Few-shot Learning for Slot Tagging with Attentive Relational Network

Cennet Oguz[1] **Ngoc Thang Vu**[2]
[1]Multilinguality and Language Technology, German Research Center for Artificial Intelligence
[2]Institute for Natural Language Processing, University of Stuttgart
cennet.oguz@dfki.de thangvu@ims.uni-stuttgart.de

Abstract

Metric-based learning is a well-known family of methods for few-shot learning, especially in computer vision. Recently, they have been used in many natural language processing applications but not for slot tagging. In this paper, we explore metric-based learning methods in the slot tagging task and propose a novel metric-based learning architecture - Attentive Relational Network. Our proposed method extends relation networks, making them more suitable for natural language processing applications in general, by leveraging pretrained contextual embeddings such as ELMO and BERT and by using attention mechanism. The results on SNIPS data show that our proposed method outperforms other state of the art metric-based learning methods.

1 Introduction

Neural networks have been successfully utilized in natural language processing (NLP) applications with a large amount of hand-labeled data whereas they suffer a persistent challenge of low-resource. The approach of learning with few samples, known as few-shot learning - a branch of meta-learning (*learn to learn*) - has recently been popularized (Fei-Fei et al., 2006; Ravi and Larochelle, 2016; Vinyals et al., 2016; Snell et al., 2017; Sung et al., 2018) in computer vision. Recently, few-shot learning has also been applied to NLP tasks, e.g. natural language understanding (Dou et al., 2019), text classification (Jiang et al., 2018; Rios and Kavuluru, 2018; Gao et al., 2019; Geng et al., 2019), machine translation (Gu et al., 2018) and relation classification (Obamuyide and Vlachos, 2019).

In the slot tagging task, we aim at predicting task-specific values (e.g. artist, time) for slots (placeholders) in user utterances. Oguz and Vu (2020) propose a two-stage modeling approach to exploit domain-agnostic features to tackle low-resource domain challenges. Besides, the other state of the art techniques e.g. based on external memory (Peng and Yao, 2015), ranking loss (Vu et al., 2016), encoder (Kurata et al., 2016), and attention (Zhu and Yu, 2017) have achieved promising results with a wide range of neural networks methods.

However as many other NLP applications, the low-resource issue is a tremendous challenge for slot tagging in new domains, although labeled samples exist in related domains. Many studies have recently proposed to overcome this low-resource challenge using different techniques, e.g. multi-task modeling (Jaech et al., 2016), adversarial training (Kim et al., 2017), and pointer networks (Zhai et al., 2017). In addition, studies like zero-shot learning has influenced the studies of the domain scaling problem for slots prediction (Bapna et al., 2017), eliminating the need of labeled examples for transferring reusable concepts (Zhu and Yu, 2018; Lee and Jha, 2019), and conveying the domain-agnostic concepts between the intents (Shah et al., 2019) by exploiting label names and descriptions. Likewise, (Hou et al., 2020) use label semantics within a few-shot classification method TapNet (Yoon et al., 2019).

We suggest using a small amount of annotated samples from different domains as training input instead of slot descriptions and slot names as in previos zero-shot (Bapna et al., 2017; Lee and Jha, 2019; Shah et al., 2019) and few-shot (Hou et al., 2020) slot tagging studies 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

Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics, pages 1566–1572
April 19 - 23, 2021.

relationships between names and tokens.

In this paper, we provide a new experimental design where the slot tagging task needs to be solved for unseen slot labels. The experimental design mimics previous few-shot learning studies (Vinyals et al., 2016; Snell et al., 2017; Sung et al., 2018). Thus, the existing data sources from different domains are used to learn meta-knowledge, whereas unseen labels from low-resource domains are used to evaluate the models. Furthermore, we propose a novel modeling approach - *Attentive Relational Network*, inspired by (Sung et al., 2018; Jiang et al., 2018; Jetley et al., 2018), that leverages contextual embeddings such as ELMO and BERT and extends the previous relation networks (Sung et al., 2018) by learning to attend local and global features (Jetley et al., 2018). Experimental results on SNIPS data show that the proposed model outperforms other few-shot learning networks.

