Leveraging Non-Conversational Tasks for Low Resource Slot Filling: Does it help?

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

Bernardo Magnini
Fondazione Bruno Kessler
magnini@fbk.eu

Abstract

Slot filling is a core operation for utterance understanding in task-oriented dialogue systems. Slots are typically domain-specific, and adding new domains to a dialogue system involves data and time-intensive processes. A popular technique to address the problem is transfer learning, where it is assumed the availability of a large slot filling dataset for the source domain, to be used to help slot filling on the target domain, with fewer data. In this work, instead, we propose to leverage source tasks based on semantically related non-conversational resources (e.g., semantic sequence tagging datasets), as they are both cheaper to obtain and reusable to several slot filling domains. We show that using auxiliary non-conversational tasks in a multi-task learning setup consistently improves low resource slot filling performance.

1 Introduction

Language understanding in task-oriented dialogue systems involves recognizing information (i.e., *slot filling*) expressed in an utterance to accomplish a particular dialogue task. For example, in a flight booking scenario, the utterance *"show me all Delta flights from Milan to New York"* contains information belonging to slots in the flight domain, namely `airline_name` (*Delta*), `origin` (*Milan*), and `destination` (*New York*). Slots are usually predefined and domain-specific, e.g. in a hotel domain slots can be different, such as `room_type`, `length_of_stay` etc. Although recent neural based models (Goo et al., 2018; Wang et al., 2018; Liu and Lane, 2016) have shown remarkable performance in slot filling, they are still based on large labeled data, which means that training a separate model for each domain involves a resource intensive process. Thus, as more domains are added to the system, methods that can generalize slot filling to new domains with *limited labeled data* (i.e., low-resource settings) are preferable.

Existing works in low resource slot filling are mostly based on transfer learning (Mou et al., 2016), whose aim is to leverage relatively large resources in a source domain ($\mathcal{D}_S$) for a source task ($\mathcal{T}_S$), to help a task ($\mathcal{T}_T$) in a target domain ($\mathcal{D}_T$), where less data are available. Depending on how the adaptation is performed, there are two notable approaches: data-driven adaptation (Jaech et al., 2016; Goyal et al., 2018; Kim et al., 2016), and model-driven adaptation (Kim et al., 2017; Jha et al., 2018). Essentially, both approaches produce a model on the target domain performing training on the same task (slot filling, in our case), i.e., assuming ($\mathcal{T}_S = \mathcal{T}_T$), although from different domains, i.e. ($\mathcal{D}_S \neq \mathcal{D}_T$). All of these approaches assume that slot filling datasets for the source domain are available, and little effort has been devoted in finding and exploiting cheaper $\mathcal{T}_S$, which is crucial in a situation where a slot filling dataset in $\mathcal{D}_S$ is not ready yet (*cold-start*).

Accordingly, we attempt to leverage non-conversational source tasks ($\mathcal{T}_S \neq \mathcal{T}_T$) i.e., tasks that use widely available non-conversational resources, to help slot filling. These resources are cheaper to obtain compared to domain-specific slot filling datasets, and many of them are annotated with rich linguistic knowledge, which is potentially useful for slot filling (Chen et al., 2016). Among these resources, we mention PropBank (Palmer et al., 2005) and FrameNet (Baker et al., 1998), which consist of annotated documents with verb and frame-based semantic roles, respectively; CoNLL 2003 (Tjong Kim Sang and De Meulder, 2003) and OntoNotes (Pradhan et al., 2013), which provide named entity information; and Abstract Meaning Representation (AMR) (Banarescu et al., 2013), which provides a graph-based seman-

Proceedings of the SIGDial 2019 Conference, pages 85–91
Stockholm, Sweden, 11-13 September 2019.

