Exploring Named Entity Recognition As an Auxiliary Task for Slot Filling in Conversational Language Understanding

Samuel Louvan
University of Trento
Fondazione Bruno Kessler
slouvan@fbk.eu

Bernardo Magnini
Fondazione Bruno Kessler
magnini@fbk.eu

Abstract

Slot filling is a crucial task in the Natural Language Understanding (NLU) component of a dialogue system. Most approaches for this task rely solely on the domain-specific datasets for training. We propose a joint model of slot filling and Named Entity Recognition (NER) in a multi-task learning (MTL) setup. Our experiments on three slot filling datasets show that using NER as an auxiliary task improves slot filling performance and achieve competitive performance compared with state-of-the-art. In particular, NER is effective when supervised at the lower layer of the model. For low-resource scenarios, we found that MTL is effective for one dataset.

1 Introduction

Most of the current dialogue systems depend on an NLU component to extract semantic information from an utterance. Such semantic information is often represented as a semantic frame which contains the domain, intent of the user, and predefined attributes (*slots*). Each word of the utterance is labeled with a slot, which defines a particular attribute (an entity, time, etc) of the utterance. Table 1 shows an example of a semantic frame for the sentence *"Show me the prices of all flights from Atlanta to Washington DC*" with Begin/In/Out (BIO) representation.

Domain	airline
Intent	search airfare
Utterance	**Slot Label**
show	O
me	O
the	O
prices	O
of	O
all	O
flights	O
from	O
Atlanta	B-fromloc.city_name
to	O
Washington	B-toloc.city_name
DC	I-toloc.city_name

Table 1: An example of a semantic frame with its coressponding domain, intent and slots.

We focus on *slot filling*, a task of automatically extracting slots for a given utterance. This task can be treated as a sequence labeling problem and the most successful approach is to employ a conditional random fields (CRF) on top of a deep recurrent neural networks (RNN). In general, there are two ways of training a slot filling model: (i) train a domain-specific model (Goo et al., 2018; Wang et al., 2018) or (ii) train a model that performs well across domains using domain adaptation or transfer learning techniques (Hakkani-Tür et al., 2016; Jaech et al., 2016; Jha et al., 2018; Kim et al., 2017). One popular transfer learning technique is multi-task learning (MTL) (Caruana, 1997) in which a joint model is trained on a target (main) task and several auxiliary tasks simultaneously to learn better feature representations across tasks. This technique has shown potential on various NLP tasks and offer flexibility as it allows transfer learning across different domains and tasks (Yang et al., 2017). On slot filling, Jaech et al. (2016) train a single slot filling model on different domains and show that MTL is particulary useful in low resource scenarios.

Identifying beneficial auxiliary task for the target task is important when applying MTL (Bingel and Søgaard, 2017). In this work, we investigate the effectiveness of Named Entity Recognition (NER) as an auxiliary task for slot filling. We propose NER because of two main reasons. First, the slot values are typically named entities, for example airline name, city name, etc. Second, the state of the art performance of models for NER have been relatively high (Lample et al., 2016; Ma and Hovy, 2016). Therefore, we expect that the

Proceedings of the 2018 EMNLP Workshop SCAI: The 2nd Int'l Workshop on Search-Oriented Conversational AI, pages 74–80
Brussels, Belgium, October 31, 2018.
ISBN 978-1-948087-75-9

learned features of NER can improve the slot filling performance. Finally, NER corpus is relatively easier to obtain compared to domain specific slot filling datasets.

We are interested to answer the following questions:

- *Does NER help the performance of slot filling in the MTL setup?* As NER labels are usually more coarse-grained than slot filling labels, predicted NER label might provide good signal to the more fine-grained slot labels. For example, the location `LOC` label in NER can be a strong indicator for slots `fromloc.city_name` or `toloc.city_name` and filter out other slot labels which are not related to location. We hope the model can learn more general knowledge first and transfer such knowledge to predict more specific slot information using MTL.

- *What is the effect of supervising NER on the lower layer of the MTL model to the slot filling performance?* Inspired by recent work of Søgaard and Goldberg (2016), we investigate the effect of supervising NER on different layers of the model. Our hypothesis is that a more "general" feature is better learned on the lower layer in order to support a task which depends on a more "specific" feature.

In addition, we also experiment on cross-domain slot filling models by jointly training slot filling datasets from *similar* domains using a MTL setup. We explore two techniques to measure similarity between domains: domain similarity by Ruder and Plank (2017a) and label embedding mapping by Kim et al. (2015).