2 Methods

2.1 Input

FastText (Mikolov et al., 2018) is an approach to enrich the word vectors with a bag of character n-gram vectors.
ELMo (Peters et al., 2018) is a contextualized word representation methods. It concatenates the output of two LSTM independently trained on the bidirectional language modeling task and return the hidden states for the given input sequence.
BERT (Devlin et al., 2019) uses a bidirectional transformer model that is trained on a masked language modeling task. Because of WordPiece embeddings (Wu et al., 2016), there are different choices of presenting words. We use the first subtoken for representing the word as proposed in (Devlin et al., 2019). Additionally, due to the structure of multiple successive layers, i.e., 24 layers and as suggested in (Oguz and Vu, 2020), we select *10th*, *11th*, *12th*, and *13th* as the focused layers on local context (Clark et al., 2019; Tenney et al., 2018) for slot tagging.

2.2 Meta-learning strategy

Despite the fact that the proposed methods differ in their learning strategies, episode-based training is the same in meta-training and meta-testing phases for proposed meta models as mentioned in (Chen et al., 2018). For the purpose of applying episodic training in a robust way, we follow the proposed procedures in Vinyals et al. (2016); Snell et al. (2017). In the episodic training, each step - episode - is formed to compute gradients and update the model parameters. An episode consists of two components: support and query sets. To construct an episode, C unique classes are randomly sampled, and for each selected C unique classes K labeled examples randomly drawn for support $S = \{(x_i, y_i)\}_{i=1}^{m}$ and query $Q = \{(x_i, y_i)\}_{i=1}^{m}$ set, where $K > 1$ and $m = K * C$. The same episode composition strategy is applied in the meta-testing stage to evaluate the performance of the trained model over unseen classes.

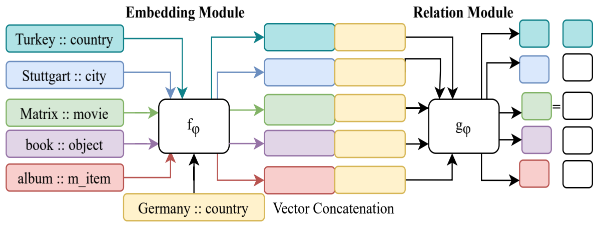

Figure 1: One-shot training example of slot tagging with Relation Network: Embedding Module extracts the feature vectors of each slot value whereas Relation Module calculates the relation scores between support samples and query. Then, the label of most relational value is assigned as a label of query.

Meta-training. The aim of this phase is to learn a meta learner that maps from a few labeled samples to a classifier. In each episode, meta-training employs a two-stage process: (1) the first stage implies producing the feature maps from the given input S and Q, called embedding function $f_\phi(x)$ (2) the second stage is to make prediction conditioned on few labeled examples, S. More formally, we define an episode E_{train} includes S and Q selected from train data,D_{train} . Then, the model is trained to minimize the label prediction error in the Q conditioned on S, i.e., $P_\theta(y_j|x_j, S)$, by utilizing the distance or relation metrics like $distance/relation(f_\phi(x_j), f_\phi(x_i))$, as also shown with an example in Figure 1.
Meta-testing. In this phase, we test the performance of trained meta-learner on unseen labels by following the same steps in meta-training phase. An episode E_{test} with S and Q is formed by randomly selecting from test data, D_{test}. The over all accuracy is computed by averaging the test episodes, $acc = \frac{1}{||E_{test}||} \sum_i E_{test}$.

We define $C = 5$ in meta-training and meta-testing stages except the meta-testing stage of *SearchCreativeW.* domain with $C = 3$ because *SearchCreativeW.* domain has only three slots. We

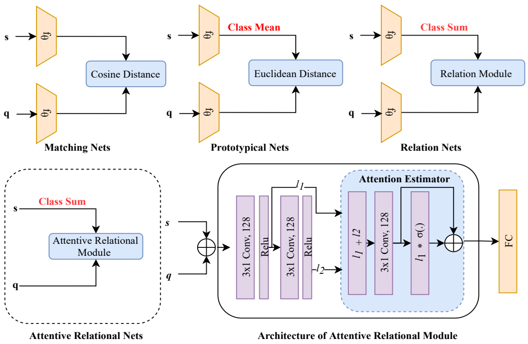

Figure 2: Few-shot learning models: Matching Nets compare each support sample with query in order to calculate the distance metric, Prototypical Nets rely on class mean of support set, and Relation Nets are based on class sum of support samples. f_θ represent the embedding module of each network and Attentive Relational Nets eject embedding module. Support sets are represented with s whereas q denotes the query.

train all the model within 10,000 episodes, and evaluate with 1000 test episode after every 500 steps of total train episodes.

