Span-ConveRT: Few-shot Span Extraction for Dialog with Pretrained Conversational Representations

Sam Coope[1]*, **Tyler Farghly**[2]*, **Daniela Gerz**[1], **Ivan Vulić**[1,3], **Matthew Henderson**[1]
[1] PolyAI Limited, London, UK
[2] Imperial College London, UK
[3] Language Technology Lab, University of Cambridge, UK
sam@polyai.com

Abstract

We introduce *Span-ConveRT*, a light-weight model for dialog slot-filling which frames the task as a turn-based span extraction task. This formulation allows for a simple integration of conversational knowledge coded in large pretrained conversational models such as ConveRT (Henderson et al., 2019a). We show that leveraging such knowledge in Span-ConveRT is especially useful for few-shot learning scenarios: we report consistent gains over **1)** a span extractor that trains representations from scratch in the target domain, and **2)** a BERT-based span extractor. In order to inspire more work on span extraction for the slot-filling task, we also release RESTAURANTS-8K, a new challenging data set of 8,198 utterances, compiled from actual conversations in the restaurant booking domain.

1 Introduction

Conversational agents are finding success in a wide range of well-defined tasks such as customer support, restaurant, train or flight bookings (Hemphill et al., 1990; Williams, 2012; El Asri et al., 2017; Budzianowski et al., 2018), language learning (Raux et al., 2003; Chen et al., 2017), and also in domains such as healthcare (Laranjo et al., 2018) or entertainment (Fraser et al., 2018). Scaling conversational agents to support new domains and tasks, and particular system behaviors is a highly challenging and resource-intensive task: it critically relies on expert knowledge and domain-specific labeled data (Williams, 2014; Wen et al., 2017b,a; Liu et al., 2018; Zhao et al., 2019).

Slot-filling is a crucial component of any task-oriented dialog system (Young, 2002, 2010; Bellegarda, 2014). For instance, a conversational agent for restaurant bookings must fill all the slots *date*, *time* and *number of guests* with correct values given by the user (e.g. *tomorrow*, *8pm*, *3 people*) in order to proceed with a booking. A particular challenge is to deploy slot-filling systems in *low-data regimes* (i.e., *few-shot learning* setups), which is needed to enable quick and wide portability of conversational agents. Scarcity of in-domain data has typically been addressed using domain adaption from resource-rich domains, e.g. through multi-task learning (Jaech et al., 2016; Goyal et al., 2018) or ensembling (Jha et al., 2018; Kim et al., 2019).

In this work, we approach slot-filling as a *turn-based span extraction* problem similar to Rastogi et al. (2019): in our *Span-ConveRT* model we do not restrict values to fixed categories, and simultaneously allow the model to be entirely independent of other components in the dialog system. In order to facilitate slot-filling in resource-lean settings, our main proposal is the effective use of knowledge coded in representations transferred from large general-purpose conversational pretraining models, e.g., the ConveRT model trained on a large Reddit data set (Henderson et al., 2019a).

To help guide other work on span extraction-based slot-filling, we also present a new data set of 8,198 user utterances from a commercial restaurant booking system: RESTAURANTS-8K. The data set spans 5 slots (*date*, *time*, *people*, *first name*, *last name*) and consists of actual user utterances collected "in the wild". This comes with a broad range of natural and colloquial expressions,[1] as illustrated in Figure 1, which makes it both a natural and challenging benchmark. Each training example is a dialog turn annotated with the slots requested by the system and character-based span indexing for all occurring values.

As our key findings show, conversational pre-

*Both authors contributed equally to the work. The work of TF was done during an internship at PolyAI.

[1]For instance, a value for the slot *people* can either be a number like *7*, or can be expressed fully in natural language, e.g., *me and my husband*.

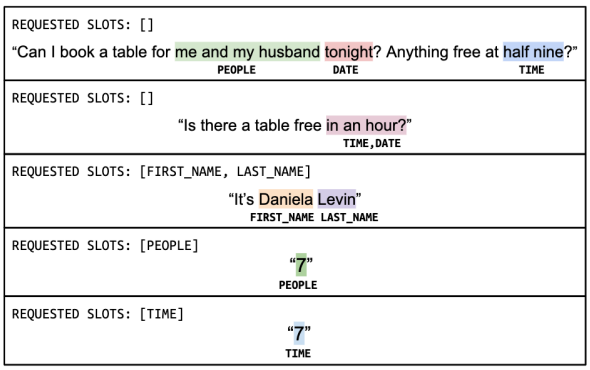

Figure 1: Turn-based span extraction with the new RESTAURANTS-8K data set. Note how the requested slot feature is needed to differentiate time or party size in short utterances like "7". The single-turn examples are extracted from different conversations.

