

Robust Zero-Shot Cross-Domain Slot Filling with Example Values

Darsh J Shah^{*1}, Raghav Gupta^{*2}, Amir A Fayazi², and Dilek Hakkani-Tür³

¹MIT CSAIL, Cambridge, MA

²Google Research, Mountain View, CA

³Amazon Alexa AI, Sunnyvale, CA

darsh@csail.mit.edu, {raghavgupta, amiraf}@google.com,
dilek@ieee.org

Abstract

Task-oriented dialog systems increasingly rely on deep learning-based slot filling models, usually needing extensive labeled training data for target domains. Often, however, little to no target domain training data may be available, or the training and target domain schemas may be misaligned, as is common for web forms on similar websites. Prior zero-shot slot filling models use slot descriptions to learn concepts, but are not robust to misaligned schemas. We propose utilizing both the slot description and a small number of examples of slot values, which may be easily available, to learn semantic representations of slots which are transferable across domains and robust to misaligned schemas. Our approach outperforms state-of-the-art models on two multi-domain datasets, especially in the low-data setting.

1 Introduction

Goal-oriented dialog systems assist users with tasks such as finding flights, booking restaurants and, more recently, navigating user interfaces, through natural language interactions. Slot filling models, which identify task-specific parameters/slots (e.g. *flight date*, *cuisine*) from user utterances, are key to the underlying spoken language understanding (SLU) systems. Advances in SLU have enabled virtual assistants such as Siri, Alexa and Google Assistant. There is also significant interest in adding third-party functionality to these assistants. However, supervised slot fillers (Young, 2002; Bellegarda, 2014) require abundant labeled training data, more so with deep learning enhancing accuracy at the cost of being data intensive (Mesnil et al., 2015; Kurata et al., 2016).

Asterisk (*) denotes equal contribution. Research conducted when all authors were at Google Research.

From?	To?	Depart	Return
DEPART	ARRIVE	DEPART DATE	RETURN DATE

Figure 1: Misaligned schemas for flight booking from [kayak.com](#) (top) and [southwest.com](#) (bottom): slot name *depart* in the two schemas refers to departure date and departure city respectively, hence models trained on one schema may falter on the other.

Two key challenges with scaling slot fillers to new domains are adaptation and misaligned schemas (here, slot name mismatches). Extent of supervision may vary across domains: there may be ample data for *Flights* but none for *Hotels*, requiring models to leverage the former to learn semantics of reusable slots (e.g. *time*, *destination*). In addition, schemas for overlapping domains may be incompatible by way of using different names for the same slot or the same name for different slots. This is common with web form filling: two sites in the same domain may have misaligned schemas, as in Figure 1, precluding approaches that rely on schema alignment.

Zero-shot slot filling, typically, either relies on slot names to bootstrap to new slots, which may be insufficient for cases like in Figure 1, or uses hard-to-build domain ontologies/gazetteers. We counter that by supplying a small number of example values in addition to the slot description to condition the slot filler. This avoids negative transfer from misaligned schemas and further helps identify unseen slots while retaining cross-domain transfer ability. Besides, example values for slots can either be crawled easily from existing web forms or specified along with the slots, with little overhead.

Given as few as 2 example values per slot, our model surpasses prior work in the zero/few-shot setting on the SNIPS dataset by an absolute 2.9% slot F1, and is robust to misaligned schemas, as experiments on another multi-domain dataset show.

