Coach: A Coarse-to-Fine Approach for Cross-domain Slot Filling

Zihan Liu, Genta Indra Winata, Peng Xu, Pascale Fung
Center for Artificial Intelligence Research (CAiRE)
Department of Electronic and Computer Engineering
The Hong Kong University of Science and Technology, Clear Water Bay, Hong Kong
zihan.liu@connect.ust.hk

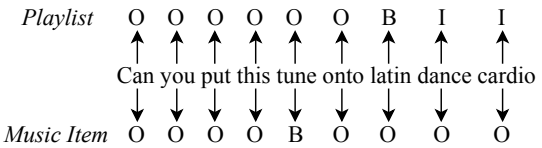
Abstract

As an essential task in task-oriented dialog systems, slot filling requires extensive training data in a certain domain. However, such data are not always available. Hence, cross-domain slot filling has naturally arisen to cope with this data scarcity problem. In this paper, we propose a **Coa**rse-to-fine approa**ch** (**Coach**) for cross-domain slot filling. Our model first learns the general pattern of slot entities by detecting whether the tokens are slot entities or not. It then predicts the specific types for the slot entities. In addition, we propose a *template regularization* approach to improve the adaptation robustness by regularizing the representation of utterances based on utterance templates. Experimental results show that our model significantly outperforms state-of-the-art approaches in slot filling. Furthermore, our model can also be applied to the cross-domain named entity recognition task, and it achieves better adaptation performance than other existing baselines. The code is available at https://github.com/zliucr/coach.

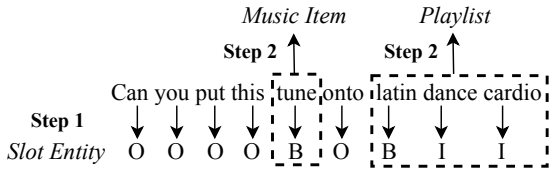
1 Introduction

Slot filling models identify task-related slot types in certain domains for user utterances, and are an indispensable part of task-oriented dialog systems. Supervised approaches have made great achievements in the slot filling task (Goo et al., 2018; Zhang et al., 2019), where substantial labeled training samples are needed. However, collecting large numbers of training samples is not only expensive but also time-consuming. To cope with the data scarcity issue, we are motivated to investigate cross-domain slot filling methods, which leverage knowledge learned in the source domains and adapt the models to the target domain with a minimum number of target domain labeled training samples.

A challenge in cross-domain slot filling is to handle unseen slot types, which prevents general classification models from adapting to the target domain without any target domain supervision signals. Recently, Bapna et al. (2017) proposed a cross-domain slot filling framework, which enables zero-shot adaptation. As illustrated in Figure 1a, their model conducts slot filling individually for each slot type. It first generates word-level representations, which are then concatenated with the representation of each slot type description, and the predictions are based on the concatenated features for each slot type. Due to the inherent variance of slot entities across different domains, it is difficult for this framework to capture the whole slot entity (e.g., "latin dance cardio" in Figure 1a) in the target domain. There also exists a multiple prediction problem. For example, "tune" in Figure 1a could be predicted as "B" for both "music item" and "playlist", which would cause additional trouble for the final prediction.



(a) Framework proposed by Bapna et al. (2017).



(b) Our proposed framework, *Coach*.

Figure 1: Cross-domain slot filling frameworks.

We emphasize that in order to capture the whole slot entity, it is pivotal for the model to share its parameters for all slot types in the source domains and learn the general pattern of slot entities. Therefore, as depicted in Figure 1b, we propose a new cross-domain slot filling framework called *Coach*,