training is instrumental to span extraction performance in few-shot setups. By using subword representations transferred from ConveRT (Henderson et al., 2019a), we demonstrate that: 1) our ConveRT-backed span extraction model outperforms the model based on transferred BERT representations, and 2) it also yields consistent gains over a span extraction model trained from scratch in the target domains, with large gains reported in few-shot scenarios. We verify both findings on the new RESTAURANTS-8K data set, as well as on four DSTC8-based data sets (Rastogi et al., 2019). All of the data sets used in this work are available online at: https://github.com/PolyAI-LDN/task-specific-datasets.

2 Methodology: Span-ConveRT

Before we delve into describing the core methodology, we note that in this work we are not concerned with the task of normalizing extracted spans to their actual values: this can be solved effectively with rule-based systems after the span extraction step for cases such as times, dates, and party sizes. There exist hierarchical rule-based parsing engines (e.g., Duckling) that allow for parsing times and dates such as *"the day after next Tuesday"*. Further, phrases such as *"Me and my wife and 2 kids"* can be parsed using singular noun and number counts in the span with high precision.

Span Extraction for Dialog. We have recently witnessed increasing interest in *intent-restricted* approaches (Coucke et al., 2018; Goo et al., 2018; Chen et al., 2019) for slot-filling. In this line of work, slot-filling is treated as a span extraction problem where slots are defined to occur only with certain intents. This solves the issue of complex categorical modeling but makes slot-filling dependent on an intent detector. Therefore, we propose a framework that treats slot-filling as a fully *intent-agnostic* span extraction problem. Instead of using rules to constrain the co-occurrence of slots and intents, we identify a slot as either a single span of text or entirely absent. This makes our approach more flexible than prior work; it is fully independent of other system components. Regardless, we can explicitly capture turn-by-turn context by adding an input feature denoting whether a slot was requested for this dialog turn (see Figure 1).

Pretrained Representations. Large-scale pretrained models have shown compelling benefits in a plethora of NLP applications (Devlin et al., 2019; Liu et al., 2019): such models drastically lessen the amount of required task/domain-specific training data with in-domain fine-tuning. This is typically achieved by adding a task-specific output layer to a large pretrained encoder and then fine-tuning the entire model (Xie et al., 2019). However, this process requires a fine-tuned model for each slot or domain, rather than a single model shared across all slots and domains. This adds a large memory and computational overhead and makes the approach impractical in real-life applications. Therefore, we propose to keep the pretrained encoder models fixed in order to emulate a production system where a single encoder model is used.[2]

Underlying Representation Model: ConveRT. ConverRT (Henderson et al., 2019a) is a lightweight sentence encoder implemented as a dual-encoder network that models the interaction between inputs/contexts and relevant (follow-up) responses. In other words, it performs conversational pretraining based on response selection on the Reddit corpus (Henderson et al., 2019a,b). It utilizes subword-level tokenization and is very compact and resource-efficient (i.e. it is 59MB in size and can be trained in less than 1 day on 12 GPUs) while achieving state-of-the-art performance on conversational tasks (Casanueva et al., 2020; Bunk et al., 2020). Through pretrained ConveRT representa-

[2] In other words, we do not fine-tune the parameters of the pretrained encoders which would require running a separate encoder for each slot. This would mean, for example, we would need 100 fine-tuned encoders running in production to support 100 different slots. As the encoder models have both high memory and runtime requirements, this would drastically increase the running costs of a conversational system.

tions, we can leverage conversational cues from over 700M conversational turns for the few-shot span extraction task.[3]

Span ConveRT: Final Model. We now describe our model architecture, illustrated in Figure 2. Our approach builds on established sequence tagging models using Conditional Random Fields (CRFs) (Ma and Hovy, 2016; Lample et al., 2016). We propose to replace the LSTM part of the model with fixed ConveRT embeddings.[4] We take contextualized subword embeddings from ConveRT, giving a sequence of the same length as the subword-tokenized sentence. For sequence tagging, we train a CNN and CRF on top of these fixed subword representations. We concatenate three binary features to the subword representations to emphasize important textual characteristics: (1) whether the token is alphanumeric, (2) numeric, or (3) the start of a new word. In addition, we concatenate the character length of the token as another integer feature. To incorporate the requested slots feature, we concatenate a binary feature representing if the slot is requested to each embedding in the sequence. To contextualize the modified embeddings, we apply a dropout layer followed by a series of 1D convolutions of increasing filter width.