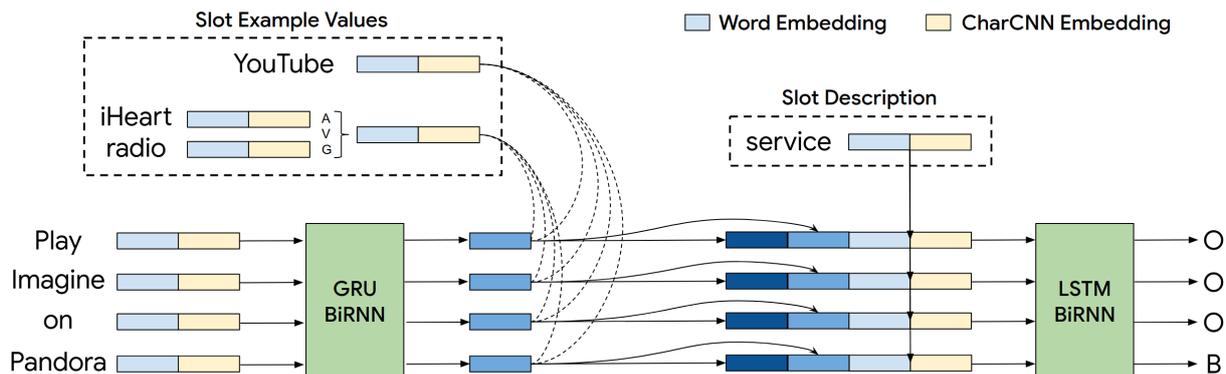


Figure 2: Illustration of the overall model with all inputs and outputs shown.

2 Related Work

Settings with resource-poor domains are typically addressed by adapting from resource-rich domains (Blitzer et al., 2006; Pan et al., 2010; Chen et al., 2018; Guo et al., 2018; Shah et al., 2018). To this end approaches such as domain adversarial learning (Liu and Lane, 2017) and multi-task learning (Jaech et al., 2016; Goyal et al., 2018; Siddhant et al., 2018) have been adapted to SLU and related tasks (Henderson et al., 2014). Work targeting domain adaptation specifically for this area includes, modeling slots as hierarchical concepts (Zhu and Yu, 2018) and using ensembles of models trained on data-rich domains (Gašić et al., 2015; Kim et al., 2017; Jha et al., 2018).

The availability of task descriptions has made zero-shot learning (Norouzi et al., 2013; Socher et al., 2013) popular. In particular, work on zero-shot utterance intent detection has relied on varied resources such as click logs (Dauphin et al., 2013) and manually defined domain ontologies (Kumar et al., 2017), as well as models such as deep structured semantic models (Chen et al., 2016) and capsule networks (Xia et al., 2018). Zero-shot semantic parsing is addressed in Krishnamurthy et al. (2017) and Herzig and Berant (2018) and specifically for SLU utilizing external resources such as label ontologies in Ferreira et al. (2015a,b) and handwritten intent attributes in Yazdani and Henderson (2015); Chen et al. (2015). Our work is closest in spirit to Bapna et al. (2017) and Lee and Jha (2018), who employ textual slot descriptions to scale to unseen intents/slots. Since slots tend to take semantically similar values across utterances, we augment our model with example values, which are easier for developers to define than manual alignments across schemas (Li et al., 2011).

3 Problem Statement

We frame our conditional sequence tagging task as follows: given a user utterance with T tokens and a slot type, we predict inside-outside-begin (IOB) tags $\{y_1, y_2 \dots y_T\}$ using 3-way classification per token, based on if and where the provided slot type occurs in the utterance. Figure 3 shows IOB tag sequences for one positive (slot *service*, present in the utterance) and one negative (slot *timeRange*, not present in the utterance) instance each.

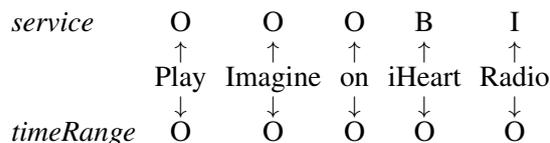


Figure 3: Example semantic frame with IOB slot annotations for a positive and a negative instance.

4 Model Architecture

Figure 2 illustrates our model architecture where a user utterance is tagged for a provided slot. To represent the input slot, along with a textual slot description as in Bapna et al. (2017), we supply a small set of example values for this slot, to provide a more complete semantic representation.¹ Detailed descriptions of each component follow.

Inputs: We use as input d_{wc} -dimensional embeddings for 3 input types: T user utterance tokens $\{u_i \in \mathbb{R}^{d_{wc}}, 1 \leq i \leq T\}$, S input slot description tokens $\{d_i \in \mathbb{R}^{d_{wc}}, 1 \leq i \leq S\}$, and K example values for the slot, with the N_k token embedding for the k^{th} example denoted by $\{e_i^k \in \mathbb{R}^{d_{wc}}, 1 \leq i \leq N_k\}$.

Utterance encoder: We encode the user utterance using a d_{en} -dimensional bidirectional GRU recur-

¹Note that the slot description is still needed since example slot values alone cannot distinguish slots which take semantically similar values (e.g. *departDate* vs *returnDate*).