We experiment with three datasets from different domains. Our experiments show that for all datasets, using NER as an auxiliary task is beneficial for the slot filling performance. NER is consistently helpful when it is supervised at the lower layer. On the low resource scenario, we found mixed results, in which MTL is only effective for 1 dataset.

2 Model

This section describes the slot filling model, the multi-task learning setup, and the data selection that we use in our experiments.

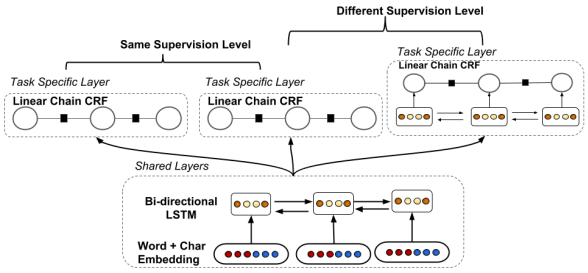

Figure 1: Multi-task Learning with different supervision level

2.1 Slot Filling Model

For the slot filling model, we adopt a neural based model similar to (Lample et al., 2016; Ma and Hovy, 2016), as it achieves the state of the art performance in sequence labeling task (NER). Recent slot filling model of Jha et al. (2018) also used a variant of this model. Given an input sentence, we represent each word w_i using a concatenation of its word embedding $\mathbf{e}(w_i)$ and character-level embeddings $\mathbf{c}(w_i)$: $\mathbf{x}_i = [\mathbf{e}(w_i); \mathbf{c}(w_i)]$. The character-level embeddings are computed using convolutional neural networks (CNN), similar to the one proposed by Kim et al. (2016). We then feed $\mathbf{x}_i$ to a bidirectional LSTM (biLSTM) word-level encoder to incorporate the contextual information of w_i. The output of the backward and forward LSTM at each time step is then concatenated and fed into a CRF layer. The CRF layer computes the final output, e.g. the tag of each input. We use one hidden layer between biLSTM and CRF as it has been shown by Lample et al. (2016) that it can improve performance.

2.2 Multi-Task Learning

One simple technique to perform MTL is by training the target and auxiliary tasks simultaneously. In this setting, the parameters of the model are shared across tasks, pushing the model to learn feature representations that work well across tasks.

Figure 1 depicts the MTL setting that we use in our work. The lower parts of the network, i.e. word embeddings, character-level embeddings, and bi-LSTM encoder are shared among tasks. After the bi-LSTM layer, we use different CRF layers for each task to predict the task-specific tags (NER or slot filling). We also experiment with MTL setup which uses different level of supervision for the auxiliary task (Søgaard and Goldberg, 2016), in which we use two layers of biLSTM encoder and only share the lower layer of

Dataset	#sent			#token	#label	Label Examples
	train	dev	test			
Slot Filling						
ATIS	4478	500	893	869	79	airport name, airline name, return date
MIT Restaurant	6128	1532	3385	4166	8	restaurant name, dish, price, hours
MIT Movie	7820	1955	2443	5953	12	actor, director, genre, title, character
NER						
CoNLL 2003	14987	3466	3684	21010	4	person, location, organization
OntoNotes 5.0	34970	5896	2327	34662	18	organization, gpe, date, money, quantity

Table 2: Statistics of the datasets. For each dataset, number of sentence in train/dev/test set, the number of unique token and label in the training set.

the encoder and keep the outer layer for the main slot filling task.

2.3 Data Selection

Ruder and Plank (2017b) demonstrate that selecting data for training the auxiliary task might improve the target task performance. We investigate two data selection techniques for our MTL experiments:

Domain Similarity. We use Jensen-Shannon divergence (JSD; Lin, 1991) to measure domain similarity as proposed by Ruder and Plank (2017b): $\frac{1}{2}(D_{KL}(P||M) + D_{KL}(K||M))$ where $M = \frac{1}{2}(P + Q)$. $D_{KL}(P||Q)$ is the Kullback-Leibler divergence between two distributions P and Q. We use term distributions (Plank and Van Noord, 2011) of each domain to compute P and Q. We select the most similar domain to the main task domain to be used as the *auxiliary task*.