Spans are represented using a sequence of *tags*, indicating which members of the subword token sequence are in the span. We use a tag representation similar to the IOB format annotating the span with a sequence of *before*, *begin*, *inside* and *after* tags, see Figure 2 for an example.

The distribution of the tag sequence is modeled with a CRF, whose parameters are predicted by a CNN that runs over the contextualized subword embeddings $\mathbf{v}$. At each step t, the CNN outputs a 4×4 matrix of transition scores $\mathbf{W}_t$ and a 4-dimensional vector of unary potentials $\mathbf{u}_t$. The probability of a predcited tag sequence $\mathbf{y}$ is then modeled as:

$$p(\mathbf{y}|\mathbf{v}) \propto \prod_{t=1}^{T-1} \exp\left(\mathbf{W}_t|y_{t+1}, y_t\right) \prod_{t=1}^{T} \exp\left(\mathbf{u}_t|y_t\right)$$

The loss is the negative log-likelihood, equal to minus the sum of the transition scores and unary potentials that correspond to the true tag labels, up to a normalization term. The top scoring tag sequences can be computed efficiently using the Viterbi algorithm.

[3] As we show later in §4, we can also leverage BERT-based representations in the same span extraction framework, but our ConveRT-based span extractors result in higher performance.

[4] LSTMs are known to be computationally expensive and require large amounts of resources to obtain any notable success (Pascanu et al., 2013). By utilizing ConveRT instead, we arrive at a much more lightweight and efficient model.

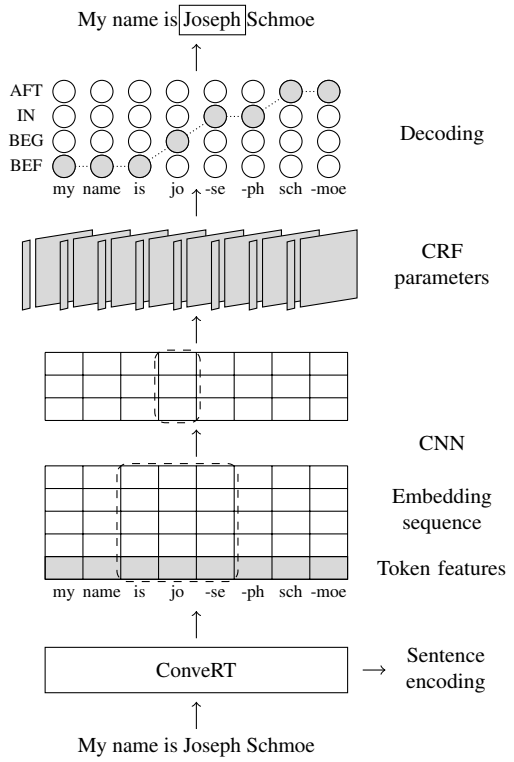

Figure 2: Span-ConveRT model architecture. Contextual subword embeddings, computed by ConveRT, are augmented with token features, and fed through a CNN. The outputs of the CNN parameterise a CRF sequence model, defining a distribution over sequence tag labellings, using the *before*, *begin*, *inside*, *after* scheme. Dashed lines denote CNN kernels.

	people	time	date	first_name	last_name	total
train	**2164** (547)	**2164** (547)	**1721** (601)	**887** (364)	**891** (353)	8198
dev	**983** (244)	**853** (276)	**802** (300)	**413** (177)	**426** (174)	3731

Table 1: The number of examples for each slot in the RESTAURANTS-8K data set. Numbers in brackets show how many examples have the slot requested.

3 Experimental Setup

New Evaluation Data Set: RESTAURANTS-8K. Data sets for task-oriented dialog systems typically annotate slots with exclusively categorical labels (Budzianowski et al., 2018). While some data sets such as SNIPS (Coucke et al., 2018) or ATIS (Tür et al., 2010) do contain span annotations, they are built with single-utterance voice commands in mind rather than a natural multi-turn dialog. To fill this gap and enable more work on span extraction for dialog, we introduce a new data

Hyperparameter	ConveRT	BERT	Vanilla
Dimensionality of the input subword embeddings	512	768	32
Size of minibatches during training	16	16	64
The learning rate for the SGD optimizer	0.01	0.01	0.1
Keep probability of elements in the sub-word embedding	0.5	0.9	0.5
Keep probability of elements in the sub-word feature embeddings	0.6	0.6	0.5
The size of the subword-CNN filters	(128, 64)	(128, 64)	(100, 100, 100)
Width of the subword CNN filters	(1, 5)	(1, 5)	(8, 4, 1)
Activation function for subword CNN	swish	swish	swish

Table 2: The final hyper-parameters used for different subword representations; *swish* refers to swish activation taken from Ramachandran et al. (2017).

