From General to Specific: Leveraging Named Entity Recognition 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

English. Slot filling techniques are often adopted in language understanding components for task-oriented dialogue systems. In recent approaches, neural models for slot filling are trained on domain-specific datasets, making it difficult porting to similar domains when few or no training data are available. In this paper we use multi-task learning to leverage general knowledge of a task, namely Named Entity Recognition (NER), to improve slot filling performance on a semantically similar domain-specific task. Our experiments show that, for some datasets, transfer learning from NER can achieve competitive performance compared with the state-of-the-art and can also help slot filling in low resource scenarios.

Italiano. *Molti sistemi di dialogo task-oriented utilizzano tecniche di slot-filling per la comprensione degli enunciati. Gli approcci piú recenti si basano su modelli neurali addestrati su dataset specializzati per un certo dominio, rendendo difficile la portabilitá su dominii simili, quando pochi o nessun dato di addestramento é disponibile. In questo contributo usiamo multi-task learning per sfruttare la conoscenza generale proveniente da un task, precisamente Named Entity Recognition (NER), per migliorare le prestazioni di slot filling su dominii specifici e semanticamente simili. I nostri esperimenti mostrano che transfer learning da NER aiuta lo slot filling in dominii con poche risorse e raggiunge risultati competitivi con lo stato dell'arte.*

1 Introduction

In dialogue systems, semantic information of an utterance is generally represented with a *semantic frame*, a data structure consisting of a domain, an intent, and a number of slots (Tur, 2011). For example, given the utterance *"I'd like a United Airlines flight on Wednesday from San Francisco to Boston"*, the domain would be **flight**, the intent is **booking**, and the slot fillers are **United Airlines** (for the slot `airline_name`), **Wednesday** (`booking_time`), **San Francisco** (`origin`), and **Boston** (`destination`). Automatically extracting this information involves domain identification, intent classification, and slot filling, which is the focus of our work.

Slots are usually domain specific as they are predefined for each domain. For instance, in the flight domain the slots might be `airline_name`, `booking_time`, and `airport_name`, while in the bus domain the slots might be `pickup_time`, `bus_name`, and `travel_duration`. Recent successful approaches related to slot filling tasks (Wang et al., 2018; Liu and Lane, 2017a; Goo et al., 2018) are based on variants of recurrent neural network architecture. In general there are two ways of approaching the task: (i) by training a single model for each domain; or (ii) by performing domain adaptation, which results in a model that learns better feature representations across domains. All these approaches directly train the models on domain-specific slot filling datasets.

In our work, instead of using a domain-specific slot filling dataset, which can be expensive to obtain being task specific, we propose to leverage knowledge gained from a more "general", but semantically related, task, referred as the *auxiliary task*, and then transfer the learned knowledge to the more specific task, namely slot filling, referred as the *target task*, through transfer learning. In the literature, the term transfer learning can be used in different ways. We follow the definition from (Mou et al., 2016), in which transfer learning is viewed as a paradigm which enables a model to use knowledge from auxiliary tasks to help the target task. There are several ways to train this model: we can directly use the trained parameters of the auxiliary tasks to initialize the parameters in the target task (*pre-train* & *fine-tuning*), or train a model of auxiliary and target tasks simultaneously, where some parameters are shared (*multi-task learning*).

We propose to train a slot filling model jointly with Named Entity Recognition (NER) as an auxiliary task through multi-task learning (Caruana, 1997). Recent studies have shown the potential of multi-task learning in NLP models. For example, (Mou et al., 2016) empirically evaluates transfer learning in sentence and question classification tasks. (Yang et al., 2017) proposes an approach for transfer learning in sequence tagging tasks.

NER is chosen as the auxiliary task for several reasons. First, named entities frequently occur as slot values in several domains, which make them a relevant general knowledge to exploit. The same NER type can refer to different slots in the same utterance. On the previous utterance example, the NER labels are LOC for both *San Francisco* and *Boston*, and ORG for *United Airlines*. Second, state-of-the-art performance of NER (Lample et al., 2016; Ma and Hovy, 2016) is relatively high, therefore we expect that the transferred feature representation can be useful for slot filling tasks. Third, large annotated NER corpora are easier to obtain compared to domain-specific slot filling datasets.

The contributions of this work are as follows: we investigate the effectiveness of leveraging Named Entity Recognition as an auxiliary task to learn general knowledge, and transfer this knowledge to slot filling as the target task in a multi-task learning setting. To our knowledge, there is no reported work that uses NER transfer learning for slot filling in conversational language understanding. Our experiments show that for some datasets multi-task learning achieves better overall performance compared to previous published results, and performs better in some low resource scenarios.