Intent	Slot Names (Training and Evaluation)	
AddToPlaylist	artist, entityName, musicItem, playlist, playlistOwner	
BookRestaurant	city, cuisine, partySizeNumber, restaurantName, restaurantType, servedDish, spatialRelation, state...	
GetWeather	city, conditionDescription, country, geographicPoi, spatialRelation, state, timeRange...	
PlayMusic	album, artist, genre, musicItem, playlist, service, sort, track, year	
RateBook	bestRating, objectName, objectPartOfSeriesType, objectSelect, objectType, ratingUnit, ratingValue	
SearchCreativeWork	objectName, objectType	
FindScreeningEvent	locationName, movieName, movieType, objectLocationType, objectType, spatialRelation, timeRange	

Intent	Training Slot Names	Evaluation Slot Names
BookBus	from, to, leaving, returning, travelers, tripType, departureTime	from, to, departOn, addReturnTrip, tripType, promoCode, discountOptions, children, adults, seniors
FindFlights	from, to, depart, return, cabinClass, flightType	depart, arrive, departDate, returnDate, searchType, promoCode
BookRoom	where, checkIn, checkOut, guests, homeType, propertyType, priceRange, amenities	location, hotelName, checkIn, checkOut, rooms, roomType, pricePerNight, rating, amenities

Table 1: Intents and training/evaluation slot schemas for SNIPS (top) and XSchema (bottom) datasets.

rent neural network (RNN) (Chung et al., 2014). We denote the set of per-token RNN hidden states by $H = \{h_i \in \mathbb{R}^{d_{en}}, 1 \leq i \leq T\}$, which are used as contextual utterance token encodings.

$$H = BiGRU(\{u_i, 1 \leq i \leq T\}) \quad (1)$$

Slot description encoder: We obtain an encoding $d^s \in \mathbb{R}^{d_{wc}}$ of the slot description by mean-pooling the embeddings for the S slot description tokens.

$$d^s = \frac{1}{S} \sum_{i=1}^S d_i \quad (2)$$

Slot example encoder: We first obtain encodings $\{e_k^x \in \mathbb{R}^{d_{wc}}, 1 \leq k \leq K\}$ for each slot example value by mean-pooling the N_k token embeddings. Then, we compute an attention weighted encoding of all K slot examples $\{e_i^a \in \mathbb{R}^{d_{wc}}, i \leq 1 \leq T\}$ for each utterance token, with the utterance token encoding as attention context. Here, $\alpha_i^x \in \mathbb{R}^K$ denotes attention weights over all K slot examples corresponding to the i^{th} utterance token, obtained with general cosine similarity (Luong et al., 2015).

$$e_k^x = \frac{1}{N_k} \sum_{i=1}^{N_k} e_i^k, 1 \leq k \leq K \quad (3)$$

$$\alpha_i^x = softmax(\{h_i \mathbf{W}_a e_k^x \forall k\}), 1 \leq i \leq T \quad (4)$$

$$e_i^a = \sum_{k=1}^K \alpha_{i_k}^x \times e_k^x \quad (5)$$

Tagger: We feed the concatenated utterance, slot description and example encodings to a d_{en} -dimensional bidirectional LSTM. The output hidden states $X = \{x_i \in \mathbb{R}^{d_{en}}, 1 \leq i \leq T\}$ are used for a 3-way IOB tag classification per token.

$$X = BiLSTM(\{h_i \oplus d^s \oplus e_i^a, 1 \leq i \leq T\}) \quad (6)$$

$$y_i = softmax(\mathbf{W}_t x_i + \mathbf{b}_t), 1 \leq i \leq T \quad (7)$$

Parameters: We use fixed $d_w=128$ -dim pre-trained word embeddings² for all tokens. We also train per-character embeddings, fed to a 2-layer convolutional neural network (Kim, 2014) to get a $d_c=32$ -dim token embedding. For all inputs, the $d_{wc}=160$ -dim final embedding is the concatenation of the word and char-CNN embeddings. The RNN encoders have hidden state size $d_{en}=128$. All trainable weights are shared across intents and slots. The model relies largely on fixed word embeddings to generalize to new intents/slots.

5 Datasets and Experiments

In this section we describe the datasets used for evaluation, baselines compared against, and more details on the experimental setup.

Datasets: In order to evaluate cross-domain transfer learning ability and robustness to misaligned schemas, respectively, we use the following two SLU datasets to evaluate all models.

