 200 hidden neurons, 0.3 dropout)
Wiki En Embedding (300d) + N-gram Embedding (100d)
Add
this
track
to
new
hip
hop
Sent Vec of:
"begin music item"
"begin playlist"
"inside playlist"
...
Slot Type Description Matrix (400d)
Coach SF
O
O
B
O
B
I
I
BiLSTM (2 layers, 200 hidden neurons, 0.3 dropout) + CRF
Wiki En Embedding (300d) + N-gram Embedding (100d)
Add
this
track
to
new
hip
hop
playlist
BiLSTM
...
...
Sent Vec of:
"music item"
"playlist"
...
Slot Type Description Matrix (400d)
Stage I: detecting possible slot spans
Stage II: Classification

Fig. 1. Illustration of Coarse SF and Coach SF. Coach SF decomposes slot filling into two stages. To deal with zero-shot slot types, the hidden outputs of each token/entity are classified based on inner products with a slot type description matrix rather than the commonly used linear classification layer.

We denote these two strategies as "Coarse SF" and "Coach SF", respectively. The illustrations are shown in Fig. 1.

Supervised slot filling needs amounts of training data which is expensive to collect and label. Cross-domain slot filling (CDSF) [4, 7, 8, 9] has been investigated to ease the data scarcity problem. However, the data heterogeneity leads to a challenge that domains have extremely various slot types. As shown at the top of Fig. 2, there are 7 domains and 39 slot types in Snips [10] benchmark. Each row shows the distribution of the slots in the corresponding domain, and the circle size shows the occurrence frequency of this slot. Clearly, the domain gaps are significant. For example, if we want to transfer the knowledge from domain "AddToPlaylist" to "PlayMusic" (abbreviated as "ATP" and "PM"), there are only 3 shared slot types. Worsely, some domains could even have no common slot types (e.g., "ATP" and "GW"). *Intuitively, the non-targeted slot types in the source domain may lead to negative transfer because they do not occur in the target domain, and the zero-shot slot types in the target domain may still be hard to identify because they are unseen in source domains.* In this paper, we first explore whether this intuitive assumption holds via comparing several domain transferability measures. We find simply calculating the shared slot numbers could be an ideal estimator. Then, we explore several advantages of this frustratingly easy estimator.

Supported by National Natural Science Foundation of China (Grant No. 41901270), NSFC-NRF Joint Research Project under Grant 61861146001, and Natural Science Foundation of Jiangsu Province (Grant No. BK20190296). De-Chuan Zhan is the corresponding author. Email: zhandc@nju.edu.cn

Authorized licensed use limited to: Nanjing University. Downloaded on January 15,2024 at 11:35:20 UTC from IEEE Xplore. Restrictions apply.

2. RELATED WORKS

Existing CDSF works focus on handling zero-shot slot types [3, 4, 7]. Resorting to external slot descriptions to obtain a semantic concept tagger [3] or introducing explicit alignment of slots and example values [11] are both effective solutions. Coach [4] decomposes CDSF into two stages, where the first stage is more transferable because it only searches possible entity spans and does not need to identify the slot types. CZSL-Adv [7] incrementally introduces contrastive learning and adversarial attacks to improve performances. Different transferability measures have been explored in supervised classification [12], multi-task learning [13], and pre-trained language models [14]. Measuring transferability is meaningful and useful to reuse models effectively even faced with domain gaps [15, 16]. *Existing CDSF works do not explore the negative transfer phenomenon brought by non-targeted slot types, let alone selecting appropriate source domains.* The most similar work to ours is the slot transferability measure [17], which evaluates the transferability from a source slot to a target slot in CDSL. Although it has been verified useful to help select source slots that are more transferable to a target slot, the slots are not independent because they may often co-occur in utterances. Hence, we explore the transferability at the domain level, which could help us select appropriate source domains.

