In Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue, Stockholm, Sweden, September 11-13, 2019. pp. 411-417. 2019.

Multi-Task Learning for Supervised Pretraining of Goal-Oriented Dialogue Policies

Sarah McLeod[1], **Ivana Kruijff-Korbayová**[2], **Bernd Kiefer**[2]

[1]Saarland University, Saarbrücken, Germany
[2]German Research Center for Artificial Intelligence (DFKI), Saarbrücken, Germany
smcleod@coli.uni-saarland.de
{ivana.kruijff, bernd.kiefer}@dfki.de

Abstract

This paper describes the use of Multi-Task Neural Networks (NNs) for system dialogue act selection. These models leverage the representations learned by the Natural Language Understanding (NLU) unit to enable robust initialization/bootstrapping of dialogue policies from medium sized initial data sets. We evaluate the models on two goal-oriented dialogue corpora in the travel booking domain. Results show the proposed models improve over models trained without knowledge of NLU tasks.

1 Introduction

To be successful, goal-oriented dialogue systems must accurately determine the intent(s) of a user, identify and understand the relevant information they have provided, and based on that information, select the appropriate response at each turn in the conversation. One way to model conversation is as a partially observable Markov decision process (Young et al., 2013). In this framework system response generation is modeled as a stochastic policy, and research into statistically optimizing dialogue policies with Reinforcement Learning (RL) is an active area of research (Gasic and Young, 2014; Lemon and Pietquin, 2007). However, learning optimal dialogue policies with RL can be challenging since large state and action spaces require large amounts of training data to densely sample the space (Lemon and Pietquin, 2007; Wen et al., 2016; Li et al., 2017). Additionally, networks trained with RL learn in a trial-and-error process, guided by a potentially delayed reward function. This exploration process can lead to poor performance in the early training stages, which in turn can lead to a negative user experience (Su et al., 2016).

To address these issues supervised learning has been used for pre-training of dialogue policies (Su et al., 2016; Henderson et al., 2007; Williams and Zweig, 2016), however the previous approaches only considered one aspect of dialogue during training. Grosz and Sidner (1986) describe discourse structure as a composite of multiple aspects that interact and co-constrain one other. This structure determines the meaning of a discourse and provides a framework for processing dialogue. The question then arises whether it would be beneficial to view dialogue policy training as a multi-task learning (MTL) problem. MTL is an active area of research and has been shown to improve performance on a number Natural Language Processing (NLP) tasks (Ruder, 2017; Zhang and Yang, 2017). In this work we propose a method to use the training signals of related tasks during supervised pre-training of system dialogue act selection as part of dialogue policy initialization. We also experiment with multiple architectures across two data sets and evaluate against two basline architectures.

Specifically, we use slot-filling and user-intent classification as auxiliary tasks for the primary task of system dialogue act selection. For many corpus trained dialogue systems slot-filling and user-intent classification are trained independently, separate from the dialogue manager. We hypothesize that the features learned when training neural models for these tasks are also informative for the initialization of a robust dialogue policy network. In MTL there can be an added cost of collecting labels for auxiliary tasks, but in this scenario the labels for user-intent and slot-filling are needed to develop a complete dialogue system already exist; the framework we propose uses these labels as additional information to initialize the dialogue manager. The next sections describe related work in MTL, including MTL for goal-oriented dialogue systems, the corpora used in our experiments, the architecture of the neural models

Corpus	# Slots	# Intents	# Actions	Avg. UL	# Train	# Dev	# Test	Vocab
ATIS	79	17	NA	11.13	4478	500	894	900
Frames	21	45	52	8.05	6131	1532	1916	3249
COMM ATT	34	10	56	1.86	3960	442	781	545
COMM BBN	37	18	48	2.30	3168	351	622	490
COMM CMU	45	32	72	2.58	2793	310	548	580
COMM SRI	37	17	25	2.44	4076	452	800	569

Table 1: The number of slot types, user speech acts, and system dialogue acts for each corpus, as well as the average length of the user input utterances.

we tested, and the results of the evaluation.