- **SNIPS:** This is a public SLU dataset (Coucke et al., 2018) of crowdsourced user utterances with 39 slots across 7 intents and ~ 2000 training instances per intent. Since 11 of these slots are shared (see Table 1), we use this dataset to evaluate cross-domain transfer learning.
- **XSchema:** This is an in-house crowdsourced dataset with 3 intents (500 training instances each). Training and evaluation utterances are annotated with different schemas (Table 1) from real web forms to simulate misaligned schemas.

Baselines: We compare with two strong zero-shot baselines: Zero-shot Adaptive Transfer (ZAT) (Lee and Jha, 2018) and Concept Tagger

²<https://tfhub.dev/google/nlm-en-dim128/1>

Target training e.g.	0			50			
Intent ↓ Model →	CT	ZAT	+2Ex	LSTM	CT	ZAT	+10Ex
AddToPlaylist	53.3	46.8	55.2	59.4	74.4	73.4	76.2*
BookRestaurant	45.7	46.6	48.6*	57.5	63.8	63.5	63.6
GetWeather	63.5	60.7	66.0*	75.7	72.1	71.1	77.5*
PlayMusic	28.7	30.1	33.8*	49.3	56.4	56.0	58.8
RateBook	24.5	31.0	28.5	85.1*	82.9	83.8	82.2
SearchCreativeWork	24.7	26.7	26.2	52.9	62.8	63.7	65.9
FindScreeningEvent	23.7	19.7	25.5*	60.8	64.9	64.6	67.0*
Average	37.7	37.4	40.6*	62.8	68.2	68.0	70.1*

Table 2: Slot F1 scores for baselines (CT, ZAT, LSTM) and our best models (with 2 slot values for zero-shot and 10 values for 50 train instances) on SNIPS. Rows represent different train-test splits, defined in Section 5. Our model consistently outperforms the baselines, with $\sim 3\%$ absolute gain in the zero-shot setting.³

(CT) (Bapna et al., 2017), in addition to a 2-layer multi-domain bidirectional LSTM baseline (Hakkani-Tür et al., 2016) for non-zero-shot setups. ZAT and CT condition slot filling only on slot descriptions, with ZAT adding slot description attention, char embeddings and CRFs on top of CT. Since labor-intensive long text descriptions are unavailable for our data, we use tokenized slot names in their place, as in Bapna et al. (2017).

Experimental Setup: We use SNIPS to test zero/few-shot transfer: for each target intent I , we train on all ~ 2000 training instances from intents other than I , and varying amounts of training data for I , evaluating exclusively on I . For XSchema, we train and evaluate on a single intent, specifically evaluating cross-schema performance.

We sample positive and negative instances (Figure 3) in a ratio of 1:3. Slot values input during training and evaluation are randomly picked from values taken by the input slot in the relevant domain’s training set, excluding ones that are also present in the evaluation set. In practice, it is usually easy to obtain such example values for each slot either using automated methods (such as crawling from existing web forms) or have them be provided as part of the slot definition, with negligible extra effort.

To improve performance on out-of-vocabulary entity names, we randomly replace slot value tokens in utterances and slot examples with a special token, and raise the replacement rate linearly from 0 to 0.3 during training (Rastogi et al., 2018).

The final cross-entropy loss, averaged over all utterance tokens, is optimized using ADAM (Kingma and Ba, 2014) for 150K training steps.

Target training e.g.	0			50		
Intent ↓ Model →	CT	ZAT	+10Ex	CT	ZAT	+10Ex
BookBus	70.9	70.1	74.1*	86.8	85.2	89.4
FindFlights	43.5	44.8	53.2*	62.3	59.7	69.2*
BookRoom	23.6	23.4	33.0*	49.7	52.1	58.7*

Table 3: Slot F1 scores on the XSchema dataset⁴. We train and evaluate on a single intent, but with different schemas, thus precluding the LSTM baseline.

Slot F1 score (Sang and Buchholz, 2000) is our final metric, reported after 3-fold cross-validation.