Sentence	what	is	the	most	expensive	flight	from	boston	to	dallas
ATIS Slot	O	O	O	B-COST_REL	I-COST_REL	O	O	B-FROM_LOC	O	B-TO_LOC
NER	O	O	O	O	O	O	O	B-GPE	O	B-GPE
SemTag	B-QUE	B-ENS	B-DEF	B-TOP	B-IST	B-CON	B-REL	B-GPE	O	B-GPE

Table 1: An example of slot filling annotation from the ATIS (Airline Travel Information System) dataset and author-annotated NER and SemTag in IOB format (Ramshaw and Marcus, 1995). Some ATIS slots correspond to NER or SemTag labels, such as FROM_LOC and TO_LOC with GPE in NER and SemTag. Some slot tags can also be composed of several SemTag labels such as COST_REL which is composed of TOP (*superlative positive*) and IST (*intersective adjective*).

tic formalism.

In this work, we leverage non-conversational tasks as auxiliary tasks in a multi-task learning (MTL) (Caruana, 1997) setup. Given appropriate auxiliary tasks, MTL has shown to be particularly effective in which labeled data is scarce and has been applied to various NLP tasks such as parsing (Søgaard and Goldberg, 2016), POS tagging (Yang et al., 2016), neural machine translation (Luong et al., 2016), and opinion role labeling (Marasovic and Frank, 2018). While there are potentially many non-conversational tasks that we can use as auxiliary tasks, we focus on those that assign semantic class categories to a word, as they are similar in nature to slot filling. In particular, in this work we choose Named Entity Recognition (NER) and the recently introduced Semantic Tagging (SemTag) (Abzianidze and Bos, 2017), motivated by the following rationales:

- Both NER and SemTag are semantically related to slot filling. As illustrated in Table 1, slot labels may correspond to either NER or SemTag labels. In addition, SemTag complements NER as its labels subsume NER labels, and thus could be useful to address linguistic phenomena (e.g. comparative expression, intersective adjective) relevant for slot filling and that are beyond named entities.
- Both NER and SemTag can be re-used in many slot filling domains. Labels in both tasks are typically more general (coarse-grained) compared to labels in slot filling.
- The resources for both tasks are cheaper to obtain compared to domain-specific slot filling datasets, as there have been several initiatives in constructing large datasets for NER and SemTag, for example OntoNotes (Pradhan et al., 2013) and Parallel Meaning Bank (PMB) (Abzianidze et al., 2017) respectively. This is beneficial in a *cold-start* situation in which no slot filling dataset is already available in $\mathcal{D}_S$.

Although NER has been already used in slot filling models, most of these approaches (Mesnil et al., 2013, 2015; Zhang and Wang, 2016; Gong et al., 2019; Louvan and Magnini, 2018) use and incorporate ground truth NER labels or output of NER systems as features to train a slot filling model, our work differs in the method of learning and leveraging such features from disjoint datasets through MTL and evaluating the performance in low-resource settings.

Our contributions are: (i) we propose to leverage non-conversational tasks, namely NER and SemTag, to improve low resource slot filling through MTL; to our knowledge this MTL combination has not been explored before. (ii) We show that MTL models with NER and SemTag strongly improve single-task slot filling models on three well known datasets. While we focus on using NER and SemTag, our study has shed light on the potential use of non-conversational tasks in general to help low resource slot filling.

2 Approach

Slot filling is often modeled as a sequence labeling problem. Given a sequence of words $\mathbf{x} = (x_1, x_2, ..., x_n)$ as input, a model $\mathcal{M}$ predicts the corresponding slot labels $\mathbf{y} = (y_1, y_2, ..., y_n)$ as output.

2.1 Base Model

State-of-the-art models on sequence labeling are typically built based on bi-directional LSTM (bi-LSTM), on top of which there is a CRF model (Lample et al., 2016; Ma and Hovy, 2016). The bi-LSTM takes $\mathbf{x}$ as input and each word x_i is represented as an embedding $\mathbf{e}_i = [\mathbf{w}_i; \mathbf{c}_i]$ composed of the concatenation of a word embedding $\mathbf{w}_i$ and character embeddings $\mathbf{c}_i$. The bi-LSTM layer produces the forward output state $\overrightarrow{\mathbf{h}_i}$ and the backward output state $\overleftarrow{\mathbf{h}_i}$. The concatenation of the output states, $\mathbf{h}_i = [\overrightarrow{\mathbf{h}_i}; \overleftarrow{\mathbf{h}_i}]$, is then fed to a

feed-forward (FF) layer, followed by a CRF as the final output layer that predicts a slot label y_i by taking into account the mixture of context information captured by the last FF layer and the slot prediction y_{i-1} from the previous word.