2 Related work

Multi-Task Learning: In MTL the training signals of related tasks are used to learn features that are relevant to multiple tasks, including a primary task of interest. In learning these shared features the model learns a representation that improves generalization on that primary task. Caruana (1998) and Zhang and Yang (2017) describe a number of tasks where the shared representation learned with MTL improves generalization. MTL has also been shown to improve a number NLP tasks (Toshniwal et al., 2017; Arik et al., 2017; Dong et al.; Zoph and Knight, 2016; Johnson et al., 2016). See Ruder (2017) and Zhang and Yang (2017) for additional examples of MTL for NLP.

Goal-Oriented Dialogue systems: Wen et al. (2016) treat dialogue as a sequence to sequence mapping problem and design a dialogue manager where each component is modularly connected and trainable from data. Previous work also learns state-tracking and other NLU tasks simultaneously. Hakkani-Tur et al. (2016) use a bi-directional LSTM to jointly model slot filling, intent determination, and domain classification for different domains. Chen et al. (2016) use a knowledge-guided structural attention network (K-SAN) to model intent prediction and slot filling simultaneously. Both published results on the ATIS corpus (Price, 1990). Padmakumar et al. (2017) train a semantic parser and policy network in batches, giving the policy network access to the updated semantic parser after every batch. Zhao and Eskenazi (2016) jointly learn policies for state tracking and dialogue strategies using Deep Recurrent Q-Network (DRQN). Li et al. (2017) use a single RNN with LSTM to jointly learn user intent as well as slot filling. Their dialogue manager is initialized by supervised learning of labels generated by a rule system, then end-to-end training is continued with RL using a user simulator. Results were published on data from movie-ticket booking domain. We also propose a method to initialize the dialogue manager with supervised learning, however we use the information from upstream dialogue system tasks during supervised pre-training. We also experiment on two distinct corpora in the travel planning domain across multiple architectures.

3 Data

We evaluated our models on three corpora: the Maluuba Frames (El Asri et al., 2017), DARPA COMMUNICATOR (Georgila et al., 2009, 2005) and ATIS (Price, 1990) data sets. The Frames corpus is a collection of human-human dialogues that captures realistic behaviors in natural conversations. The DARPA COMMUNICATOR corpus is a collection of human-computer interactions from users calling into the COMMUNICATOR travel planning system. We use the version described in (Georgila et al., 2009, 2005), which includes annotations from the original corpus plus additional user-intent and task level annotations automatically added by a system they designed. The complete COMMUNICATOR corpus includes data for all systems evaluated as part of the DARPA program. As in Henderson et al. (2007) we use only the data from the ATT, BBN, CMU and SRI systems. The ATIS corpus is a collection of spontaneous speech and associated annotations, collected wizard-of-oz style. The corpus was included in the software released by Hakkani-Tur et al. (2016) and was used to facilitate comparison to their work. The number of unique labels for each task as well as the train, dev and test data splits for each corpus are listed in Table 1.

3.1 Preprocessing

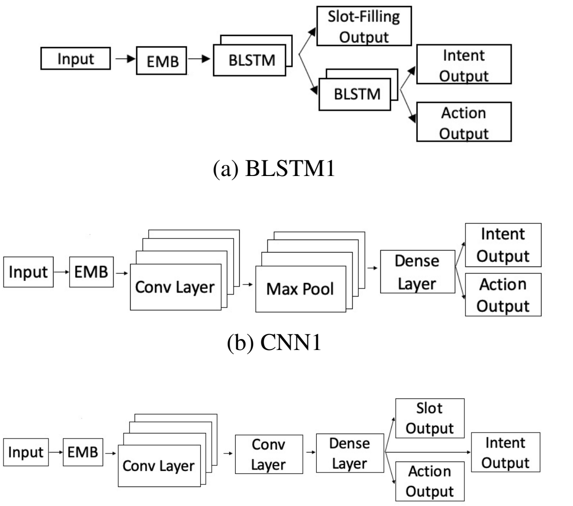
We used the common IOB (in-out-begin) format to annotate slot-tags for each token. In this schema, for each input sequence X tokens $t_1, ..., t_n$ are assigned a slot label $s_1, ..., s_n$ and multi-token values are labeled with B (begin) and I (inside) to indicate the extent of the tokens that fill that slot. Tokens that are not relevant to any slot are tagged with O (outside). Some turns in the Frames and COMMUNICATOR corpora were labeled with duplicate user-intent labels and system action labels. One option was to ignore these duplicate labels, however these duplicates occurred frequently enough to be considered informative; therefore when more than one class label exists for a single input utterance, we concatenated all of the labels into a single label. For example, if the system dialogue act was annotated with *negate*, *negate*, and *inform* the labels are concatenated to create a single *negate#negate#inform* label.