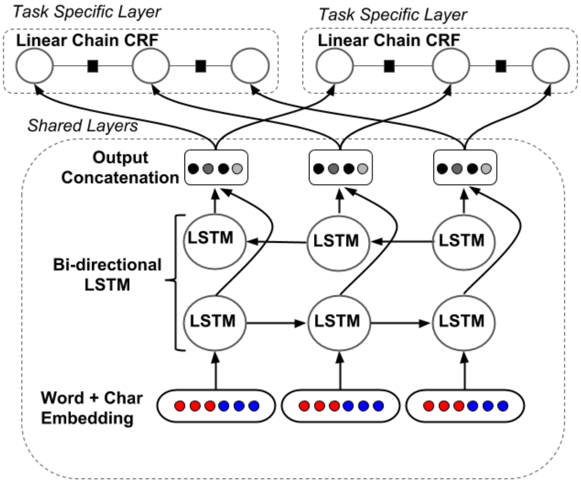

Figure 1: Multi-task Learning Network architecture.

2 Related Work

Recent approaches on slot filling for conversational agents are based mostly on neural models. The work by (Wang et al., 2018) introduces a bi-model Recurrent Neural Network (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) extends the attention mechanism using a slot gated model to learn relationships between slot and intent attention vectors. The work from (Hakkani-Tür et al., 2016) uses bidirectional RNN as a single model that handles multiple domains by adding a final state that contains domain identifier. (Jha et al., 2018; Kim et al., 2017) uses expert based domain adaptation while (Jaech et al., 2016) proposes a multi-task learning approach to guide the training of a model for new domains. All of these studies train their model solely on slot filling datasets, while our focus is to leverage more "general" resources, such as NER, by training the model simultaneously with slot filling through multi-task learning.

3 Model

In this Section we describe the base model that we use for the slot filling task and the transfer learning model between NER and slot filling.

3.1 Base Model

The model that we use is a hierarchical neural based model, as it has shown to be the state of the art in sequence tagging tasks such as named entity recognition (Ma and Hovy, 2016; Lample

Sentence	find	flights	from	Atlanta	to	Boston
Slot	O	O	O	`B-fromloc`	O	`B-toloc`

Table 1: An example output from the model.

et al., 2016). Figure 1 depicts the overall architecture of the model. The model consists of several stacked bidirectional RNNs and a CRF layer on top to compute the final output. The input of the model are both words and characters in the sentence. Each word is represented with a word embedding, which is simply a lookup table. Each word embedding is concatenated with its character representation. The character representation itself can be composed from a concatenation of the final state of bidirectional LSTM (Hochreiter and Schmidhuber, 1997) over characters in a word or extracted using a Convolutional Neural Network (CNN) (LeCun et al., 1998). The concatenation of word and character embeddings is then passed to a LSTM cell. The output of the LSTM in each time step is then fed to a CRF layer. Finally, the output of the CRF layer is the slot tag for a word in the sentence, as shown in Table 1.

3.2 Transfer Learning Model

In the context of NLP, recent studies have applied transfer learning in tasks such as POS tagging, NER, and semantic sequence tagging (Yang et al., 2017; Alonso and Plank, 2017). In general, a popular mechanism is to do multitask learning with a network that optimizes the feature representation for two or more tasks simultaneously. In particular, among the tasks we can set target tasks and auxiliary tasks. In our case, the target task is the slot filling task and the auxiliary task is the NER task. Both tasks are using the base model explained in the previous section with a task specific CRF layer on top.

4 Experimental Setup

The objective of our experiment is to validate the hypothesis that by training a slot filling model with semantically related tasks, such as NER, can be helpful to the slot filling performance. We compare the performance of Single Task Learning (STL) and Multi-Task Learning (MTL). STL uses the Bi-LSTM + CRF model described in (§3.1) and it is trained directly on the target slot filling task. MTL refers to (§3.2), in which models for slot filling and NER are trained simultaenously and some parameters are shared.

Dataset	#sents	#tokens	#label	Label Examples
Slot Filling				
ATIS	4478	869	79	airport name, airline name, return date
MIT Restaurant	6128	3385	20	restaurant name, dish, price, hours
MIT Movie	7820	5953	8	actor, director, genre, title, character
NER				
CoNLL 2003	14987	23624	4	person, location, organization
OntoNotes 5.0	34970	39490	18	organization, gpe, date, money, quantity

Table 2: Training data statistics.