2.3 Models

We focus on three *metric-based learning* few-shot learning methods such as Matching Networks (Vinyals et al., 2016), Prototypical Network (Snell et al., 2017), and Relation Network (Sung et al., 2018). Each network consists of two consecutive modules. The first module, called embedding function, focuses on the learning of the transferable embeddings for support and query samples. The second module is the classifier which identify the corresponding classes over the defined metric scores, e.g., distance and relation.

Matching Networks (MatchingNets) compare the cosine distance between the query feature and each support feature, and computes average cosine distance for each class.

Prototypical Networks (PrototypicalNets) compare the Euclidean distance between query features and the class mean of support features.

Relation Networks (RelationNets) propose a learnable non-linear relation module to output the relation scores over element-wise sum of each support and query features.

In one-shot scenario, MatchingNets and PrototypicalNets could be interpreted as identical, RelationNets differs with the relation module in order to calculate the relation score.

2.3.1 Attentive Relational Networks

We propose a novel metric-learning approach - Attentive Relational Networks (AttentiveRelationalNets) that highlight the relevant, and suppress the misleading between support and query samples. AttentiveRelationalNets address the few-shot classification problem by utilizing *learn to compare based on attention* insight. This can be seen as extending the strategy of Sung et al. (2018) to include a learnable attention module. A trainable attention module, inspired from Jetley et al. (2018), is added to incorporate the relation module of RelationNets. Besides, we make use of pretrained (contextual) embeddings since they have the proven strength on feature extraction for linguistics items instead of using embedding module, Figure 2.

For AttentiveRelationalNets, as shown in Figure 2, we implement two convolution blocks as it is in RelationNets with residual connection, as proposed in He et al. (2016).Then, the convolution blocks produce local descriptors, i.e., l_1 and l_2, as the output of activation function and pass them to the attention estimator in order to find the global g feature vector.

In order to compute the compatibility function, we define a convolution function with the input of two local features to an addition operation, $c = \langle u, l_1 + l_2 \rangle$. Here, u represents the universal set of features relevant to the s and q pairs in the object categories. We normalize the compatibility scores by using sigmoid operation, $a = \sigma(c)$. Then, the global feature vector is assessed by element-wise weighted average, i.e, $g = l_1 * a$. Afterwards, we concatenate the global features g with learned compatibility scores c as the input of the linear classifier which eventually produces a scalar in range of 0 to 1 representing the similarity between s and q , which is called relation score r. We define *mean square error*, as proposed in RelationNets, as the objective function of our model.

2.4 Evaluation

As we use the same implementation details for meta-training and meta-testing stages, we also evaluate the performance by few-shot classification accuracy following previous studies few-shot learning (Vinyals et al., 2016; Snell et al., 2017; Sung et al., 2018) with a small change: since the meta-learning approaches are fast learning methods, we present the average accuracies of training epochs instead of presenting the best accuracy.

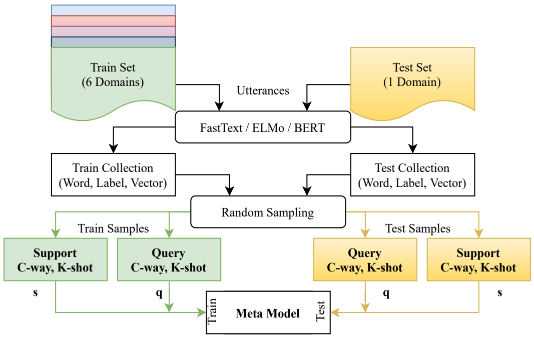

Figure 3: The schema of few-shot data construction for train and test episodes. 6 different domains are used for training phase whereas one different domain is used for test. Green path indicates the training support and query samples while yellow path represents the samples of test episodes. Train and test collections include N number of values of each slot. Random sampling function draws K samples for each of C slots.