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 19–25

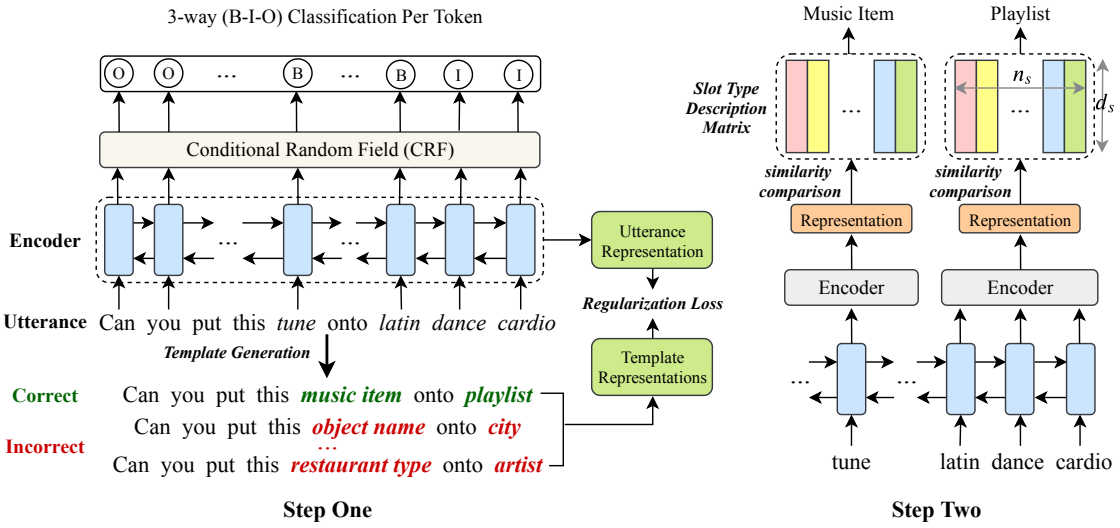


Figure 2: Illustration of our framework, *Coach*, and the template regularization approach.

a coarse-to-fine approach. It first **coarsely** learns the slot entity pattern by predicting whether the tokens are slot entities or not. Then, it combines the features for each slot entity and predicts the specific (**fine**) slot type based on the similarity with the representation of each slot type description. In this way, our framework is able to avoid the multiple predictions problem. Additionally, we introduce a *template regularization* method that delexicalizes slot entity tokens in utterances into different slot labels and produces both correct and incorrect utterance templates to regularize the utterance representations. By doing so, the model learns to cluster the representations of semantically similar utterances (i.e., in the same or similar templates) into a similar vector space, which further improves the adaptation robustness.

Experimental results show that our model surpasses the state-of-the-art methods by a large margin in both zero-shot and few-shot scenarios. In addition, further experiments show that our framework can be applied to cross-domain named entity recognition, and achieves better adaptation performance than other existing frameworks.

2 Related Work

Coarse-to-fine methods in NLP are best known for syntactic parsing (Charniak et al., 2006; Petrov, 2011). Zhang et al. (2017) reduced the search space of semantic parsers by using coarse macro grammars. Different from the previous work, we apply the idea of coarse-to-fine into cross-domain slot filling to handle unseen slot types by separating the slot filling task into two steps (Zhai et al., 2017; Guerini et al., 2018).

Coping with low-resource problems where there are zero or few existing training samples has always been an interesting and challenging task (Kingma et al., 2014; Lample et al., 2018; Liu et al., 2019a,b; Lin et al., 2020). Cross-domain adaptation addresses the data scarcity problem in low-resource target domains (Pan et al., 2010; Jaech et al., 2016; Guo et al., 2018; Jia et al., 2019; Liu et al., 2020; Winata et al., 2020). However, most research studying the cross-domain aspect has not focused on predicting unseen label types in the target domain since both source and target domains have the same label types in the considered tasks (Guo et al., 2018). In another line of work, to bypass unseen label types, Ruder and Plank (2018) and Jia et al. (2019) utilized target domain training samples, so that there was no unseen label type in the target domain. Recently, based on the framework proposed by Bapna et al. (2017) (discussed in Section 1), Lee and Jha (2019) added an attention layer to produce slot-aware representations, and Shah et al. (2019) leveraged slot examples to increase the robustness of cross-domain slot filling adaptation.

3 Methodology

3.1 Coach Framework

As depicted in Figure 2, the slot filling process in our Coach framework consists of two steps. In the first step, we utilize a BiLSTM-CRF structure (Lample et al., 2016) to learn the general pattern of slot entities by having our model predict whether tokens are slot entities or not (i.e.,

3-way classification for each token). In the second step, our model further predicts a specific type for each slot entity based on the similarities with the description representations of all possible slot types. To generate representations of slot entities, we leverage another encoder, BiLSTM (Hochreiter and Schmidhuber, 1997), to encode the hidden states of slot entity tokens and produce representations for each slot entity.