Data. We use three conversational slot filling datasets to evaluate the performance of our approach: the ATIS dataset on Airline Travel Information Systems (Tür et al., 2010), the MIT Restaurant and the MIT Movie datasets[1] (Liu et al., 2013; Liu and Lane, 2017a) on restaurant reservations and movie information respectively. Each dataset provides a number of conversational user utterances, where tokens in the utterance are annotated with their domain specific slot. 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. Table 2 shows the statistics and example labels of each dataset. We use the training-test split provided by the developers of the datasets, and have further split the training data into 80% training and 20% development sets.

Implementation. We use the multi-task learning implementation from (Reimers and Gurevych, 2017) and have adapted it for our experiments. We consider slot filling as the target task and NER as the auxiliary task. We use a pretrained embedding

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

Model	ATIS	MIT Restaurant	MIT Movie
Bi-model based (Wang et al., 2018)	**96.89**	-	-
Slot gated model (Goo et al., 2018)	95.20	-	-
Recurrent Attention (Liu and Lane, 2016)	95.78	-	-
Adversarial (Liu and Lane, 2017b)	95.63	74.47	85.33
Base model (STL)	95.68	78.58	**87.34**
MTL with CoNLL 2003	95.43	78.82	87.31
MTL with OntoNotes	95.78	**79.81**††	87.20
MTL with CoNLL 2003 + OntoNotes	95.69	78.52	86.93

Table 3: F1 score comparison of MTL, STL and the state of the art approaches. †† indicates significant improvement over STL baseline with $p < 0.05$ using approximate randomization testing.

Slot	ATIS		MIT Restaurant		MIT Movie	
	STL	MTL	STL	MTL	STL	MTL
PER	-	-	-	-	**90.73**	89.58
LOC	98.91	**99.32**	81.95	**83.47**††	-	-
ORG	100.00	100.00	-	-	-	-

Table 4: Performance on slots related to CoNLL tags on the development set (MTL with CONLL).

Dataset	#training sents	STL	MTL-C	MTL-O
ATIS	200	84.37	83.15	**84.97**
	400	**87.04**	86.54	86.93
	800	90.67	91.15	**91.58**††
MIT Restaurant	200	54.65	**56.95**††	56.79
	400	62.91	**63.91**	62.29
	800	68.15	**68.52**	68.47
MIT Movie	200	69.97	**71.11**††	69.78
	400	**75.88**	75.23	75.18
	800	79.33	**80.28**††	78.65

Table 5: Performance comparison on low resource scenarios. MTL-C and MTL-O are MTL models trained on CoNLL and OntoNotes datasets respectively. †† indicates significant improvement over STL with $p < 0.05$ using approximate randomization testing.

from (Komninos and Manandhar, 2016) to initialize the word embedding layer. We did not tune the hyperparameters extensively, although we followed the suggestions in a comprehensive study of hyperparameters in sequence labeling tasks from (Reimers and Gurevych, 2017). The word and character embedding dimensions, and dropout rate are set to 300, 30, and 0.25 respectively. The LSTM size is set to 100 following (Lample et al., 2016). We use CNN to generate the character embedding as in (Ma and Hovy, 2016). For each epoch in the training, we train both the target task and the auxiliary task and keep the data size between them proportional. We train the network using Adam (Kingma and Ba, 2014) optimizer. Each model is trained for 50 epochs with early stopping on the target task. We evaluate the performance of the target task by computing the F1-score of the test data following the standard CoNLL-2000 evaluation[2].

5 Results and Analysis

Overall performance. Table 3 shows the comparison of our Single Task Learning (STL) and Multi-Task Learning (MTL) models with the current state of the art performance for each dataset. For the ATIS dataset, the performance of the STL model is comparable to most of the state-of-the-art approaches, however not all MTL models lead to an increase in the performance. As for the MIT Restaurant, both STL and MTL models achieve better performance compared to the previously published results (Liu and Lane, 2017a). For the MIT movie dataset, STL achieves better results by a small margin over MTL. Both STL and MTL performs better than the previous approach for the MIT movie dataset. When we combine CoNLL and OntoNotes into three tasks in the MTL setting, the overall performance tends to decrease across datasets compared to MTL with OntoNotes only.