6 Results

For the SNIPS dataset, Table 2 shows slot F1 scores for our model trained with randomly-picked slot value examples in addition to slot descriptions vis-à-vis the baselines. Our best model consistently better than the zero-shot baselines CT and ZAT, which use only slot descriptions, overall and individually for 5 of 7 intents. The average gain over CT and ZAT is $\sim 3\%$ in the zero-shot case. In the low-data setting, all zero-shot models gain $\geq 5\%$ over the multi-domain LSTM baseline (with the 10-example-added model further gaining $\sim 2\%$ on CT/ZAT). All models are comparable when all target data is used for training, with F1 scores of 87.8% for the LSTM, and 86.9% and 87.2% for CT and our model with 10 examples respectively.

Table 3 shows slot F1 scores for XSchema data. Our model trained with 10 example values is robust to varying schemas, with gains of $\sim 3\%$ on *BookBus*, and $\sim 10\%$ on *FindFlights* and *BookRoom* in the zero-shot setting.

For both datasets, as more training data for the target domain is added, the baselines and our approach perform more similarly. For instance, our approach improves upon the baseline by $\sim 0.2\%$ on SNIPS when 2000 training examples are used for the target domain, affirming that adding example values does not hurt in the regular setting.

Results by slot type: Example values help the most with limited-vocabulary slots not encountered during training: our approach gains $\geq 20\%$ on slots such as *conditionDescription*, *bestRating*, *service* (present in intents *GetWeather*, *RateBook*, *PlayMusic* respectively). Intents *PlayMusic* and *GetWeather*, with several limited-vocabulary slots, see significant gains in the zero-shot setting.

³Asterisk (*) indicates a statistically significant gain over the second-best model as per McNemar’s test ($p < 0.05$).

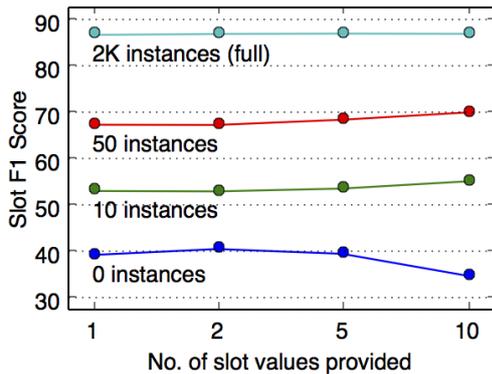


Figure 4: Variation of overall slot F1 score with number of slot value examples input to the model, with varying number of target intent training instances for SNIPS.

For compositional open-vocabulary slots (*city*, *cuisine*), our model also compares favorably - e.g. 53% vs 27% slot F1 for unseen slot *cuisine* (intent *BookRestaurant*) - since the semantic similarity between entity and possible values is easier to capture than between entity and description.

Slots with open, non-compositional vocabularies (such as *objectName*, *entityName*) are hard to infer from slot descriptions or examples, even if these are seen during training but in other contexts, since utterance patterns are lost across intents. All models are within 5% slot F1 of each other for such slots. This is also observed for unseen open-vocabulary slots in the XSchema dataset (such as *promoCode* and *hotelName*).

For XSchema experiments, our model does significantly better on slots which are confusing across schemas (evidenced by gains of >20% on *depart* in *FindFlights*, *roomType* in *BookRoom*).

Effect of number of examples: Figure 4 shows the number of slot value examples used versus performance on SNIPS. For the zero-shot case, using 2 example values per slot works best, possibly due to the model attending to perfect matches during training, impeding generalization when more example values are used. In the few-shot and normal-data settings, using more example values helps accuracy, but the gain drops with more target training data. For XSchema, in contrast, adding more example values consistently improves performance, possibly due to more slot name mismatches in the dataset. We avoid using over 10 example values, in contrast to prior work (Krishnamurthy et al., 2017; Naik et al., 2018) since it may be infeasible to easily provide or extract a large number of values for unseen slots.

Ablation: Slot replacement offsets overfitting in our model, yielding gains of 2–5% for all models incl. baselines. Fine-tuning the pretrained word embeddings and removing character embeddings yielded losses of ~1%. We tried more complex phrase embeddings for the slot description and example values, but since both occur as short phrases in our data, a bag-of-words approach worked well.

Comparison with string matching: A training and evaluation setup including example values for slots may lend itself well to adding string matching-based slot fillers for suitable slots (for example, slots taking numeric values or having a small set of possible values). However, this is not applicable to our exact setting since we ensure that the slot values to be tagged during evaluation are never provided as input during training or evaluation. In addition, it is difficult to distinguish two slots with the same expected semantic type using such an approach, such as for slots *ratingValue* and *bestRating* from SNIPS intent *RateBook*.