3. DOMAIN TRANSFERABILITY MEASURES

Snips Benchmark Snips [10][1] contains 7 domains and 39 slot types. Each domain contains approximately 2000 training utterances. Each utterance has 2.6 slots on average. The slot distribution across 7 domains is shown in Fig. 2. For CDSF, existing works use all data from 6 source domains for training, and the left one as target domain. They explore neither the negative transfer phenomenon nor domain selection. Additionally, they do not explore pairwise domain transfer procedures. In this section, we explore this on Snips and introduce some measures for *Domain Transferability* (DT).

Experimental Details We follow the experiments in [4, 7]. We use both word-level [18] and character-level [19] embeddings to obtain 400d vectors for tokens and slot descriptions. We use a two-layer BiLSTM with 200 hidden neurons. In Coarse SF, we classify each token's hidden outputs via calculating the inner products with the slot description matrix. We try adding a CRF layer, while we obtain worse performances. However, in Coach SF, we find the CRF layer useful and add it behind the BiLSTM to obtain the BIO tags. Then, the detected spans will be aggregated via another BiLSTM for slot type classification. The network architectures can be found in Fig. 1. We use the first 500 utterances in target domain for validation. We utilize seqeval[2] to calculate slot F1 as the performance criterion. More details could be found in [4].

Oracle DT via Cross-Domain Performance (DT-CDP) We first explore the pairwise transfer performances. Specifically, we denote $\text{Trf}(S \rightarrow T) = F1_{x,y\sim\mathcal{T}_{xy}}(y, x; \theta_S)$ as the performance of transferring model trained on domain "S" to "T". θ_S is the source model. $\mathcal{T}_{xy}$ is the target data distribution. x and y are sequences of tokens and tags. This is the most concerned transferability measure in transfer learning [12], which is often used as oracle values to evaluate other approximate measures. The pairwise F1 scores of Coarse SF and Coach SF are listed in Fig. 2 (a) and (b), respectively. The diagonal shows the in-domain F1s as references, where we use the 500 validation utterances in target domain for training and the others for evaluation. Rows are sources and columns are targets. *We note that most of the transferability values are small, which verifies that large gaps really exist among domains.* Additionally, the results of Coarse and Coach SF differ a little, while the calculated results enjoy a large Spearman correlation (i.e., 0.811).

DT via Extending STM (DT-STM) Slot Transferability Measure (STM) [17] estimates transferability among slots. First, it collects utterances to extract slot value representations $\Omega_{v\star}$ (i.e., the embeddings of tokens belong to specific slot types) and slot context representations $\Omega_{c\star}$ (i.e., the embeddings of tokens around specific slot types). $\star \in \{s, t\}$ denotes pairwise slots. Then, it calculates the value and context MMD [20] as $d_v = \text{MMD}(\Omega_{vs}, \Omega_{vt})$ and $d_c = \text{MMD}(\Omega_{cs}, \Omega_{ct})$. The transferability between slots is defined as $\text{STM}(s, t) = 1.0 - \tanh\left(\frac{(1+\beta^2)d_v d_c}{\beta^2 d_v + d_c}\right)$, where β weights the value and context similarities. This measure only considers slot transferability, and we extend it to calculate domain transferability. Given source and target slots, denoted as $\{s_i\}_{i=1}^m$ and $\{t_j\}_{j=1}^n$, we formulate the domain transferability as an optimal transport problem, i.e., $\text{Trf}(S, T) = \max_{C\geq 0} \sum_{i,j} C_{i,j}\text{STM}(i, j), s.t. \sum_{j=1}^n C_{ij} = \frac{1}{m}, \sum_{i=1}^m C_{ij} = \frac{1}{n}$, where $C \in \mathcal{R}^{m\times n}$ is the transport matrix. For fast evaluation, we use the relaxed approximation in [21]. The evaluated results are shown in Fig. 2 (c).