We represent the user utterance with n tokens as $\mathbf{w} = [w_1, w_2, ..., w_n]$, and $\mathbf{E}$ denotes the embedding layer for utterances. The whole process can be formulated as follows:

$$[h_1, h_2, ..., h_n] = \text{BiLSTM}(\mathbf{E}(\mathbf{w})), \quad (1)$$

$$[p_1, p_2, ..., p_n] = \text{CRF}([h_1, h_2, ..., h_n]), \quad (2)$$

where $[p_1, p_2, ..., p_n]$ are the logits for the 3-way classification. Then, for each slot entity, we take its hidden states to calculate its representation:

$$r_k = \text{BiLSTM}([h_i, h_{i+1}, ...h_j]), \quad (3)$$

$$s_k = M_{desc} \cdot r_k, \quad (4)$$

where r_k denotes the representation of the k^{th} slot entity, $[h_i, h_{i+1}, ..., h_j]$ denotes the BiLSTM hidden states for the k^{th} slot entity, $M_{desc} \in R^{n_s \times d_s}$ is the representation matrix of the slot description (n_s is the number of possible slot types and d_s is the dimension of slot descriptions), and s_k is the specific slot type prediction for this k^{th} slot entity. We obtain the slot description representation $r^{desc} \in R^{d_s}$ by summing the embeddings of the N slot description tokens (similar to Shah et al. (2019)):

$$r^{desc} = \sum_{i=1}^{N} \mathbf{E}(t_i), \quad (5)$$

where t_i is the i^{th} token and $\mathbf{E}$ is the same embedding layer as that for utterances.

3.2 Template Regularization

In many cases, similar or the same slot types in the target domain can also be found in the source domains. Nevertheless, it is still challenging for the model to recognize the slot types in the target domain owing to the variance between the source domains and the target domain. To improve the adaptation ability, we introduce a template regularization method.

As shown in Figure 2, we first replace the slot entity tokens in the utterance with different slot labels to generate correct and incorrect utterance templates. Then, we use BiLSTM and an attention layer (Felbo et al., 2017) to generate the utterance and template representations:

$$e_t = h_t w_a, \ \ \alpha_t = \frac{exp(e_t)}{\sum_{j=1}^{n} exp(e_j)}, \ \ R = \sum_{t=1}^{n} \alpha_t h_t, \quad (6)$$

where h_t is the BiLSTM hidden state in the t^{th} step, w_a is the weight vector in the attention layer and R is the representation for the input utterance or template.

We minimize the regularization loss functions for the right and wrong templates, which can be formulated as follows:

$$L^r = \text{MSE}(R^u, R^r), \quad (7)$$

$$L^w = -\beta \times \text{MSE}(R^u, R^w), \quad (8)$$

where R^u is the representation for the user utterance, R^r and R^w are the representations of right and wrong templates, we set β as one, and MSE denotes mean square error. Hence, in the training phase, we minimize the distance between R^u and R^r and maximize the distance between R^u and R^w. To generate a wrong template, we replace the correct slot entity with another random slot entity, and we generate two wrong templates for each utterance. To ensure the representations of the templates are meaningful (i.e., similar templates have similar representations) for training R^u, in the first several epochs, the regularization loss is only to optimize the template representations, and in the following epochs, we optimize both template representations and utterance representations.

By doing so, the model learns to cluster the representations in the same or similar templates into a similar vector space. Hence, the hidden states of tokens that belong to the same slot type tend to be similar, which boosts the robustness of these slot types in the target domain.

4 Experiments

4.1 Dataset

We evaluate our framework on SNIPS (Coucke et al., 2018), a public spoken language understanding dataset which contains 39 slot types across seven domains (intents) and ∼2000 training samples per domain. To test our framework, each time, we choose one domain as the target domain and the other six domains as the source domains.