2.2 Multi-task Learning Models

In the context of MTL, models for $\mathcal{T}_S$, often referred as **auxiliary tasks**, and for $\mathcal{T}_T$, referred as the **target task**, are simultaneously trained (Yang et al., 2017). In order to perform adaptation, the MTL model $\mathcal{M}$ is partitioned into task-specific parts ($\mathcal{M}_{\mathcal{T}_S}$ and $\mathcal{M}_{\mathcal{T}_T}$) and task-shared-parts ($\mathcal{M}_{\mathcal{T}_S \cap \mathcal{T}_T}$). We use two notable MTL architectures:

- **MTL-Fully Shared Network (MTL-FSN).** The word and character embeddings, and the bi-LSTM layers, are parts of $\mathcal{M}_{\mathcal{T}_S \cap \mathcal{T}_T}$. The hidden state outputs of the bi-LSTM are passed to each of the CRF output layers in $\mathcal{M}_{\mathcal{T}_S}$ and $\mathcal{M}_{\mathcal{T}_T}$. During training a mini-batch of a particular task, the output layers of other tasks are not updated.
- **Hierarchical-MTL (H-MTL).** Inspired by (Søgaard and Goldberg, 2016; Sanh et al., 2019), we introduce a hierarchy of tasks in $\mathcal{M}$ to create different levels of supervision. Instead of placing the output CRF layers for all tasks after the shared bi-LSTM layer, we add a task-specific bi-LSTM in $\mathcal{M}_{\mathcal{T}_T}$ after the shared bi-LSTM and then attach the output layer. In other words, we supervise $\mathcal{T}_S$, which have coarse-grained labels in the lower level output layer and $\mathcal{T}_T$, which has more fine-grained labels in the higher level output layer.

3 Experiments

The main objective of our experiments is to validate the hypothesis that using non-conversational tasks as auxiliary tasks in a MTL setup can help low resource slot filling. In our MTL configuration, the **target task** ($\mathcal{T}_T$) is slot filling, and the **auxiliary tasks** ($\mathcal{T}_S$) are set to NER or SemTag or both.

Baselines. We compare the two MTL approaches (see §2.2) with the following baselines:

- **Single-Task Learning (STL).** The base model is directly trained and tested on $\mathcal{T}_T$, without incorporating any information from $\mathcal{T}_S$. The base model (see §2.1) is a bi-LSTM-CRF which is the core of many models for slot filling (Goo et al., 2018; Wang et al., 2018; Liu and Lane, 2016) and sequence tagging tasks in general.
- **STL + Feature Based (STL + FB).** The same model as STL but incorporating the outputs of the independently trained NER and SemTag models as an additional feature in the input embeddings.

Dataset	Task	#train	#dev	#test	#label
ATIS	Slot Filling	4478	500	893	79
MIT Restaurant	Slot Filling	6128	1532	3385	8
MIT Movie	Slot Filling	7820	1955	2443	12
OntoNotes 5.0	NER	34970	5896	2327	18
PMB	SemTag	67965	682	650	73

Table 2: Statistics about the datasets, reporting the number of sentences in train/dev/test set, and the number of labels.