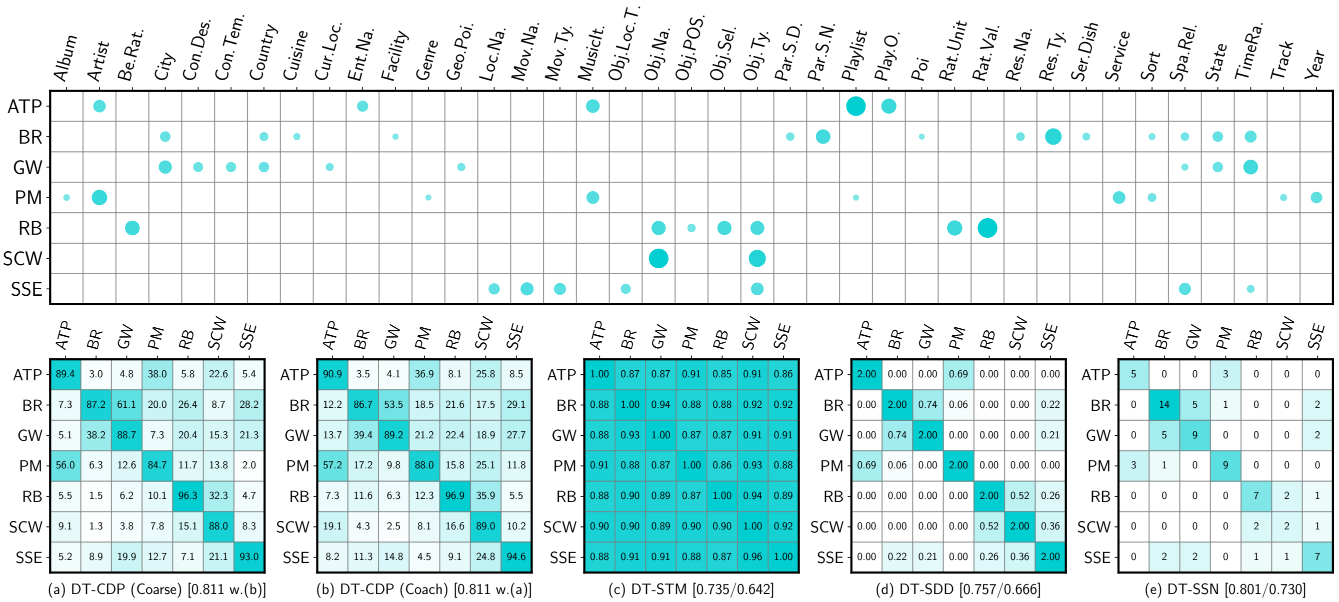
DT via Slot Distribution Discrepancy (DT-SDD) As shown in Fig. 1, the slot distributions among domains are extremely distinct. Previous works [12] declare that conditional entropy of label distributions is directly related to the loss of the transferred model. Hence, we try to measure the transferability via the slot distribution discrepancy. Specifically, we obtain the source and target slot distributions P_S, P_T via normalizing the occurrence frequencies of all slots, i.e., $P_{\star,i} = \frac{\#s_i}{\sum_j \#s_j}, \star \in \{S, T\}$. Then we use the slot distribution discrepancy to calculate the transferability measure, i.e., $\text{Trf}(S, T) = 2.0 - |P_S - P_T|_1$. The measured results are shown in Fig. 2 (d).

DT via Shared Slot Number (DT-SSN) We further explore a frustratingly easy way of calculating transferability with only the access of source and target slot types. We denote

[1] https://github.com/zliucr/coach

[2] https://github.com/chakki-works/seqeval

Authorized licensed use limited to: Nanjing University. Downloaded on January 15,2024 at 11:35:20 UTC from IEEE Xplore. Restrictions apply.



	ATP	BR	GW	PM	RB	SCW	SSE
ATP	89.4	3.0	4.8	38.0	5.8	22.6	5.4
BR	7.3	87.2	61.1	20.0	26.4	8.7	28.2
GW	5.1	38.2	88.7	7.3	20.4	15.3	21.3
PM	56.0	6.3	12.6	84.7	11.7	13.8	2.0
RB	5.5	1.5	6.2	10.1	96.3	32.3	4.7
SCW	9.1	1.3	3.8	7.8	15.1	88.0	8.3
SSE	5.2	8.9	19.9	12.7	7.1	21.1	93.0

(a) DT-CDP (Coarse) [0.811 w.(b)]