Training Setting	Zero-shot				Few-shot on 20 (1%) samples				Few-shot on 50 (2.5%) samples			
Domain ↓ Model →	**CT**	**RZT**	**Coach**	**+TR**	**CT**	**RZT**	**Coach**	**+TR**	**CT**	**RZT**	**Coach**	**+TR**
AddToPlaylist	38.82	42.77	45.23	**50.90**	58.36	**63.18**	58.29	62.76	68.69	**74.89**	71.63	74.68
BookRestaurant	27.54	30.68	33.45	**34.01**	45.65	50.54	61.08	**65.97**	54.22	54.49	72.19	**74.82**
GetWeather	46.45	50.28	47.93	**50.47**	54.22	58.86	67.61	**67.89**	63.23	58.87	**81.55**	79.64
PlayMusic	32.86	**33.12**	28.89	32.01	46.35	47.20	53.82	**54.04**	54.32	59.20	62.41	**66.38**
RateBook	14.54	16.43	**25.67**	22.06	64.37	63.33	**74.87**	74.68	76.45	76.87	**86.88**	84.62
SearchCreativeWork	39.79	44.45	43.91	**46.65**	57.83	**63.39**	60.32	57.19	66.38	**67.81**	65.38	64.56
FindScreeningEvent	13.83	12.25	**25.64**	25.63	48.59	49.18	66.18	**67.38**	70.67	74.58	78.10	**83.85**
Average F1	30.55	32.85	35.82	**37.39**	53.62	56.53	63.17	**64.27**	64.85	66.67	74.02	**75.51**

Table 1: Slot F1-scores based on standard BIO structure for SNIPS. Scores in each row represents the performance of the leftmost target domain, and TR denotes template regularization.

Moreover, we also study another adaptation case where there is no unseen label in the target domain. We utilize the CoNLL-2003 English named entity recognition (NER) dataset as the source domain (Tjong Kim Sang and De Meulder, 2003), and the CBS SciTech News NER dataset from Jia et al. (2019) as the target domain. These two datasets have the same four types of entities, namely, PER (person), LOC (location), ORG (organization), and MISC (miscellaneous).

4.2 Baselines

We use word-level (Bojanowski et al., 2017) and character-level (Hashimoto et al., 2017) embeddings for our model as well as all the following baselines.

Concept Tagger (CT) Bapna et al. (2017) proposed a slot filling framework that utilizes slot descriptions to cope with the unseen slot types in the target domain.

Robust Zero-shot Tagger (RZT) Based on CT, Shah et al. (2019) leveraged example values of slots to improve robustness of cross-domain adaptation.

BiLSTM-CRF This baseline is only for the cross-domain NER. Since there is no unseen label in the NER target domain, the BiLSTM-CRF (Lample et al., 2016) uses the same label set for the source and target domains and casts it as an entity classification task for each token, which is applicable in both zero-shot and few-shot scenarios.

4.3 Training Details

We use a 2-layer BiLSTM with a hidden size of 200 and a dropout rate of 0.3 for both the template encoder and utterance encoder. Note that the parameters in these two encoders are not shared. The BiLSTM for encoding the hidden states of slot entity tokens has one layer with a hidden size of 200, which would output the same dimension as the concatenated word-level and char-level embeddings. We use Adam optimizer with a learning rate of 0.0005. Cross-entropy loss is leveraged to train the 3-way classification in the first step, and the specific slot type predictions are used in the second step. We split 500 data samples in the target domain as the validation set for choosing the best model and the remainder are used for the test set. We implement the model in CT and RZT and follow the same setting as for our model for a fair comparison.

5 Results & Discussion

5.1 Cross-domain Slot Filling

Quantitative Analysis As illustrated in Table 1, we can clearly see that our models are able to achieve significantly better performance than the current state-of-the-art approach (RZT). The CT framework suffers from the difficulty of capturing the whole slot entity, while our framework is able to recognize the slot entity tokens by sharing its parameters across all slot types. Based on the CT framework, the performance of RZT is still limited, and Coach outperforms RZT by a ∼3% F1-score in the zero-shot setting. Additionally, template regularization further improves the adaptation robustness by helping the model cluster the utterance representations into a similar vector space based on their corresponding template representations.