Per slot performance. Although the overall performance using MTL is not necessarily helpful, we analyze the per slot performance in the development set to get better understanding of the model's behaviour. In particular, we want to know whether slots that are related to CoNLL tags perform better through MTL compared to STL, as evidence of transferable knowledge. To this goal, we manually created a mapping between NER CoNLL tags and slot tags for each dataset. For example in the ATIS dataset, some of the slots that are related to the LOC tags are `fromloc.airport_name` and `fromloc.city_name`. We compute the micro-F1 scores for the slots based on this mapping. Table 4 shows the performance of the slots related to CoNLL tags on the development set. For the ATIS and MIT Restaurant datasets we can see that MTL improves the performance in recognizing `LOC` related tags. While for the MIT Movie dataset, MTL suffers from performance decrease on `PER` tag. There are three slots related to `PER` in MIT Movie namely `CHARACTER`, `ACTOR`, and `DIRECTOR`. We found that the decrease is on `DIRECTOR` while for `ACTOR` and `CHARACTER` there is actually an improvement. We sample 10 sentences in which the model makes mistakes on `DIRECTOR` tag. Of these sentences, four sentences are wrongly annotated. Another four sentences are errors by the model although the sentence seems easy, typically the model is confused between `DIRECTOR` and `ACTOR`. The rests are difficult sentences. For example, the sentence: "*Can you name Akira Kurusawas first color film*". This sentence is somewhat general and the model needs more information to discriminate between `ACTOR` and `DIRECTOR`.

[2] https://www.clips.uantwerpen.be/conll2000/chunking/output.html

Low resource scenario. In Table 5 we compare STL and MTL under varying numbers of training sentences to simulate low resource scenarios. We did not perform MTL including *both* CoNLL and OntoNotes, as the results from Table 3 show that performance tends to degrade when we include both resources. For the MIT Restaurant, for all the low resource scenarios, MTL consistently gives better results. In the MIT Restaurant dataset, it is evident that the less number of training sentences that we have, the more helpful is MTL. For the ATIS and MIT Movie, MTL performs better than STL except for the 400 sentence training scenario. We suspect that to have a more consistent MTL improvement in different low resource scenarios, a different training strategy is needed. In our current experiments, the number of training data is proportional between the target task and auxiliary task. In the future, we would like to try other training strategies, such as using the full training data from the auxiliary task. As the data from the target task is much smaller, we plan to repeat the batch of the target task until we finish training all the batches from the auxiliary task in an epoch. This strategy is similar to (Jaech et al., 2016).

Regarding the variation of results that we get from CoNLL or OntoNotes, we believe that selecting promising auxiliary tasks, or selecting data from a particular auxiliary task, are important to alleviate *negative transfer*. This also has been shown empirically in (Ruder and Plank, 2017; Bingel and Søgaard, 2017). Another alternative to reduce negative transfer, which would be interesting to try in the future, is by using a model which can decide which knowledge to share (or not to share) among tasks (Ruder et al., 2017; Meyerson and Miikkulainen, 2017).

6 Conclusion

In this work we train a slot filling domain-specific model adding NER information, under the assumption that NER introduces useful "general" labels, and that it is cheaper to obtain compared to task specific slot filling datasets. We use multi-task learning to leverage the learned knowledge from NER to slot filling task. Our experiments show evidence that we can achieve comparable or better performance against the state-of-the-art approaches and against single task learning, both in full training data and low resource scenarios. In the future, we are interested in working on datasets in Italian and explore more sophisticated multi-task learning strategies.

Acknowledgments

We would like to thanks three anonymous reviewers and Simone Magnolini, Marco Guerini, Serra Sinem Tekiroğlu for helpful comments and feedback. This work was supported by the grant of Fondazione Bruno Kessler PhD scholarship.

References

Héctor Martínez Alonso and Barbara Plank. 2017. When is multitask learning effective? semantic sequence prediction under varying data conditions. In *15th Conference of the European Chapter of the Association for Computational Linguistics*.

Joachim Bingel and Anders Søgaard. 2017. Identifying beneficial task relations for multi-task learning in deep neural networks. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 164–169. Association for Computational Linguistics.

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 Çelikyilmaz, 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*.

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

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.

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

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.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

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. In *NIPS Workshop on Conversational AI*.

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

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.

Elliot Meyerson and Risto Miikkulainen. 2017. Beyond shared hierarchies: Deep multitask learning through soft layer ordering. *arXiv preprint arXiv:1711.00108*.

Lili Mou, Zhao Meng, Rui Yan, Ge Li, Yan Xu, Lu Zhang, and Zhi Jin. 2016. How transferable are neural networks in nlp applications? *arXiv preprint arXiv:1603.06111*.

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, 09.

Sebastian Ruder and Barbara Plank. 2017. 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, September. Association for Computational Linguistics.

Sebastian Ruder, Joachim Bingel, Isabelle Augenstein, and Anders Søgaard. 2017. Learning what to share between loosely related tasks. *arXiv preprint arXiv:1705.08142*.

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.

Gokhan Tur. 2011. *Spoken Language Understanding: Systems for Extracting Semantic Information from Speech*. John Wiley and Sons, New York, NY, January.

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*.