Fraction	Span-ConveRT	V-CNN-CRF	Span-BERT
1 (8198)	**0.96**	0.94	0.92
1/2 (4099)	**0.95**	0.92	0.91
1/4 (2049)	**0.93**	0.89	0.87
1/8 (1024)	**0.90**	0.85	0.80
1/16 (512)	**0.81**	0.75	0.71
1/32 (256)	**0.64**	0.57	0.47
1/64 (128)	**0.55**	0.39	0.23
1/128 (64)	**0.41**	0.26	0.17

Table 3: Average F_1 scores across all slots for RESTAURANTS-8K with varying training set fractions. Numbers in brackets represent training set sizes.

set called RESTAURANTS-8K. It comprises conversations from a commercial restaurant booking system, and covers 5 slots essential for the booking task: *date*, *time*, *people*, *first name*, *last name*. The data statistics are provided in Table 1.[5]

DSTC8 Data Sets. The Schema-Guided Dialog Dataset (SGDD) (Rastogi et al., 2019) released for DSTC8 contains span annotations for a subset of slots. We extract span annotated data sets from SGDD in four different domains based on their large variety of slots: (1) *bus and coach booking* (labelled *Buses_1*), (2) *buying tickets for events* (*Events_1*), (3) *property viewing* (*Homes_1*) and *renting cars* (*RentalCars_1*). A detailed description of the data extraction protocol and the statistics of the data sets, also released with this paper, are available in appendix A.

Baseline Models. We compare our proposed model with two strong baselines: **V-CNN-CRF** is a vanilla approach that uses no pretrained model and instead learns sub-word representations from scratch. **Span-BERT** uses fixed BERT subword representations. All use the same CNN+CRF architecture on top of the subword representations. For each baseline, we conduct hyper-parameter optimization similar to Span-ConveRT: this is done via grid search and evaluation on the development set of RESTAURANTS-8K. The final sets of hyper-parameters are provided in Table 2. Span-BERT relies on BERT-base, with 12 transformer layers and 768-dim embeddings. ConveRT uses 6 transformer layers with 512-dim embeddings, so it is roughly 3 times smaller.

Following prior work (Coucke et al., 2018; Rastogi et al., 2019), we report the F_1 scores for extracting the correct span per user utterance. If the models extract part of the span or a longer span, this is treated as an incorrect span prediction.

Few-Shot Scenarios. For both data sets, we measure performance on smaller sets sampled from the full data. We gradually decrease training sets in size whilst maintaining the same test set: this provides insight on performance in low-data regimes.

4 Results and Discussion

The results across all slots are summarized in Table 3 for RESTAURANTS-8K, and in Table 4 for DSTC8. First, we note the usefulness of conversational pretraining and transferred representations: Span-ConveRT outperforms the two baselines in almost all evaluation runs, and the gain over V-CNN-CRF directly suggests the importance of transferred pretrained *conversational* representations. Second, we note prominent gains with Span-ConveRT especially in few-shot scenarios with reduced training data: e.g., the gap over V-CNN-CRF widens from 0.02 on the full RESTAURANTS-8K training set to 0.15 when using only 64 training examples. Simi-

[5]The data set contains some challenging examples where multiple values are mentioned, or values are mentioned that do not pertain to a slot. For example, in the utterance *"I said 5pm not 6pm"* multiple times are mentioned; in *"I called earlier today"* a date is mentioned that is not the day of the booking. Further, there are noticeable differences compared to previous data sets such as DSTC8 (Rastogi et al., 2019): e.g., while all slots in other datasets which pertained to integers (e.g. the number of travelers for a coach journey, number of tickets for an event booking) are modeled categorically (i.e. all numbers from 1 to 10 are separate classes), we model the number of people coming for a booking using spans because people often mention this value indirectly. For example *me and my husband*, *3 adults, 4 kids*, *2 couples*.