3 Settings

3.1 Resources

In our study we address the few-shot learning approaches to recognize novel slot categories with very few examples from a new domain. In order to provide a deep experimental analysis of proposed networks and language models and to compare our model among each other, we set various experimental scenarios with different data and different K-shot sizes. Hence, 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. Thus, it is a well-categorized dataset which include tasks in domains, which makes the setup more realistic; learn to learn on a bunch of domains and test on new domains. We split SNIPS with the purpose of creating a single-domain dataset. We combine the originally divided training, testing, and development sets and separate them into domain sets in order to create new train and test data.

3.2 Few-shot Data Construction

Meta-learning models aim to learn from the training tasks, i.e. the train label space is disjoint with test label space and the trained model evaluated on unseen classes. Therefore, we utilize other domain data as the training set whereas the models are evaluated by using the current domain. Thus, we created 7 different sets contain a train which consists of 6 different domains as well as a test set which includes only one test domain.

As can be seen from Figure 3, we aggregate the six different domain data for the training set, whereas the remain one domain is used for testing, aiming at evaluating the performance of models on unseen classes per domain. Then, we convert the train and test sets to *train* and *test* collection that contain triplets in order to mimic the same data organization in the previous meta-learning studies. The triplet consists of three items: token, label, vector. The vector of the corresponding token is produced by using different (contextual) embeddings from randomly selected sentences for each label from the train and test set separately. Thus, the train collection is formed with the triplets from the different domain slots, whereas the test collection includes only the triplets of labels from the corresponding domain.

To investigate the efficiency of different models according to data availability in the few-shot setting, we experiment with three data collections sizes of slot values: 50, 100, and 200. Note that the collection size controls the total number of values that can be seen for each slot during training. In the meta-testing stage, we only use the test collection with size of 200 to be able to keep the comparative analyses under control. Furthermore, we examine different numbers of K shot, namely 5-shot, 10-shot, and 15-shot.

4 Results and Analysis

4.1 Different Network Architectures

Table 1 shows the performance in comparison with state of the art metric-based learning models on slot tagging task with different (contextual) embeddings. As can be seen from the scores, RelationNets with ELMo embeddings constantly give the best results, whereas MatchingNets present the lowest scores for each embedding variance. We assume that learning from the distance or relation scores with individual support and query samples instead of the class sum, i.e., as it is in PrototypicalNets, or class mean, i.e., as it is in RelationNets, decrease the learning performance.

Furthermore, Table 2 presents the results from AttentiveRelationalNets with different embeddings methods and demonstrates that our proposed model with ELMo and BERT outperforms the previous models consistently. Additionally, AttentiveRelationalNets significantly improve the results with BERT from the previous experiments in Table 1. When the success of AttentiveRelationalNets is

Table 1: Few-shot slot tagging on SNIPS data. Results are accuracies averaged of three different slot value sizes (50, 100, and 200) with different Ks (5, 10, and 15).