7 Conclusions and Future Work

We show that extending zero-shot slot filling models to use a small number of easily obtained example values for slots, in addition to textual slot descriptions, is a scalable solution for zero/few-shot slot filling tasks on similar and heterogenous domains, while resistant to misaligned overlapping schemas. Our approach surpasses prior state-of-the-art models on two multi-domain datasets.

The approach can, however, be inefficient for intents with many slots, as well as potentially sacrificing accuracy in case of overlapping predictions. Jointly modeling multiple slots for the task is an interesting future direction. Another direction would be to incorporate zero-shot entity recognition (Guerini et al., 2018), thus eliminating the need for example values during inference.

In addition, since high-quality datasets for downstream tasks in dialogue systems (such as dialogue state tracking and dialogue management) are even more scarce, exploring zero-shot learning approaches to these problems is of immense value in building generalizable dialogue systems.

Acknowledgements

We would like to thank Ankur Bapna for the insightful discussions that have notably shaped this work. We would also like to thank the Deep Dialogue team at Google Research for their support.

References

- Ankur Bapna, Gökhan Tür, Dilek Hakkani-Tür, and Larry P. Heck. 2017. [Towards zero-shot frame semantic parsing for domain scaling](#). In *Interspeech 2017, 18th Annual Conference of the International Speech Communication Association, Stockholm, Sweden, August 20-24, 2017*.
- Jerome R Bellegarda. 2014. Spoken language understanding for natural interaction: The siri experience. In *Natural Interaction with Robots, Knowbots and Smartphones*, pages 3–14. Springer.
- 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.
- Xilun Chen, Yu Sun, Ben Athiwaratkun, Claire Cardie, and Kilian Weinberger. 2018. [Adversarial deep averaging networks for cross-lingual sentiment classification](#). *Transactions of the Association for Computational Linguistics*, 6:557–570.
- Yun-Nung Chen, Dilek Hakkani-Tür, and Xiaodong He. 2016. Zero-shot learning of intent embeddings for expansion by convolutional deep structured semantic models. In *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*, pages 6045–6049. IEEE.
- Yun-Nung Chen, William Yang Wang, Anatole Gershan, and Alexander Rudnicky. 2015. Matrix factorization with knowledge graph propagation for unsupervised spoken language understanding. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, volume 1, pages 483–494.
- Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.
- 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*.
- Yann N Dauphin, Gokhan Tur, Dilek Hakkani-Tur, and Larry Heck. 2013. Zero-shot learning for semantic utterance classification. *arXiv preprint arXiv:1401.0509*.
- Emmanuel Ferreira, Bassam Jabaian, and Fabrice Lefevre. 2015a. Online adaptative zero-shot learning spoken language understanding using word-embedding. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5321–5325. IEEE.
- Emmanuel Ferreira, Bassam Jabaian, and Fabrice Lefevre. 2015b. Zero-shot semantic parser for spoken language understanding. In *Sixteenth Annual Conference of the International Speech Communication Association*.
- M Gašić, N Mrkšić, Pei-hao Su, David Vandyke, Tsung-Hsien Wen, and Steve Young. 2015. Policy committee for adaptation in multi-domain spoken dialogue systems. In *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pages 806–812. IEEE.
- Anuj Goyal, Angeliki Metallinou, and Spyros Matsoukas. 2018. Fast and scalable expansion of natural language understanding functionality for intelligent agents. *arXiv preprint arXiv:1805.01542*.
- 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, Brussels, Belgium. Association for Computational Linguistics.
- D. Hakkani-Tür, G. Tur, A. Celikyilmaz, Y.-N. Chen, J. Gao, L. Deng, and Y.-Y. Wang. 2016. Multi-domain joint semantic frame parsing using bi-directional rnn-lstm. In *Proceedings of Interspeech*.
- Matthew Henderson, Blaise Thomson, and Jason D Williams. 2014. The third dialog state tracking challenge. In *2014 IEEE Spoken Language Technology Workshop (SLT)*, pages 324–329. IEEE.
- Jonathan Herzig and Jonathan Berant. 2018. Decoupling structure and lexicon for zero-shot semantic parsing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1619–1629.
- A. Jaech, L. Heck, and M. Ostendorf. 2016. Domain adaptation of recurrent neural networks for natural language understanding. In *Proceedings of Interspeech*.
- Rahul Jha, Alex Marin, Suvamsh Shivaprasad, and Imed Zitouni. 2018. Bag of experts architectures for model reuse in conversational language understanding. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 3 (Industry Papers)*, volume 3, pages 153–161.