4 Experiments

We completed three sets of experiments: two baseline experiments and a final experiment with the multi-task architecture. Each of these experiments included three tasks: slot-filling, framed as sequence prediction, user-intent classification, and system dialogue act selection. In the first baseline experiment the models described in Hakkani-Tur et al. (2016) were extended to new corpora and new tasks using the software released by the authors. In the second baseline experiment we trained single-task models for each of the three tasks individually, on each corpus. Following the methodology suggested in Caruana (1998), these models were tuned for each corpus and architecture. The Maluuba Frames and DARPA COMMUNICATOR Corpora were used in baseline and multi-task experiments; the ATIS corpus does not contain annotations for system dialogue act selection and was therefore only used in baseline experiments.

4.1 Architectures

Baseline A: Hakkani-Tur et al. (2016) describe a recurrent neural network (RNN) architecture for simultaneous learning of slot-filling, domain classification, and user intent classification. They treat joint learning as a sequence labeling task and use a modification of the encoder-decoder model. To represent the data they use the IOB style annotations for slots and for each utterace U associate the sentence final token with a single label generated by concatenating the associated domain d and user-intent u labels. In this framework the input and output utterances become:



(a) BLSTM1

(b) CNN1

(c) CNN2

Figure 1: Model architectures for the multi-task experiments.

$$X = t_1, \ldots, t_n, < EOS >$$

$$Y = s_1, \ldots, s_n, d_u\_u_u$$

The model weights are learned by maximizing the conditional likelihood of the training set labels.

In our first baseline experiment we use this architecture to jointly learn user-intent classification, slot-filling, and system dialogue act selection (replacing domain classification) on the Frames and COMMUNICATOR corpora. In our experiments the sentence final token is created by concatenating the user-intent and system dialogue act labels.

Baseline B: Next we trained Bi-directional LSTM (BLSTM) and Convolutional Neural Network (CNN) single-task models to perform each task individually. The BLSTM consisted of an input layer, hidden layer, and output layer. Softmax is used to produce a distribution (p_t) of likely labels at each time-step . The final output is then $argmax(p_t)$. The CNN network consists of two convolutional layers, connected in series, each followed by max pooling layers. A dense layer connects the output of the final convolutional layer to the softmax layer. For slot-filling the models predict a label for each word in the input sequence. For user-intent and system dialogue act selection the models predict a single label for the input utterance. The BLSTM architecture was used to train individual models for all three tasks. The CNN architecture was used to train individual user-intent and system dialogue act selection models only.

Multi-Task Models: Lastly, we created multi-task models with BLSTMs and CNNs, and a combination of the two. In these architectures each task has a separate output, and all tasks share hidden layers. We implemented three BLSTM versions. BLSTM1 consists of two stacked BLSTMs and the slot-filling output layer is positioned as an auxiliary output at the first BLSTM. For BLSTM1 the loss for slot-filling is backprogagated through the first BLSTM. The loss for user-intent and system dialogue act selection is backprogagated through both BLSTM layers. Figure 1a illustrates this architecture. BLSTM2 uses the BLSTM1 architecture plus a skip connection from the embedding layer to the second BLSTM layer. In BLSTM3 the first BLSTM layer weights are initialized with the weights learned when training slot-filling alone. The intent was to explore the possible benefit of transfer learning from a previously trained model. Experiments on the subsets of the COMMUNICATOR corpus with BLSTM3 include model training where the weights of the first BLSTM layer are initialized with the weights learned on the Frames data (BLSTM3b). Finally, ablation testing was also done to explore the influence of each auxiliary task. The BLSTM1 model was trained on all three tasks simultaneously (BLSTM1a), on slot-filling and the primary task alone (BLSTM1b), and on user-intent classification and the primary task alone (BLSTM1c).