Datasets. The language of all the datasets that we use is English. We evaluate our approach on three slot filling datasets, namely ATIS (Price, 1990), MIT Restaurant, and Movie (Liu et al., 2013). ATIS is a widely used dataset for spoken language understanding which contains utterances requesting flight related information. While MIT Restaurant and Movie contain utterances requesting information related to restaurants and movies. For NER, we use the newswire section of OntoNotes 5.0 (Pradhan et al., 2012), which is compiled from English Wall St. Journal. For SemTag, we use Parallel Meaning Bank (PMB) (Abzianidze et al., 2017) 2.2.0. The PMB dataset is constructed from twelve different sources, including OPUS News Commentary (Tiedemann, 2012), Tatoeba[1], Sherlock-Holmes stories, Recognizing Textual Entailment (Giampiccolo et al., 2007), and the bible (Christodoulopoulos and Steedman, 2015). Following the previous publication related to SemTag (Abzianidze and Bos, 2017), we train the SemTag model using the silver data and test on gold data. For all datasets, we use the provided train/dev/test splits. Table 2 shows the overall statistics of each dataset. To simulate the low resource settings, in all experiments we only use 10% training data on $\mathcal{T}_T$.

Training. We do not tune the hyperparameters[2] but follow the suggestions and adapt the implementation of Reimers and Gurevych (2017)[3]. The MTL models are trained in an alternate fashion

[1] https://tatoeba.org/eng/

[2] The hyperparameters are listed in Appendix B

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

(Jaech et al., 2016) between $\mathcal{T}_T$ and $\mathcal{T}_S$. Consequently, as the training data size of $\mathcal{T}_S$ is larger than $\mathcal{T}_T$, the same $\mathcal{T}_T$ data is reused until the whole $\mathcal{T}_S$ is used in the training. We evaluate the performance by computing the F1-score on the test set using the standard CoNLL-2000 evaluation[4].

4 Results and Discussion

Model	$\mathcal{T}_S$	$\mathcal{T}_T$		
		ATIS	MIT-R	MIT-M
STL	-	$87.91_{0.56}$	$67.37_{0.26}$	$80.71_{0.63}$
STL+FB	-	$87.79_{0.67}$	$67.27_{0.64}$	$80.56_{0.54}$
MTL-FSN	*N*	$\mathbf{89.56}_{0.16}$	$68.82_{0.18}$	$80.77_{0.13}$
	S	$89.19_{0.26}$	$68.21_{0.71}$	$80.57_{0.32}$
	N,S	$89.10_{0.41}$	$68.21_{0.43}$	$79.69_{0.33}$
H-MTL	*N*	$89.17_{0.33}$	$\mathbf{69.22}_{1.00}$	$\mathbf{81.79}_{0.26}$
	S	$88.96_{0.41}$	$69.09_{0.24}$	$81.59_{0.17}$
	N,S	$88.78_{0.37}$	$68.96_{0.50}$	$81.15_{0.25}$

Table 3: Average F1-score and standard deviation (numbers in subscript) of the performance on the test sets. For the $\mathcal{T}_T$ training split, only 10% data is used. **Bold** indicates the best score for each $\mathcal{T}_T$. *N* and *S* in $\mathcal{T}_S$ denote NER and SemTag, respectively.

Overall Performance. Table 3 lists the overall performance of the baselines and of the MTL models. We report the average F-1 score and also the standard deviation, as recommended by Reimers and Gurevych (2018), over three runs from different random seeds. For all $\mathcal{T}_T$, it is evident that the MTL models with NER or SemTag combinations yield the best results compared to STL. MTL models also outperform the STL + FB baseline, indicating that training the model simultaneously with the auxiliary task is better than incorporating the output of the independently trained auxiliary models as features for the slot filling model. In terms of the effectiveness of the auxiliary tasks, using NER produces the best results compared to the other $\mathcal{T}_S$ combinations. The difference between MTL with NER and MTL with SemTag is marginal. Regarding the MTL models, on average, H-MTL yields better scores compared to MTL-FSN in MIT-R and MIT-M, which suggests that supervising tasks with coarse-grained labels and fine-grained labels on different layers is beneficial.