Label Embedding Mapping. In an MTL setup, sometimes we only want to keep auxiliary labels which are semantically similar to target task labels and remove other irrelevant labels of the auxiliary task. For example, the slot filling label `airport.statename` is similar to `LOC` but not to `TIME` auxiliary NER label. We employ label embedding mapping approach by Kim et al. (2015) using Canonical Correlation Analysis (CCA). The idea is to construct matrix representation where rows are labels and columns are words in the vocabulary. The cell value in the matrix is the pointwise mutual information (PMI) between the label and the word. After that, we perform rank-k SVD on the matrix and normalized the rows of the matrix. Each row with k dimension of the matrix is the label embedding of a particular label. We use the cosine similarity between two label embedding representations to obtain the nearest neighbor.

Target Task	Most Similar Domain
ATIS	MIT-R
MIT-R	MIT-M
MIT-M	MIT-R

Table 3: Most similar domain for each target task computed with JSD

3 Experimental Setup

Data. We use three slot filling datasets (Table 2): Airline Travel Information System (ATIS; Tür et al., 2010), MIT Restaurant (MIT-R) and MIT Movie (MIT-M) (Liu et al., 2013; Liu and Lane, 2017b). The ATIS dataset is widely used in conversational language understanding and contains queries to a flight database. We use the provided slot annotations and use the same split as in Hakkani-Tür et al. (2016). The MIT-R contains utterances related to restaurant search and MIT-M contains queries related to movie information. For both datasets, we use the default split.[1] As for the NER dataset, we use two datasets : CoNLL 2003 (Tjong Kim Sang and De Meulder, 2003) and Ontonotes 5.0 (Pradhan et al., 2013). For OntoNotes, we use the Newswire section for our experiments.

Implementation. We use the existing BiLSTM-CRF sequence tagger implementation from Reimers and Gurevych (2017) for all experiments.[2] We use the pre-trained word embedding from (Komninos and Manandhar, 2016). We set the LSTM hidden units to 100. The word and character embeddings dimensions are set to 300 and 30 respectively. We use dropout rate of 0.25. We train the model using the Adam optimizer (Kingma and Ba, 2014) for 25 epochs with early stopping on the target task. For each epoch, we

[1] https://groups.csail.mit.edu/sls/downloads/

[2] https://github.com/UKPLab/emnlp2017-bilstm-cnn-crf

Model	Aux. Task		Target Task		
	SF	NER	ATIS	MIT-R	MIT-M
Bi-model based (Wang et al., 2018)	-	-	**96.89**	-	-
Slot gated model (Goo et al., 2018)	-	-	95.20	-	-
Recurrent Attention (Liu and Lane, 2016)	-	-	95.87	-	-
Adversarial(Liu and Lane, 2017a)	-	-	95.63	74.47	85.33
Single task (STL)	-	-	95.68	78.58	<u>87.34</u>
MTL, same supervision level	most similar	-	95.47	78.56	86.89
MTL, same supervision level	all	-	95.68	78.70	87.22
MTL, same supervision level	most similar	✓	95.50	78.41	86.77
MTL, same supervision level	all	✓	95.34	78.27	86.76
MTL, same supervision level	-	✓	95.71	78.40	87.09
MTL, different supervision level	most similar	✓	95.70	<u>79.10</u>	86.94
MTL, different supervision level	all	✓	<u>95.94</u>	79.00	86.92
MTL, different supervision level	-	✓	95.40	**79.13**	**87.41**

Table 4: F1 scores comparison between MTL, STL, and previous published results on each dataset. "Most Similar" auxiliary task means we take the most similar slot filling domain (excluding NER) as the auxiliary task. "All" includes all the slot filling domains as the auxiliary tasks (excluding NER). For the "different supervision level", NER is supervised at the lower layer and slot filling tasks at the higher layer. Bold: best, Underline: second best.

train the model of each task in alternate fashion. We evaluate the performance by computing the F1-score on the test set using the standard CoNLL-2000 evaluation[3]

Target Task & Auxiliary Tasks. For each MTL experiment, there is exactly one target task and one or more auxiliary task(s). The target task is always a slot filling task, i.e. either ATIS, MIT-R, or MIT-M. The auxiliary task(s) consist of a combination of slot filling tasks from different domains of the target task with (or without) a NER task. We select the most similar slot filling task for the target task using the domain similarity technique described in (§2.3). Table 3 presents the most similar slot filling domain for each slot filling task.