	ATP	BR	GW	PM	RB	SCW	SSE
ATP	90.9	3.5	4.1	36.9	8.1	25.8	8.5
BR	12.2	86.7	53.5	18.5	21.6	17.5	29.1
GW	13.7	39.4	89.2	21.2	22.4	18.9	27.7
PM	57.2	17.2	9.8	88.0	15.8	25.1	11.8
RB	7.3	11.6	6.3	12.3	96.9	35.9	5.5
SCW	19.1	4.3	2.5	8.1	16.6	89.0	10.2
SSE	8.2	11.3	14.8	4.5	9.1	24.8	94.6

(b) DT-CDP (Coach) [0.811 w.(a)]

	ATP	BR	GW	PM	RB	SCW	SSE
ATP	1.00	0.87	0.87	0.91	0.85	0.91	0.86
BR	0.88	1.00	0.94	0.88	0.88	0.92	0.92
GW	0.88	0.93	1.00	0.87	0.87	0.91	0.91
PM	0.91	0.88	0.87	1.00	0.86	0.93	0.88
RB	0.88	0.90	0.89	0.87	1.00	0.94	0.89
SCW	0.90	0.90	0.89	0.90	0.90	1.00	0.92
SSE	0.88	0.91	0.91	0.88	0.87	0.96	1.00

(c) DT-STM [0.735/0.642]

	ATP	BR	GW	PM	RB	SCW	SSE
ATP	2.00	0.00	0.00	0.69	0.00	0.00	0.00
BR	0.00	2.00	0.74	0.06	0.00	0.00	0.22
GW	0.00	0.74	2.00	0.00	0.00	0.00	0.21
PM	0.69	0.06	0.00	2.00	0.00	0.00	0.00
RB	0.00	0.00	0.00	0.00	2.00	0.52	0.26
SCW	0.00	0.00	0.00	0.00	0.52	2.00	0.36
SSE	0.00	0.22	0.21	0.00	0.26	0.36	2.00

(d) DT-SDD [0.757/0.666]

	ATP	BR	GW	PM	RB	SCW	SSE
ATP	5	0	0	3	0	0	0
BR	0	14	5	1	0	0	2
GW	0	5	9	0	0	0	2
PM	3	1	0	9	0	0	0
RB	0	0	0	0	7	2	1
SCW	0	0	0	0	2	2	1
SSE	0	2	2	0	1	1	7

(e) DT-SSN [0.801/0.730]

Fig. 2. Top: the slot type distributions across Snips [10] domains. The rows and columns are abbreviations of domains and slot types. Bottom: several Domain Transferability (DT) measures. The values in "[]" denote the average Spearman correlations.

$C_S = \{s_i\}_{i=1}^m$ and $C_T = \{t_j\}_{j=1}^n$ as the slot type sets of two domains. We count the shared slot numbers across domains, i.e., $\text{Trf}(S,T) = |C_S \cap C_T|$, as the simple estimator. The counts on Snips could be found in Fig. 2 (e). First, this measure is very efficient to estimate with the only need to know the source and target slot types, while the aforementioned ways need to access the slot distributions (DT-SDD) or user utterances (DT-STM). Second, we calculate the correlations with the oracle transferability. We calculate the rank-based correlations, i.e., Spearman correlation, of the estimated results in (c-e) with (a-b), respectively. Specifically, we calculate Spearman correlation for each target domain respectively (corresponding columns in Fig. 2 (a-e)) and average them as the overall correlation, which is shown in the "[]" (two numbers in (c-e) show the correlations with (a/b)). We excitingly observe that DT-SSN could better reflect the oracle ones. Hence, *DT-SSN is an efficient and effective transferability measure in CDSF*. We also extend this to consider an asymmetric estimator via dividing the total number of source slot types, i.e., $\text{Trf}(S \to T) = \frac{|C_S \cap C_T|}{|C_S|}$. However, this leads to a lower Spearman correlation compared with DT-SSN.