Slot-wise Performance. One of our motivations for using NER and SemTag is that their labels are coarse-grained, and that they can be re-used for several slot filling domains. We are interested to see whether MTL improves the performance of slots related to these coarse-grained concepts. In order to do this, we manually created a mapping[5] from the slots to some coarse-grained entity concepts used by CoNLL-2003 (Tjong Kim Sang and De Meulder, 2003) including Person, Organization, and Location. For example, in ATIS, the slot `airline_name` is mapped to Organization, the slot `fromloc.city_name` is mapped to Location, etc. We perform the analysis on the dev set by re-running the evaluation based on the mapping. Results in Table 4 show that in ATIS and MIT-R, MTL brings improvements on slots related to Location and Organization. However, MTL does not help in slots related to Person names in MIT-M. Based on our observation on the prediction results, most errors come from misclassifying `DIRECTOR` slots as `ACTOR` slots.

$\mathcal{T}_T$	Concept	Model	
		STL	MTL
ATIS	LOC	$94.74_{0.37}$	$\mathbf{95.82}_{0.34}$
	ORG	$92.52_{0.89}$	$\mathbf{93.37}_{0.29}$
MIT-R	LOC	$75.29_{0.46}$	$\mathbf{76.02}_{0.39}$
MIT-M	PER	$\mathbf{85.04}_{0.24}$	$84.58_{0.56}$

Table 4: Performance on slots related to person (PER), location (LOC), and organization (ORG) concepts. We use the best MTL from Table 3 for each $\mathcal{T}_T$.

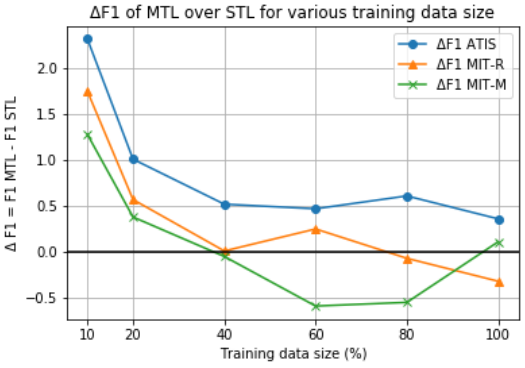

Figure 1: Gain ($\Delta F1$) obtained using MTL over STL on increasing training data. Positive numbers mean MTL is better, negative numbers mean MTL is worse. We use the best MTL from Table 3 for each $\mathcal{T}_T$.

Performance Gain on Increasing Data Size. We also carried on an experiment by increasing

[4]https://www.clips.uantwerpen.be/conll2000/

[5]We provide the mapping in Appendix A

the amount of training data on $\mathcal{T}_\mathcal{T}$, and evaluated the performance on the dev set to understand the usefulness of MTL on varying data size. As shown in Figure 1, as we increase the size of the training data, the gain that we obtain using MTL tends to decrease. The results suggest that MTL is indeed more useful in very low resource scenarios, according to our initial hypothesis. After 40% training data size is used (around 2K utterances), MTL is less useful. We believe that this is because the slot filling datasets are relatively simple, e.g. the texts are short and most of them express a single specific request, thus, it is relatively easy for the model to capture the regularities.

Impact on Auxiliary Tasks Performance. We also perform an analysis to understand the effect of MTL to the model performance for $\mathcal{T}_\mathcal{S}$. The STL performance of OntoNotes and Semantic Tagging are around 89% and 96% respectively in terms of F1-score. With MTL, on average, the $\mathcal{T}_\mathcal{S}$ model performance decrease about 0.7 points for OntoNotes and 0.2 points for Semantic Tagging. This suggests that $\mathcal{T}_\mathcal{S}$ models do not benefit from the low resource $\mathcal{T}_\mathcal{T}$ through the MTL framework and the training mechanism that we use. In general, whether MTL can benefit model performance in a target task given auxiliary tasks (or vice versa) is still a question and beyond the scope of this paper. While there is no exact answer yet for this question, we refer to (Bingel and Søgaard, 2017; Alonso and Plank, 2017) which study the characteristics of auxiliary tasks that is potential to help target task performance (Bingel and Søgaard, 2017; Alonso and Plank, 2017) .