4 Results and Analysis

Overall Performance. Table 4 summarizes the slot filling performance of our single task (STL) versus MTL models. The performance from previous studies are directly copied from their reported numbers. When using the same supervision level for both target and auxiliary tasks, using the most similar domain performs worse than using all domains. In contrast, using NER together with the most similar domain as auxiliary tasks performs better than using all the domains.

Experiments on different supervision level show that using NER as an auxiliary task consistently improves slot filling performance. This result matches our intuition that the task with more coarse-label, such as NER, is better to be supervised at the lower layer of the model. On ATIS and MIT-R datasets, MTL achieves better performance compared to STL. However, on MIT-M, STL outperforms some MTL models.

In order to understand better the behavior of the models, we analyze the results from the development set. For the ATIS dataset, STL and MTL have the same performance in 44 out of 67 slots in the development set. For the rest of the slots, STL performs better mostly on slots related to time such as `arrive_time.time` and `depart_date.month_name` while MTL is better on recognizing location related slots such as `city_name` and `toloc.state_name`. For the MIT Restaurant dataset, MTL performs better on 5 out of 8 slots. MTL performs well in identifying slots related to time and location in the MIT Restaurant dataset. For the MIT movie, MTL yields better results for time related slots. As for the person related slots such as `character` , `actor`, and `director`, STL gives better results. Overall, although incorporating NER with slot filling shows improvements, the difference is still rather small especially for the ATIS and the MIT Movie datasets. Further work is needed to explore better mechanism to inject NER information to help slot filling in the MTL setup. It is also interesting to compare the performance of MTL and pipeline based system which utilizes NER prediction as one of the feature for the slot filling model.

[3]https://www.clips.uantwerpen.be/conll2000/chunking/output.htm

Model	ATIS	MIT-R	MIT-M
MTL	**95.94**	**79.10**	**87.34**
MTL+Label Emb.	95.66	78.37	86.84

Table 5: The effect of the label filtering on MTL performance

Dataset	# training sents	STL	MTL
ATIS	200	**83.88**	81.27
	400	**85.54**	85.21
	800	90.48	**90.68**
MIT-R	200	54.65	**54.91**
	400	61.36	**61.88**
	800	67.48	**68.27**
MIT-M	200	68.28	**69.12**
	400	74.09	**75.15**††
	800	**79.33**	79.08

Table 6: Performance comparison between STL and MTL for low resource scenarios. †† indicates significant improvement over STL baseline with $p < 0.05$ using approximate randomization testing.

Effect of Label Embedding Mapping. We apply label filtering on the auxiliary tasks using the label embedding mapping (§2.3). On the auxiliary dataset(s), we keep the most similar labels and replace irrelevant labels with O. The MTL setup that we use is the best performing MTL for each dataset in Table 4. As shown in Table 5, the performance of MTL drops when we apply filtering to the auxiliary labels. We suspect that this is due to the quality of the label mapping and also a high number of "O" label after the filtering process.

Low Resource Scenarios. We experiment on low resource scenarios where we vary the number of training sentences to 200, 400, and 800 sentences for each dataset. The MTL setup that we use is the best performing MTL for each dataset in Table 4. As shown in Table 6, MTL consistently performs better than STL for the MIT-R dataset. While for the ATIS and MIT-M datasets, STL mostly gives better results than MTL.

5 Related Work

Recent studies on slot filling in conversational systems are mostly based on neural models. Wang et al. (2018) introduce a bi-model (RNN) structure to consider cross-impact between intent detection and slot filling. Liu and Lane (2016) propose an attention mechanism on the encoder-decoder model for joint intent classification and slot filling. (Goo et al., 2018) extend the attention mechanism using a slot gated model to learn relationship between slot and intent attention vectors. Hakkani-Tür et al. (2016) use bidirectional RNN as a single model that handle multiple domains by adding a final state that contains domain identifier. The work by Jha et al. (2018); Kim et al. (2017) uses expert based domain adaptation while Jaech et al. (2016) propose a multi-task learning approach to guide the training of a model for new domain. All of these studies train their model solely on slot filling datasets, while our focus is to exploit a more "general" resource, such as NER, by training the model jointly with slot filling through MTL with different supervision level.

6 Conclusion

In this work, we investigate the effectiveness of training a slot filling model jointly with NER as an auxiliary task through MTL setup. Our experiments demonstrate that NER is helpful for slot filling. In particular, NER is more effective when it is supervised at the lower layer of the MTL model. However, further work is needed to investigate the effectiveness of domain similarity metric or label embedding mapping as a way to perform data selection in the preprocessing step.