4. ADVANTAGES OF DT-SSN

The SOTA results via Selecting Source Domains Given a target domain, existing CDSF methods [4, 7] take the other 6 domains as sources without filtering irrelated domains. As aforementioned, the non-targeted slots in some domains could lead to negative transfer. Hence, we propose a simple heuristical domain selection way via the guidance of DT-SSN. Specifically, we sort all other six domains by descending via their shared slot numbers with the target domain. Then, we simply select top-k domains as sources and use their data to train a single global model. We use the Coach SF and follow the settings in Coach [4]. We compare our methods with the reported results in [3, 11, 4, 7, 17]. We select top-1 and top-3 models respectively. The comparisons are listed in Tab. 1. Coach-TR [4] and CZSL-Adv [7] introduces additional techniques to enhance model performances. However, *our methods does not introduce any complex training techniques and could still obtain SOTA results*. An interesting observation is that utilizing only a single source domain for ATP could lead to an F1 score as high as 0.572, while utilizing 3 source domains decreases to 0.548, and Coach only gets 0.452 using all source domains. This implies that introducing more source domains leads to negative transfer.

Showing the Negative Transfer More Clearly As aforementioned, we still sort the source domains according to the DT-SSN by descending. Then we dynamically add these source domains' data for training the source model. Take the target domain "ATP" as an example, we first train a model only using the PM domain, then we use both the PM and BR domain for training, and last, we train on all six domains. The dynamic transfer results are shown in Fig. 3, where the solid blue bars denote source domains that have at least one common slot type with target domain, while the dashed orange ones have totally different slot types (called non-overlapped domains). We can clearly observe that the F1 scores continually decrease on most domains when non-overlapped domains are continually added. This verifies

Authorized licensed use limited to: Nanjing University. Downloaded on January 15,2024 at 11:35:20 UTC from IEEE Xplore. Restrictions apply.

Table 1. Slot F1 score comparisons. The last three columns show our proposed methods. The columns Coach-1 and Coach-3 denote training with Coach SF via selecting top-1, 3 source domains according to DT-SSN. The final column shows the results with black-box (BB) models.

	CT [3]	RZT [11]	Coach [4]	Coach-TR [4]	CZSL-A [7]	STM [17]	Coach-1	Coach-3	Coarse-BB
ATP	38.82	42.77	45.23	50.90	53.89	50.54	**57.22**	54.81	53.32
BR	27.54	30.68	33.45	34.01	34.06	32.89	**39.40**	38.92	38.30
GW	46.45	50.28	47.93	50.47	52.24	**62.38**	53.55	51.97	57.66
PM	32.86	33.12	28.89	32.01	34.59	34.45	36.95	**39.27**	36.72
RB	14.54	16.43	25.67	22.06	**31.53**	25.39	16.63	18.26	15.43
SCW	39.79	44.45	43.91	46.65	50.61	52.21	35.86	**53.88**	39.27
FSE	13.83	12.25	25.64	25.63	30.05	26.05	29.07	**31.31**	22.44
Avg	30.55	32.85	35.82	37.39	40.99	40.56	38.38	**41.20**	37.59

again the assumption that non-targeted slot types could lead to negative transfer in CDSF. Exceptionally, adding more non-overlapped source domains sometimes leads to better F1 scores, e.g., adding "BR" (Book Restaurant) when the target domain is "RB" (Rate Book). The two domains may be more similar in the feature space. Because this type of similarity is hard to evaluate and we leave it as future work. The bottom right of Fig. 3 plots the average results of these 7 target domains, which shows that selecting 5 or 6 source domains lead to obvious performance degradation.