The CNN1 network design was inspired by Yoon (2014) and is illustrated in Figure 1b. This network uses 4 filters of different widths each followed by max pooling over time. Filter widths, the number of feature maps, and the number of nodes in the fully connected layer were chosen based on the suggestions of Zhang and Wallace (2015). Early experiments on the BLSTM networks showed a potential benefit to using user-intent classification alone as an auxiliary task, therefore these experiments used only user-intent classification as the auxiliary task.

We also conducted experiments with networks inspired by Google's Inception architecture (Szegedy et al., 2014). This is a general purpose architecture where the output from multiple convolutional layers is passed to a single convolutional layer, called a bottle-neck, which constrains the number of features that subsequent layers take as input, keeping the number of parameters low while retaining the expressive power of the network. Our architecture is illustrated in Figure 1c. This network uses 5 convolutional layers of different filter widths followed by a single bottle-neck convolutional layer. The CNN2b network is composed of three CNN2a networks concatenated together.

Corpus	best F	Avg F
ATIS	95.48%	94.70%
Frames	74.26%	73.05%
COMMUNICATOR ATT	48.17%	45.98%
COMMUNICATOR BBN	50.34%	48.77%
COMMUNICATOR CMU	53.96%	52.59%
COMMUNICATOR SRI	59.74%	58.55%

Table 2: The best F-measure and average F-measure on slot-filling alone for each corpus using the architecture released by Hakkani-Tur et al. (2016).

The final multi-task network is a hybrid CNN + BLSTM architecture. In this network the input is connected to a CNN network of three convolutional layers with different filter widths each followed by max pooling. This is then connected to the BLSTM1 architecture. The goal was to explore the possibility of extracting features with a CNN layer that could then be used by the BLSTM1 network.

4.2 Training

All network development and training was done in Keras (Chollet et al., 2015) and the code will be released with the final version of this paper. We experimented with batch sizes of 15, 25, 50 and 100, hidden layers of 25, 50 and 100 units, and drop-out ratios of 0,0.25, and 0.5 on the fully-connected layers. GloVe (Pennington et al., 2014) word embeddings were used as pre-trained word embeddings. The Adam optimizer was used with a learning rate of 0.001. All weights were initialized with glorot uniform. The BLSTM layers used tanh as the activation function. During training the validation loss was monitored and early stopping was used to prevent over-fitting.

5 Evaluation

Table 2 shows the best and average F-measure for slot-filling alone on each corpus using the architecture released by Hakkani-Tur et al. (2016). Both best and average F-measure were calculated on the held-out test set, where the average was calculated over 10 different weight initilizations.

Model	Frames	ATT	BBN	CMU	SRI
Baseline A	34.77%	52.55%	40.59%	36.11%	57.72%
BLSTM Baseline	36.26%	52.56%	43.52%	37.06%	57.36%
BLSTM1a	36.08%	52.52%	**43.71%**	**37.49%**	**58.12%**
BLSTM1b	35.88%	52.31%	42.52%	**38.06%**	57.32%
BLSTM1c	**37.93%***	**53.12%**	42.96%	36.58%	57.21%
BLSTM2	**36.29%**	51.74%	**43.56%**	**37.55%**	**58.61%**
BLSTM3	**37.49%***	52.13%	42.27%	**37.34%**	**58%**
BLSTM3b	NA	52.29%	43.19%	**38.01%**	**58.30%**
CNN Baseline	35.37%	46.93%	43.93%	36.44%	57.39%
CNN1	34.59%	**51.90%***	43.93%	**36.73%**	**59.09%**
CNN2a	32.78%	**52.62%***	43.29%	**37.59%**	56.09%
CNN2b	32.80%	**53.28%***	43.74%	**38.47%***	56%
CNN+BLSTM	**36.82%***	**51.38%***	43.73%	**38.53%***	**58.49%**

Table 3: The best F-measure achieved for each multi-task model on the system action classification task. Results in bold indicate an improvement over the associated single-task baseline (BLSTM or CNN baseline). An asterisk indicates a statistically significant improvement over the respective baseline.