	FastText									ELMo									BERT								
	MatchingNet			PrototypicalNet			RelationNet			MatchingNet			PrototypicalNet			RelationNet			MatchingNet			PrototypicalNet			RelationNet		
Domain / K-shot	5	10	15	5	10	15	5	10	15	5	10	15	5	10	15	5	10	15	5	10	15	5	10	15	5	10	15
AddToPlaylist	20.4	20.5	20.5	67.4	67.6	67.6	65.9	70.4	71.7	65.4	71.7	73.9	72.1	71.8	71.1	70.0	75.1	**76.3**	65.9	70.6	73.2	70.8	71.5	71.4	69.9	74.0	74.3
PlayMusic	20.9	20.8	20.7	68.9	68.6	69.1	68.8	72.2	73.7	67.3	73.7	75.9	73.5	72.2	71.5	72.2	76.5	**78.0**	64.8	70.9	73.3	67.4	68.2	67.7	67.7	71.6	73.0
BookRestaurant	23.5	23.8	24.0	75.0	74.5	74.3	78.7	82.5	83.6	71.3	76.8	78.9	81.2	79.5	77.5	82.9	85.9	**86.5**	68.4	72.7	74.8	75.8	76.5	76.8	78.4	81.8	83.7
GetWeather	20.7	25.7	20.7	76.6	75.9	75.7	79.5	84.2	85.3	75.6	82.2	84.8	80.1	78.5	77.9	79.8	85.1	**87.0**	75.3	81.7	84.3	78.5	78.9	79.1	83.6	86.2	85.8
RateBook	34.5	35.1	35.2	87.4	88.3	87.2	89.8	92.2	93.9	83.8	89.3	90.6	90.2	90.2	89.6	90.0	93.9	**95.1**	88.0	92.0	93.7	92.8	93.6	93.2	92.0	94.1	93.9
SearchCreativeW.	37.8	37.9	38.2	78.1	78.2	78.2	82.8	85.9	86.3	82.8	86.4	88.6	85.8	85.7	83.8	89.7	92.1	**92.9**	79.7	84.4	87.1	81.4	82.0	81.8	86.0	90.3	91.3
FindScreeningE.	21.2	21.3	21.4	73.4	73.5	73.7	78.8	83.1	84.6	80.7	87.7	90.9	83.5	81.6	81.0	86.8	89.9	**91.7**	78.6	86.4	89.5	78.0	78.1	78.3	81.3	86.8	87.9

Table 2: Few-shot slot tagging on SNIPS data with Attentive Relational Networks. Results are accuracies averaged of three different slot value sizes (50, 100, and 200) with different Ks (5, 10, and 15).

	FastText			ELMo			BERT		
Domain	5	10	15	5	10	15	5	10	15
AddToPlaylist	63.6	66.1	68.3	71.9	75.8	77.7	72.6	76.6	**78.5**
PlayMusic	68.5	71.8	73.2	74.7	77.7	79.2	72.8	77.0	**78.7**
BookRestaurant	78.9	82.0	83.1	84.2	87.2	**87.6**	82.7	86.0	87.5
GetWeather	79.0	82.7	84.5	83.3	87.1	88.5	84.9	89.5	**90.2**
RateBook	88.1	90.9	91.4	92.3	94.8	95.6	94.8	96.5	**96.7**
SearchCreativeW.	81.7	84.4	85.1	90.8	93.0	**93.2**	88.2	91.5	92.6
FindScreeningE.	76.9	81.2	82.1	87.3	90.8	**92.3**	82.9	87.0	88.3

examined extensively, it seems that our proposed model gives better results on slot labels that categorize common nouns, while provides relative competitive results for proper nouns. Note that proper nouns, in general, seem to be challenging for all setups. In addition, RelationalNets outperform AttentiveRelationalNets with FastText along with the explanation that embedding function is still effective with average embeddings. However, as opposed to RelationNets, Attentive Relational Networks do better classification overall because of trainable attention. Since trainable attention highlights the relevant features between the slot values labeled with the same slot, whereas it suppresses the misleading them. In an other word, slot local features are able to be more informative for the model while the global features are suppressed.

4.2 Different Contextual Embeddings

Table 2 shows that ELMo and BERT have comparable performance, with BERT slightly better on most tasks: ELMo, however, scores higher on *FindScreeningE.* consistently with AttentiveRelationalNets and all different Ks. Although embedding modules are presented as a feature extraction method for inputs according to distance or relational score, the significant performance gap between FastText and contextualized embeddings shows that the contextualized features outperform the embedding module of few-shot classification models. On the other hand, when we compare FastText embeddings with contextualized word vectors in Table 2, the lower results can be seen. Additionally, when FastText features are compared between the results of RelationNets and AttentiveRelationalNets, we observe that RelationNets outperform AttentiveRelationalNets.

We further look at the wrong predictions in order to understand the reason for the success of ELMo on *FindScreeningE.* and observe that ELMo shows high performance on the labels of proper nouns such as city and location and the labels like *object_type* and *movie_type*. However, BERT demonstrates high performance on proper nouns such as artist, album, hence it outperforms ELMo on *AddToPlaylist* and *PlayMusic* domains. In addition, the improvement of BERT with AttentiveRelationalNets mostly relies on the increase of the accuracy of overall labels, but especially the slot labels that contain common nouns.