Interestingly, our models achieve impressive performance in the few-shot scenario. In terms of the averaged performance, our best model (Coach+TR) outperforms RZT by ∼8% and ∼9% F1-scores on the 20-shot and 50-shot settings, respectively. We conjecture that our model is able to better recognize the whole slot entity in the target domain and map the representation of the slot entity belonging to the same slot type into a similar vector space

Target Samples‡	0 samples		20 samples		50 samples	
	unseen	seen	unseen	seen	unseen	seen
CT	27.10	44.18	50.13	61.21	62.05	69.64
RZT	28.28	47.15	52.56	63.26	63.96	73.10
Coach	32.89	50.78	61.96	73.78	74.65	76.95
Coach+TR	**34.09**	**51.93**	**64.16**	**73.85**	**76.49**	**80.16**

Table 2: Averaged F1-scores for seen and unseen slots over all target domains. ‡ represent the number of training samples utilized for the target domain.

to the representation of this slot type based on Eq (4). This enables the model to quickly adapt to the target domain slots.

Analysis on Seen and Unseen Slots We take a further step to test the models on seen and unseen slots in target domains to analyze the effectiveness of our approaches. To test the performance, we split the test set into "unseen" and "seen" parts. An utterance is categorized into the "unseen" part as long as there is an unseen slot (i.e., the slot does not exist in the remaining six source domains) in it. Otherwise we categorize it into the "seen" part. The results for the "seen" and "unseen" categories are shown in Table 2. We observe that our approaches generally improve on both unseen and seen slot types compared to the baseline models. For the improvements in the unseen slots, our models are better able to capture the unseen slots since they explicitly learn the general pattern of slot entities. Interestingly, our models also bring large improvements in the seen slot types. We conjecture that it is also challenging to adapt models to seen slots due to the large variance between the source and target domains. For example, slot entities belonging to the "object type" in the "RateBook" domain are different from those in the "SearchCreativeWork" domain. Hence, the baseline models might fail to recognize these seen slots in the target domain, while our approaches can adapt to the seen slot types more quickly in comparison. In addition, we observe that template regularization improves performance in both seen and unseen slots, which illustrates that clustering representations based on templates can boost the adaptation ability.

5.2 Cross-domain NER

From Table 3, we see that the Coach framework is also suitable for the case where there are no unseen labels in the target domain in both the zero-shot and few-shot scenarios, while CT and RZT are not as effective as BiLSTM-CRF. However, we observe that template regularization loses its effectiveness in this task, since the text in NER is relatively more open, which makes it hard to capture the templates for each label type.

Target Samples	0 samples	50 samples
CT (Bapna et al. (2017))	61.43	65.85
RZT (Shah et al. (2019))	61.94	65.21
BiLSTM-CRF	61.77	66.57
Coach	64.08	**68.35**
Coach + TR	**64.54**	67.45

Table 3: F1-scores on the NER target domain (CBS SciTech News).

Task	zero-shot			few-shot on 50 samples		
	sum	trs	bilstm	sum	trs	bilstm
Slot Filling	33.89	34.33	**35.82**	73.80	72.66	**74.02**
NER	63.04	63.29	**64.47**	66.98	68.04	**68.35**

Table 4: Ablation study in terms of the methods to encode the entity tokens on Coach.

5.3 Ablation Study

We conduct an ablation study in terms of the methods to encode the entity tokens (described in Eq. (3)) to investigate how they affect the performance. Instead of using BiLSTM, we try two alternatives. One is to use the encoder of Transformer (trs) (Vaswani et al., 2017), and the other is to simply sum the hidden states of slot entity tokens. From Table 4, we can see that there is no significant performance difference among different methods, and we observe that using BiLSTM to encode the entity tokens generally achieves better results.

6 Conclusion

We introduce a new cross-domain slot filling framework to handle the unseen slot type issue. Our model shares its parameters across all slot types and learns to predict whether input tokens are slot entities or not. Then, it detects concrete slot types for these slot entity tokens based on the slot type descriptions. Moreover, template regularization is proposed to improve the adaptation robustness further. Experiments show that our model significantly outperforms existing cross-domain slot filling approaches, and it also achieves better performance for the cross-domain NER task, where there is no unseen label type in the target domain.