Hakkani-Tur et al. (2016) experimented with multiple LSTM and BLSTM models, but noted that comparable results were achieved on each and therefore only report results on the BLSTM models. We do the same and only report on experiments with their BLSTM architecture. The results on the ATIS corpus are the metrics reported by the authors (and confirmed by us).

For each corpus many of the multi-task models achieved a higher metric score than the Baseline B models on the test data, however significance testing showed not all of these improvements were statistically significant. Significance testing was done with randomized approximation (Yeh, 2000). Table 3 lists the best F-measure values for each model for the primary task of system action selection.

The majority of the multi-task models, as well as the Baseline B models on the Frames, BBN, and SRI corpora, achieved a higher F-measure than the Baseline A models. (We did not test for statistical significance between the MTL models and the Baseline A). The multi-task CNN models showed statistically significant improvement on three data sets and were faster to train than the BLSTM models, even when larger. Half of the BLSTM models achieved significant improvement on the Frames corpus, but improvement was more sporadic on the COMMUNICATOR corpus. In the Frames corpus most input utterances are much longer since the user provides significant context at each turn. In the COMMUNICATOR corpus after the initial request most user utterances are limited to one or two word responses to questions presented by the system. This creates a dialogue that looks more like a system initiative dialogue, as compared to the more unconstrained Frames corpus. The CNN+BLSTM network improved performance on three data sets and is the largest of the proposed models.

6 Conclusion

We present multi-task BLSTM and CNN models that use slot-filling and user-intent classification as auxiliary tasks for the primary task of system dialogue act selection as part of dialogue policy initialization. The models bootstrap dialogue policy optimization without the need for hand-written rules, as done, e.g., in (Li et al., 2017). We also empirically evaluate multiple RNN and CNN architectures on multiple data sets against two baselines architectures. Our MTL models improve over the performance achieved on single task baseline models (Baseline B) as well as the jointly trained BLSTM model released by Hakkani-Tur et al. (2016).

A dialogue manager that is initialized from corpus data is not flexible enough for new user interactions, therefore additional training is necessary. Future work will include deploying our MTL models as part of a complete dialogue system and continued training with RL. This will allow us to explore the performance of MTL models experimentally on end-to-end systems. Additionally, future work will incorporate additional dialogue context into system dialogue act selection, and model the scenario where more than one system dialogue act may be valid at a given point in the dialogue.

References

Sercan O. Arik, Mike Chrzanowski, Adam Coates, Gregory Diamos, Andrew Gibiansky, Yongguo Kang, Xian Li, John Miller, Andrew Ng, Jonathan Raiman, Shubho Sengupta, and Mohammad Shoeybi. 2017. Deep voice: Real-time neural text-to-speech. *ICML 2017*.

Rich Caruana. 1998. Multitask learning. *Autonomous Agents and Multi-Agent Systems*, 27(1):95133.

Yun-Nung (Vivian) Chen, Dilek Hakkani-Tur, Gokhan Tur, Asli Celikyilmaz, and Jianfeng and Gao. 2016. Syntax or semantics? knowledge-guided joint semantic frame parsing. IEEE Workshop on Spoken Language Technology (SLT 2016).

François Chollet et al. 2015. Keras. GitHub.

Daxiang Dong, Hua Wu, Wei He, Dianhai Yu, and Haifeng Wang. Multi-task learning for multiple language translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, pages 1723–1732.

Layla El Asri, Hannes Schulz, Shikhar Sharma, Jeremie Zumer, Justin Harris, Emery Fine, Rahul Mehrotra, and Kaheer Suleman. 2017. Frames: a corpus for adding memory to goal-oriented dialogue systems. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 207–219, Saarbrücken, Germany. Association for Computational Linguistics.