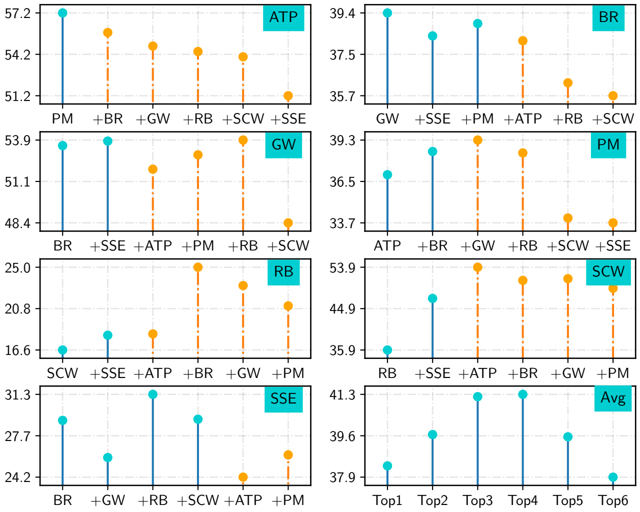


Fig. 3. The dynamic transfer results based on Coach. Each shows a target domain. The last shows the average results.

Source-Free Scenes with only Black-Box Models The hyper-parameter k in domain selection is hard to determine if without any prior knowledge. Furthermore, we may not obtain source models' information and could only obtain black-box models due to data privacy. Because the source models are black-box, we could only obtain the prediction results, e.g., the slot probability distribution of every token. Assume we have K source models, where each model could feedback a probability matrix $Q_k \in \mathcal{R}^{(b\times l)\times n}$, where b, l, n denote the number of target utterances, tokens in each utterance, and target slot types, respectively. Then we calculate the average entropy via $\overline{E}_k = \frac{1}{b\times l}\sum_{i=1}^{b\times l}\sum_{j=1}^{n} -Q_{k,ij}\log Q_{k,ij}$. Next, we calculate Softmax$(-\{\overline{E}_k\}_{k=1}^{K})$ as the weights of each source domain. This is reasonable and related to DT-SSN because if one model tends to output uniform predictions for all target samples, it is less possible for the source domain to own some shared slot types with the target domain. Finally, we utilize the estimated weights to ensemble the prediction probabilities. In our experiments, we train the black-box models via Coarse SF for each domain. For a target domain, we use this entropy-based mechanism to ensemble 6 source black-box models. The results are listed in the last column of Tab. 1. The proposed ensemble strategy could still lead to notable performances (comparable with Coach-TR [4]) even only using black-box source models.

More Sot Filling Datasets We also analyze other CDSF benchmarks such as MultiWOZ [22] (8 domains and 61 slot types) and SGD [23] (20 domains and 240 slot types). The matrix of shared slot numbers is also very sparse, which is similar to Fig. 2 (e), e.g., the domain "Taxi" and "Restaurant" in MultiWOZ do not share any slot types. We expect our proposed methods could also work on these benchmarks. We leave this as future work due to content limitations.

5. CONCLUSION

We explore several domain transferability measures in CDSF. Although slot sparsity leads to non-targeted and zero-shot slot types that exacerbate the difficulty of cross-domain transfer, it inspires a frustratingly easy domain transferability measure, i.e., DT-SSN. An exciting finding is that this simple estimator could reflect the transferring performances well. We then propose to use this estimator to select domains and obtain SOTA results on Snips. With DT-SSN, we further clearly show the negative transfer phenomenon in CDSF. We also investigate a more challenging source-free scene that could only access black-box source models, and the ensemble strategy inspired by DT-SSN gives notable results.

Authorized licensed use limited to: Nanjing University. Downloaded on January 15,2024 at 11:35:20 UTC from IEEE Xplore. Restrictions apply.

6. REFERENCES

[1] Kaisheng Yao, Baolin Peng, Geoffrey Zweig, Dong Yu, Xiaolong Li, and Feng Gao, "Recurrent conditional random field for language understanding," in *ICASSP*, 2014, pp. 4077–4081.

[2] Xiaodong Zhang and Houfeng Wang, "A joint model of intent determination and slot filling for spoken language understanding," in *IJCAI*, 2016, pp. 2993–2999.

[3] Ankur Bapna, Gökhan Tür, Dilek Hakkani-Tür, and Larry P. Heck, "Towards zero-shot frame semantic parsing for domain scaling," in *INTERSPEECH*, 2017, pp. 2476–2480.

[4] Zihan Liu, Genta Indra Winata, Peng Xu, and Pascale Fung, "Coach: A coarse-to-fine approach for cross-domain slot filling," in *ACL*, 2020, pp. 19–25.