Acknowledgments

This work is partially funded by ITF/319/16FP and MRP/055/18 of the Innovation Technology Commission, the Hong Kong SAR Government.

References

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

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Eugene Charniak, Mark Johnson, Micha Elsner, Joseph Austerweil, David Ellis, Isaac Haxton, Catherine Hill, R Shrivaths, Jeremy Moore, Michael Pozar, et al. 2006. Multilevel coarse-to-fine pcfg parsing. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 168–175.

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*.

Bjarke Felbo, Alan Mislove, Anders Søgaard, Iyad Rahwan, and Sune Lehmann. 2017. Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1615–1625.

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

Marco Guerini, Simone Magnolini, Vevake Balaraman, and Bernardo Magnini. 2018. Toward zero-shot entity recognition in task-oriented conversational agents. In *Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue*, pages 317–326.

Jiang Guo, Darsh Shah, and Regina Barzilay. 2018. Multi-source domain adaptation with mixture of experts. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4694–4703.

Kazuma Hashimoto, Yoshimasa Tsuruoka, Richard Socher, et al. 2017. A joint many-task model: Growing a neural network for multiple nlp tasks. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1923–1933.

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

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

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.

Durk P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. 2014. Semi-supervised learning with deep generative models. In *Advances in neural information processing systems*, pages 3581–3589.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270.

Guillaume Lample, Myle Ott, Alexis Conneau, Ludovic Denoyer, and MarcAurelio Ranzato. 2018. Phrase-based & neural unsupervised machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5039–5049.

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.

Zhaojiang Lin, Zihan Liu, Genta Indra Winata, Samuel Cahyawijaya, Andrea Madotto, Yejin Bang, Etsuko Ishii, and Pascale Fung. 2020. Xpersona: Evaluating multilingual personalized chatbot. *arXiv preprint arXiv:2003.07568*.

Zihan Liu, Jamin Shin, Yan Xu, Genta Indra Winata, Peng Xu, Andrea Madotto, and Pascale Fung. 2019a. Zero-shot cross-lingual dialogue systems with transferable latent variables. 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 1297–1303.

Zihan Liu, Genta Indra Winata, and Pascale Fung. 2020. Zero-resource cross-domain named entity recognition. *arXiv preprint arXiv:2002.05923*.

Zihan Liu, Genta Indra Winata, Zhaojiang Lin, Peng Xu, and Pascale Fung. 2019b. Attention-informed mixed-language training for zero-shot cross-lingual task-oriented dialogue systems. *arXiv preprint arXiv:1911.09273*.

Sinno Jialin Pan, Xiaochuan Ni, Jian-Tao Sun, Qiang Yang, and Zheng Chen. 2010. Cross-domain sentiment classification via spectral feature alignment. In *Proceedings of the 19th international conference on World wide web*, pages 751–760. ACM.

Slav Petrov. 2011. *Coarse-to-fine natural language processing*. Springer Science & Business Media.

Sebastian Ruder and Barbara Plank. 2018. Strong baselines for neural semi-supervised learning under domain shift. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1044–1054.

Darsh Shah, Raghav Gupta, Amir Fayazi, and Dilek Hakkani-Tur. 2019. Robust zero-shot cross-domain slot filling with example values. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5484–5490, Florence, Italy. Association for Computational Linguistics.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

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

Genta Indra Winata, Samuel Cahyawijaya, Zihan Liu, Zhaojiang Lin, Andrea Madotto, Peng Xu, and Pascale Fung. 2020. Learning fast adaptation on cross-accented speech recognition. *arXiv preprint arXiv:2003.01901*.

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

Chenwei Zhang, Yaliang Li, Nan Du, Wei Fan, and Philip Yu. 2019. Joint slot filling and intent detection via capsule neural networks. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5259–5267, Florence, Italy. Association for Computational Linguistics.

Yuchen Zhang, Panupong Pasupat, and Percy Liang. 2017. Macro grammars and holistic triggering for efficient semantic parsing. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1214–1223.