Milica Gasic and Steve Young. 2014. Gaussian processes for pomdp-based dialogue manager optimisation. *IEEE Transactions on Audio, Speech, and Language Processing*.

Kallirroi Georgila, Oliver Lemon, and James Henderson. 2005. Annotation of communicator dialogue data for learning dialogue strategies and user simulations. *Proceedings of the 9th Workshop on the Semantics and Pragmatics of Dialogue(SEMDIAL:DIALOR)*, pages 61–68.

Kallirroi Georgila, Oliver Lemon, James Henderson, and Johanna D. Moore. 2009. Automatic annotation of context and speech acts for dialogue corpora. *Natural Language Engineering*, 15(3):315–353.

Barbara J. Grosz and Candace L. Sidner. 1986. Attention, intentions, and the structure of discourse. *Computational Linguistics*, 12:175–204.

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

James Henderson, Oliver Lemon, and Kallirroi Georgila. 2007. Hybrid reinforcement/supervised learning of dialogue policies from fixed data sets. *ACL.*

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Vigas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Googles multilingual neural machine translation system: Enabling zero-shot translation. arXiv:1611.0455.

Oliver Lemon and Oivier Pietquin. 2007. Machine learning for spoken dialogue systems. *INTERSPEECH*.

Xiujun Li, Yun-Nung Chen, Lihong Li, Jianfeng Gao, and Asli Celikyilmaz. 2017. End-to-end task-completion neural dialogue systems. In *Proceedings of the The 8th International Joint Conference on Natural Language Processing*, pages 733–743.

Aishwarya Padmakumar, Jesse Thomason, and Raymond J. Mooney. 2017. Integrated learning of dialog strategies and semantic parsing. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

P.J. Price. 1990. Evaluation of spoken language systems: The atis domain. In *Proceedings of the DARPA Workshop on Speech and Natural Language*.

Sebastian Ruder. 2017. Multi-task learning objectives for natural language processing.

Pei-Hao Su, Milica Gasic, Nikola Mrksic, Lina Rojas-Barahona, Stefan Ultes, David Vandyke, Tsung-Hsien Wen, and Steve Young. 2016. Continuously learning neural dialogue management. arXiv:1606.02689.

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2014. Going deeper with convolutions. *CoRR*, pages 1746–1751. Abs/1409.4842.

Shubham Toshniwal, Hao Tang, Liang Lu, and Karen Livescu. 2017. Multitask learning with low-level auxiliary tasks for encoder-decoder based speech recognition. arXiv:1704.01631.

Tsung-Hsien Wen, David Vandyke, Nikola Mrksic, Milica Gasic, Lina M. Rojas-Barahona, Pei-Hao Su, Stefan Ultes, and Steve Young. 2016. A network-based end-to-end trainable task-oriented dialogue system. In *Proceedings of the 15th Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

Jason D. Williams and Geoffrey Zweig. 2016. End-to-end lstm-based dialog control optimized with supervised and reinforcement learning. arXiv preprint arXiv:1606.01269.

Alexander Yeh. 2000. More accurate tests for the statistical significance of result differences. In *Proceedings of the 18th conference on Computational linguistics, COLING '00*.

Kim Yoon. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751.

S. Young, M. Gasic, B. Thomson, and J.D. Williams. 2013. Pomdp-based statistical spoken dialog systems: A review. proceedings of the ieee. 101(5):1160–1179.

Ye Zhang and Byron C. Wallace. 2015. A sensitivity analysis of (and practitioners guide to) convolutional neural networks for sentence classification. arXiv:1510.03820.

Yu Zhang and Qiang Yang. 2017. A survey on multi-task learning. arXiv:1707.08114.

Tiancheng Zhao and Maxine Eskenazi. 2016. Towards end-to-end learning for dialog state tracking and management using deep reinforcement learning. *the Annual SIGdial Meeting on Discourse andDialogue (SIGDIAL)*.

Barret Zoph and Kevin Knight. 2016. Multi-source neural translation. *NAACL*, pages 30–34.