5 Conclusions

We proposed to leverage non-conversational tasks, Named Entity Recognition and Semantic Tagging, through multi-task learning to help low resource slot filling. Our experiments demonstrate that: (i) non-conversational tasks are effective to improve slot filling performance, and they are reusable in different slot filling domains; (ii) incorporating a task-hierarchy in the multi-task architecture based on the granularity of the labels is beneficial for the model performance on two out of three datasets.

In the future, we plan to explore other non-conversational resources such as FrameNet (Baker et al., 1998) which provide a repository of event frames and semantic roles that can be relevant for intent classification and slot filling in task-oriented dialogue systems. Also another direction is to apply fine-tuning with the recently popular pre-trained language model e.g. BERT (Devlin et al., 2018).

References

Lasha Abzianidze, Johannes Bjerva, Kilian Evang, Hessel Haagsma, Rik van Noord, Pierre Ludmann, Duc-Duy Nguyen, and Johan Bos. 2017. The Parallel Meaning Bank: Towards a Multilingual Corpus of Translations Annotated with Compositional Meaning Representations. In *EACL*.

Lasha Abzianidze and Johan Bos. 2017. Towards Universal Semantic Tagging. In *IWCS 2017 — 12th International Conference on Computational Semantics — Short papers*.

Hector Martinez Alonso and Barbara Plank. 2017. When is multitask learning effective? semantic sequence prediction under varying data conditions. In *EACL 2017-15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1–10.

Collin F Baker, Charles J Fillmore, and John B Lowe. 1998. The Berkeley Framenet Project. In *Proceedings of the 17th international conference on Computational linguistics-Volume 1*, pages 86–90. Association for Computational Linguistics.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract Meaning Representation for Sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 178–186.

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.

Yun-Nung Chen, Dilek Hakanni-Tur, Gökhan Tür, Asli Çelikyilmaz, Jianfeng Gao, and Li Deng. 2016. Syntax or Semantics? Knowledge-guided Joint Semantic Frame Parsing. In *2016 IEEE Spoken Language Technology Workshop, SLT 2016, San Diego, CA, USA, December 13-16, 2016*, pages 348–355.

Christos Christodoulopoulos and Mark Steedman. 2015. A massively parallel corpus: the bible in 100 languages. In *Language Resources and Evaluation*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *CoRR*, abs/1810.04805.

Danilo Giampiccolo, Bernardo Magnini, Ido Dagan, and Bill Dolan. 2007. The third PASCAL recognizing textual entailment challenge. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 1–9, Prague. Association for Computational Linguistics.

Yu Gong, Xusheng Luo, Yu Zhu, Wenwu Ou, Zhao Li, Muhua Zhu, Kenny Q. Zhu, Linfu Duan, and Xi Chen. 2019. Deep cascade multi-task learning for slot filling in online shopping assistant. In *AAAI 2019*.

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, Volume 2 (Short Papers)*, pages 753–757. Association for Computational Linguistics.

Anuj Kumar Goyal, Angeliki Metallinou, and Spyros Matsoukas. 2018. Fast and Scalable Expansion of Natural Language Understanding Functionality for Intelligent Agents . In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 3 (Industry Papers)*, pages 145–152. Association for Computational Linguistics.

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)*, pages 153–161. 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)*, pages 643–653. Association for Computational Linguistics.

Young-Bum Kim, Karl Stratos, and Ruhi Sarikaya. 2016. Domainless Adaptation by Constrained Decoding on a Schema Lattice. In *COLING 2016, 26th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, December 11-16, 2016, Osaka, Japan*, pages 2051–2060.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270. Association for Computational Linguistics.

Bing Liu and Ian Lane. 2016. Joint Online Spoken Language Understanding and Language Modeling With Recurrent Neural Networks. In *Proceedings of the SIGDIAL 2016 Conference, The 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue, 13-15 September 2016, Los Angeles, CA, USA*, pages 22–30.

Jingjing Liu, Panupong Pasupat, Scott Cyphers, and Jim Glass. 2013. Asgard: A Portable Architecture for Multilingual Dialogue Systems. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 8386–8390. IEEE.