Acknowledgments

The authors would like to thank anonymous reviewers and Clara Vania for the helpful comments and feedback. This work was supported by the grant of Fondazione Bruno Kessler PhD scholarship.

References

Joachim Bingel and Anders Søgaard. 2017. Identifying beneficial task relations for multi-task learning in deep neural networks. *EACL 2017*, page 164.

Rich Caruana. 1997. Multitask learning. *Machine Learning*, 28:41–75.

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, NAACL-HLT, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 2 (Short Papers)*, pages 753–757.

Dilek Z. Hakkani-Tür, Gökhan Tür, Asli elikyilmaz, Yun-Nung Chen, Jianfeng Gao, Li Deng, and Ye-Yi Wang. 2016. Multi-domain joint semantic frame

parsing using bi-directional rnn-lstm. In *INTERSPEECH*.

Aaron Jaech, Larry P. Heck, and Mari Ostendorf. 2016. Domain adaptation of recurrent neural networks for natural language understanding. In *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, Yacine Jernite, David Sontag, and Alexander Rush. 2016. Character-aware neural language models. In *Proceedings of the 2016 Conference on Artificial Intelligence (AAAI)*.

Young-Bum Kim, Karl Stratos, and Dongchan Kim. 2017. Domain attention with an ensemble of experts. In *ACL*.

Young-Bum Kim, Karl Stratos, Ruhi Sarikaya, and Minwoo Jeong. 2015. New transfer learning techniques for disparate label sets. 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 473–482.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Alexandros Komninos and Suresh Manandhar. 2016. Dependency based embeddings for sentence classification tasks. In *HLT-NAACL*.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *Proceedings of NAACL-HLT*, pages 260–270.

Jianhua Lin. 1991. Divergence measures based on the shannon entropy. *IEEE Transactions on Information theory*, 37(1):145–151.

Bing Liu and Ian Lane. 2016. Attention-based recurrent neural network models for joint intent detection and slot filling. In *Interspeech 2016*.

Bing Liu and Ian Lane. 2017a. Multi-Domain Adversarial Learning for Slot Filling in Spoken Language Understanding.

Bing Liu and Ian Lane. 2017b. Multi-domain adversarial learning for slot filling in spoken language understanding. In *NIPS Workshop on Conversational AI*.

Jingjing Liu, Panupong Pasupat, Yining Wang, Scott Cyphers, and James R. Glass. 2013. Query understanding enhanced by hierarchical parsing structures. In *2013 IEEE Workshop on Automatic Speech Recognition and Understanding, Olomouc, Czech Republic, December 8-12, 2013*, pages 72–77.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional lstm-cnns-crf. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1064–1074.

Barbara Plank and Gertjan Van Noord. 2011. Effective measures of domain similarity for parsing. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 1566–1576. Association for Computational Linguistics.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Hwee Tou Ng, Anders Björkelund, Olga Uryupina, Yuchen Zhang, and Zhi Zhong. 2013. Towards robust linguistic analysis using ontonotes. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 143–152.

Nils Reimers and Iryna Gurevych. 2017. Reporting Score Distributions Makes a Difference: Performance Study of LSTM-networks for Sequence Tagging. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 338–348, Copenhagen, Denmark.

Sebastian Ruder and Barbara Plank. 2017a. Learning to select data for transfer learning with bayesian optimization. In *EMNLP*.

Sebastian Ruder and Barbara Plank. 2017b. Learning to select data for transfer learning with bayesian optimization. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 372–382, Copenhagen, Denmark. Association for Computational Linguistics.

Anders Søgaard and Yoav Goldberg. 2016. Deep multi-task learning with low level tasks supervised at lower layers. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 231–235.

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-Volume 4*, pages 142–147. Association for Computational Linguistics.

Gökhan Tür, Dilek Z. Hakkani-Tür, and Larry P. Heck. 2010. What is left to be understood in atis? *2010 IEEE Spoken Language Technology Workshop*, pages 19–24.

Yu Wang, Yilin Shen, and Hongxia Jin. 2018. A bi model based rnn semantic frame parsing model for intent detection and slot filling. In *NAACL*.

Zhilin Yang, Ruslan Salakhutdinov, and William W Cohen. 2017. Transfer learning for sequence tagging with hierarchical recurrent networks. *arXiv preprint arXiv:1703.06345*.