4.3 Different Collection and Shot Sizes

AttentiveRelationalNets demonstrate a linear correlation between the increase of performance and the increase of collection size. In addition, the increase of shot size mostly shows improvement in overall results, apart from PrototypicalNets which show their highest result with 10-shot.

5 Conclusion

We presented a deep analysis with a wide variety of few-shot learning methods and pretrained (contextual) embeddings for slot tagging. Furthermore, we proposed a novel architecture that leverages attention mechanism attending both, local and global features of given support samples. Experimental results on SNIPS dataset show that a) pretrained contextual embeddings contributed to high performance and b) our proposed approach consistently outperformed other methods in all setups.

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.

Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang. 2018. A closer look at few-shot classification. In *International Conference on Learning Representations*.

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, Maël Primet, and Joseph Dureau. 2018. Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces. *ArXiv*, abs/1805.10190.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *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.

Zi-Yi Dou, Keyi Yu, and Antonios Anastasopoulos. 2019. Investigating meta-learning algorithms for low-resource natural language understanding tasks. 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 1192–1197.

Li Fei-Fei, Rob Fergus, and Pietro Perona. 2006. One-shot learning of object categories. *IEEE transactions on pattern analysis and machine intelligence*, 28(4):594–611.

Tianyu Gao, Xu Han, Zhiyuan Liu, and Maosong Sun. 2019. Hybrid attention-based prototypical networks for noisy few-shot relation classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6407–6414.

Ruiying Geng, Binhua Li, Yongbin Li, Xiaodan Zhu, Ping Jian, and Jian Sun. 2019. Induction networks for few-shot text classification. 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 3904–3913, Hong Kong, China. Association for Computational Linguistics.

Jiatao Gu, Yong Wang, Yun Chen, Victor OK Li, and Kyunghyun Cho. 2018. Meta-learning for low-resource neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3622–3631.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Yutai Hou, Wanxiang Che, Yongkui Lai, Zhihan Zhou, Yijia Liu, Han Liu, and Ting Liu. 2020. Few-shot slot tagging with collapsed dependency transfer and label-enhanced task-adaptive projection network. *arXiv preprint arXiv:2006.05702*.

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

Saumya Jetley, Nicholas A Lord, Namhoon Lee, and Philip HS Torr. 2018. Learn to pay attention. In *International Conference on Learning Representations*.

Xiang Jiang, Mohammad Havaei, Gabriel Chartrand, Hassan Chouaib, Thomas Vincent, Andrew Jesson, Nicolas Chapados, and Stan Matwin. 2018. Attentive task-agnostic meta-learning for few-shot text classification.

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.

Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. 2018. Advances in pre-training distributed word representations. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.

Abiola Obamuyide and Andreas Vlachos. 2019. Model-agnostic meta-learning for relation classification with limited supervision. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5873–5879.

Cennet Oguz and Ngoc Thang Vu. 2020. A two-stage model for slot filling in low-resource settings: Domain-agnostic non-slot reduction and pretrained contextual embeddings. In *Proceedings of SustaiNLP: Workshop on Simple and Efficient Natural Language Processing*, pages 73–82.

Baolin Peng and Kaisheng Yao. 2015. Recurrent neural networks with external memory for language understanding. *ArXiv*, abs/1506.00195.

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.

Sachin Ravi and Hugo Larochelle. 2016. Optimization as a model for few-shot learning.

Anthony Rios and Ramakanth Kavuluru. 2018. Few-shot and zero-shot multi-label learning for structured label spaces. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing. Conference on Empirical Methods in Natural Language Processing*, volume 2018, page 3132. NIH Public Access.

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.

Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, pages 4077–4087.

Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. 2018. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1199–1208.

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. 2018. What do you learn from context? probing for sentence structure in contextualized word representations. In *International Conference on Learning Representations*.

Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. 2016. Matching networks for one shot learning. In *Advances in neural information processing systems*, pages 3630–3638.

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, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Gregory S. Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *ArXiv*, abs/1609.08144.

Sung Whan Yoon, Jun Seo, and J. Moon. 2019. Tapnet: Neural network augmented with task-adaptive projection for few-shot learning. In *ICML*.

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

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.