[5] Chenwei Zhang, Yaliang Li, Nan Du, Wei Fan, and Philip S. Yu, "Joint slot filling and intent detection via capsule neural networks," in *ACL*, 2019, pp. 5259–5267.

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

[7] Keqing He, Jinchao Zhang, Yuanmeng Yan, Weiran Xu, Cheng Niu, and Jie Zhou, "Contrastive zero-shot learning for cross-domain slot filling with adversarial attack," in *COLING*, 2020, pp. 1461–1467.

[8] Luchen Liu, Xixun Lin, Peng Zhang, and Bin Wang, "Improving cross-domain slot filling with common syntactic structure," in *ICASSP*, 2021, pp. 7638–7642.

[9] A. B. Siddique, Fuad T. Jamour, and Vagelis Hristidis, "Linguistically-enriched and context-aware zero-shot slot filling," in *WWW*, 2021.

[10] 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, "Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces," *CoRR*, vol. abs/1805.10190, 2018.

[11] Darsh J. Shah, Raghav Gupta, Amir A. Fayazi, and Dilek Hakkani-Tür, "Robust zero-shot cross-domain slot filling with example values," in *ACL*, 2019, pp. 5484–5490.

[12] Anh Tuan Tran, Cuong V. Nguyen, and Tal Hassner, "Transferability and hardness of supervised classification tasks," in *ICCV*, 2019, pp. 1395–1405.

[13] Amir Roshan Zamir, Alexander Sax, William B. Shen, Leonidas J. Guibas, Jitendra Malik, and Silvio Savarese, "Taskonomy: Disentangling task transfer learning," in *CVPR*, 2018, pp. 3712–3722.

[14] Alex Tamkin, Trisha Singh, Davide Giovanardi, and Noah D. Goodman, "Investigating transferability in pretrained language models," in *Findings of EMNLP*, 2020, pp. 1393–1401.

[15] Zhi-Hua Zhou, "Learnware: on the future of machine learning," *FSC*, vol. 10, no. 4, pp. 589–590, 2016.

[16] Xin-Chun Li, De-Chuan Zhan, Jia-Qi Yang, Yi Shi, Cheng Hang, and Yi Lu, "Towards understanding transfer learning algorithms using meta transfer features," in *PAKDD*, 2020, pp. 855–866.

[17] Hengtong Lu, Zhuoxin Han, Caixia Yuan, Xiaojie Wang, Shuyu Lei, Huixing Jiang, and Wei Wu, "Slot transferability for cross-domain slot filling," in *Findings of ACL/IJCNLP*, 2021, pp. 4970–4979.

[18] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomás Mikolov, "Enriching word vectors with subword information," *TACL*, vol. 5, pp. 135–146, 2017.

[19] Kazuma Hashimoto, Caiming Xiong, Yoshimasa Tsuruoka, and Richard Socher, "A joint many-task model: Growing a neural network for multiple NLP tasks," in *EMNLP*, 2017, pp. 1923–1933.

[20] Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander J. Smola, "A kernel method for the two-sample-problem," in *NeurIPS*, 2006, pp. 513–520.

[21] Matt J. Kusner, Yu Sun, Nicholas I. Kolkin, and Kilian Q. Weinberger, "From word embeddings to document distances," in *ICML*, 2015, pp. 957–966.

[22] Xiaoxue Zang, Abhinav Rastogi, Srinivas Sunkara, Raghav Gupta, Jianguo Zhang, and Jindong Chen, "Multiwoz 2.2 : A dialogue dataset with additional annotation corrections and state tracking baselines," *CoRR*, vol. abs/2007.12720, 2020.

[23] Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, Raghav Gupta, and Pranav Khaitan, "Towards scalable multi-domain conversational agents: The schema-guided dialogue dataset," in *AAAI*, 2020, pp. 8689–8696.

Authorized licensed use limited to: Nanjing University. Downloaded on January 15,2024 at 11:35:20 UTC from IEEE Xplore. Restrictions apply.