Samuel Louvan and Bernardo Magnini. 2018. From General to Specific : Leveraging Named Entity Recognition for Slot Filling in Conversational Language Understanding. In *CLiC-it*.

Minh-Thang Luong, Quoc V. Le, Ilya Sutskever, Oriol Vinyals, and Lukasz Kaiser. 2016. Multi-task sequence to sequence learning. *ICLR*, abs/1511.06114.

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.

Ana Marasovic and Anette Frank. 2018. SRL4ORL: Improving Opinion Role Labelling using Multi-task Learning with Semantic Role Labeling. In *NAACL-HLT*.

Grégoire Mesnil, Yann Dauphin, Kaisheng Yao, Yoshua Bengio, Li Deng, Dilek Hakkani-Tur, Xiaodong He, Larry Heck, Gokhan Tur, Dong Yu, and Geoffrey Zweig. 2015. Using recurrent neural networks for slot filling in spoken language understanding. *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, 23(3):530–539.

Grégoire Mesnil, Xiaodong He, Li Deng, and Yoshua Bengio. 2013. Investigation of recurrent-neural-network architectures and learning methods for spoken language understanding. In *INTERSPEECH*.

Lili Mou, Zhao Meng, Rui Yan, Ge Li, Yan Xu, Lu Zhang, and Zhi Jin. 2016. How Transferable are Neural Networks in NLP Applications? In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 479–489. Association for Computational Linguistics.

Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The Proposition Bank: An Annotated Corpus of Semantic Roles. *Computational linguistics*, 31(1):71–106.

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.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Olga Uryupina, and Yuchen Zhang. 2012. CoNLL-2012 Shared Task: Modeling Multilingual Unrestricted Coreference in OntoNotes. In *EMNLP-CoNLL Shared Task*.

Patti J Price. 1990. Evaluation of Spoken Language Systems: The ATIS Domain. In *Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania, June 24-27, 1990*.

Lance Ramshaw and Mitch Marcus. 1995. Text Chunking using Transformation-Based Learning. In *Third Workshop on Very Large Corpora*.

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.

Nils Reimers and Iryna Gurevych. 2018. Why Comparing Single Performance Scores Does Not Allow to Draw Conclusions About Machine Learning Approaches. *CoRR*, abs/1803.09578.

Victor Sanh, Thomas Wolf, and Sebastian Ruder. 2019. A Hierarchical Multi-task Approach for Learning Embeddings from Semantic Tasks. *AAAI*.

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.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in opus. In *LREC*.

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.

Yu Wang, Yilin Shen, and Hongxia Jin. 2018. A bi-model based rnn semantic frame parsing model for intent detection and slot filling. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 309–314. Association for Computational Linguistics.

Zhilin Yang, Ruslan Salakhutdinov, and William W. Cohen. 2016. Transfer Learning for Sequence Tagging with Hierarchical Recurrent Networks. *CoRR*, abs/1703.06345.

Zhilin Yang, Ruslan Salakhutdinov, and William W Cohen. 2017. Transfer Learning for Sequence Tagging with Hierarchical Recurrent Networks. In *ICLR*.

Xiaodong Zhang and Houfeng Wang. 2016. A joint model of intent determination and slot filling for spoken language understanding. In *IJCAI*.

A Mapping of entity concepts and slots for each dataset

Concept	ATIS	MIT-R	MIT-M
LOC	fromloc.airport_code fromloc.airport_name fromloc.city_name fromloc.state_code fromloc.state_name stoploc.airport_name stoploc.city_name stoploc.state_code toloc.airport_code toloc.airport_name toloc.city_name toloc.country_name toloc.state_code toloc.state_name	location	-
ORG	airline_name	-	-
PER	-	-	character actor director

Table 5: The mapping of entity concepts, namely Location (LOC), Organization (ORG), and Person (PER) to their corresponding slots in each dataset.

B Hyperparameters

Hyperparameter	Value
LSTM cell size	100
Dropout	0.5
Word embedding dimension	300
Character embedding dimension	100
Mini-batch size	32
Optimizer	Adam
Number of epoch	50
Early stopping	10