- Yoon Kim. 2014. [Convolutional neural networks for sentence classification](#). In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751. Association for Computational Linguistics.
- Young-Bum Kim, Karl Stratos, and Dongchan Kim. 2017. Domain attention with an ensemble of experts. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 643–653.
- Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Jayant Krishnamurthy, Pradeep Dasigi, and Matt Gardner. 2017. Neural semantic parsing with type constraints for semi-structured tables. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1516–1526.
- Anjishnu Kumar, Pavankumar Reddy Muddireddy, Markus Dreyer, and Björn Hoffmeister. 2017. [Zero-shot learning across heterogeneous overlapping domains](#). In *Interspeech 2017, 18th Annual Conference of the International Speech Communication Association, Stockholm, Sweden, August 20-24, 2017*, pages 2914–2918.
- Gakuto Kurata, Bing Xiang, Bowen Zhou, and Mo Yu. 2016. Leveraging sentence-level information with encoder lstm for semantic slot filling. *arXiv preprint arXiv:1601.01530*.
- Sungjin Lee and Rahul Jha. 2018. Zero-shot adaptive transfer for conversational language understanding. *arXiv preprint arXiv:1808.10059*.
- Xiao Li, Ye-Yi Wang, and Gokhan Tur. 2011. Multi-task learning for spoken language understanding with shared slots. In *Twelfth Annual Conference of the International Speech Communication Association*.
- Bing Liu and Ian Lane. 2017. Multi-domain adversarial learning for slot filling in spoken language understanding. *arXiv preprint arXiv:1711.11310*.
- Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421.
- Grégoire Mesnil, Yann Dauphin, Kaisheng Yao, Yoshua Bengio, Li Deng, Dilek Hakkani-Tur, Xiaodong He, Larry Heck, Gokhan Tur, Dong Yu, et al. 2015. Using recurrent neural networks for slot filling in spoken language understanding. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(3):530–539.
- Chetan Naik, Arpit Gupta, Hancheng Ge, Mathias Lambert, and Ruhi Sarikaya. 2018. Contextual slot carryover for disparate schemas. *Proc. Interspeech 2018*, pages 596–600.
- Mohammad Norouzi, Tomas Mikolov, Samy Bengio, Yoram Singer, Jonathon Shlens, Andrea Frome, Greg S Corrado, and Jeffrey Dean. 2013. Zero-shot learning by convex combination of semantic embeddings. *arXiv preprint arXiv:1312.5650*.
- 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.
- Abhinav Rastogi, Raghav Gupta, and Dilek Hakkani-Tur. 2018. Multi-task learning for joint language understanding and dialogue state tracking. In *Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue*, pages 376–384.
- Erik F Tjong Kim Sang and Sabine Buchholz. 2000. Introduction to the conll-2000 shared task: Chunking. In *Proceedings of the 2nd workshop on Learning language in logic and the 4th conference on Computational natural language learning-Volume 7*, pages 127–132. Association for Computational Linguistics.
- Darsh Shah, Tao Lei, Alessandro Moschitti, Salvatore Romeo, and Preslav Nakov. 2018. [Adversarial domain adaptation for duplicate question detection](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1056–1063, Brussels, Belgium. Association for Computational Linguistics.
- Aditya Siddhant, Anuj Goyal, and Angeliki Metallinou. 2018. Unsupervised transfer learning for spoken language understanding in intelligent agents. *arXiv preprint arXiv:1811.05370*.
- 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.
- Congying Xia, Chenwei Zhang, Xiaohui Yan, Yi Chang, and Philip Yu. 2018. [Zero-shot user intent detection via capsule neural networks](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3090–3099. Association for Computational Linguistics.
- Majid Yazdani and James Henderson. 2015. [A model of zero-shot learning of spoken language understanding](#). In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 244–249. Association for Computational Linguistics.
- Steve Young. 2002. Talking to machines (statistically speaking). In *Seventh International Conference on Spoken Language Processing*.
- 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.