Fraction	Span-ConveRT	V-CNN-CRF	Span-BERT
Buses_1			
1 (1133)	0.92	**0.93**	0.89
1/2 (566)	**0.87**	0.83	0.84
1/4 (283)	**0.87**	0.77	0.80
1/8 (141)	**0.79**	0.71	0.62
1/16 (70)	**0.60**	0.53	0.44
Events_1			
1 (1498)	**0.92**	**0.92**	0.79
1/2 (749)	**0.86**	0.84	0.73
1/4 (374)	**0.81**	0.77	0.70
1/8 (187)	**0.65**	0.54	0.36
1/16 (93)	**0.66**	0.52	0.42
Homes_1			
1 (2064)	**0.98**	0.95	0.97
1/2 (1032)	**0.96**	0.90	0.94
1/4 (516)	**0.95**	0.88	0.87
1/8 (258)	**0.92**	0.82	0.80
1/16 (129)	**0.88**	0.69	0.70
RentalCars_1			
1 (874)	**0.91**	0.89	0.89
1/2 (437)	**0.87**	0.83	0.82
1/4 (218)	**0.81**	0.69	0.74
1/8 (109)	**0.75**	0.59	0.56
1/16 (54)	**0.62**	0.31	0.38

Table 4: Average F_1 scores on the DSTC8 single-domain datasets. A full breakdown of results for each individual slot is available in appendix B.

lar trends are observed on all four DSTC8 subsets. Again, this indicates that general-purpose conversational knowledge coded in ConveRT can indeed boost dialog modeling in low-data regimes. If sufficient domain-specific data is available (e.g., see the results of V-CNN-CRF with full data), learning domain-specialized representations from scratch can lead to strong performance, but using transferred conversational representations seems to be widely useful and robust.

We also observe consistent gains over Span-BERT, and weaker performance of Span-BERT even in comparison to V-CNN-CRF in some runs (see Table 3). These results indicate that for conversational end-applications such as slot-filling, pretraining on a *conversational* task (such as response selection) is more beneficial than standard language modeling-based pretraining. Our hypothesis is that both the vanilla baseline and ConveRT leverage some "domain adaptation": ConveRT is trained on rich conversational data, while the baseline representations are learned directly on the training data. BERT, on the other hand, is not trained on conversational data directly and usually relies on much longer passages of text. This might not make the BERT representations suitable for conversational tasks such as span extraction. Similar findings, where ConveRT-based conversational representations outperform BERT-based baselines (even with full fine-tuning), have recently been established in other dialog tasks such as intent detection (Henderson et al., 2019a; Casanueva et al., 2020; Bunk et al., 2020). In general, our findings also call for investing more effort in investigating different pretraining strategies that are better aligned to target tasks (Mehri et al., 2019; Henderson et al., 2019a; Humeau et al., 2020).

Error Analysis. To better understand the performance of Span-ConveRT on the RESTAURANTS-8K data set, we also conducted a manual error analysis, comparing it with the best performing baseline model, V-CNN-CRF. In Appendix C we lay out the types of errors that occur in a generic span extraction task and investigate the distribution of these types of errors across slots and models. We show that when trained in the high-data setting the distribution is similar between the two models, suggesting that gains from Span-ConveRT are across all types of error. We also show that the distribution varies more in the low-data setting and discuss how that might impact their comparative performance in practice. Additionally, in Appendix D we provide a qualitative analysis on the errors the two models make for the slot *first name*. We show that the baseline model has a far greater tendency to wrongly identify generic out-of-vocabulary words as names.

5 Conclusion and Future Work

We have introduced *Span-ConveRT*, a light-weight model for dialog slot-filling that approaches the problem as a turn-based span extraction task. The formulation allows the model to effectively leverage representations available from large-scale conversational pretraining. We have shown that, due to pretrained representations, Span-ConveRT is especially useful in few-shot learning setups on small data sets. We have also introduced RESTAURANTS-8K, a new challenging data set that will hopefully encourage further work on span extraction for dialogue. In future work, we plan to experiment with multi-domain span extraction architectures.

Acknowledgments

We thank the three anonymous reviewers for their helpful suggestions and feedback. We are grateful to our colleagues at PolyAI, especially Georgios Spithourakis and Iñigo Casanueva, for many fruitful discussions and suggestions.

References

Jerome R. Bellegarda. 2014. Spoken language understanding for natural interaction: The siri experience. In *Natural Interaction with Robots, Knowbots and Smartphones, Putting Spoken Dialog Systems into Practice*, pages 3–14.

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. 2018. MultiWOZ - A large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. In *Proceedings of EMNLP*, pages 5016–5026.

Tanja Bunk, Daksh Varshneya, Vladimir Vlasov, and Alan Nichol. 2020. DIET: Lightweight language understanding for dialogue systems. *CoRR*, abs/2004.09936.

Iñigo Casanueva, Tadas Temcinas, Daniela Gerz, Matthew Henderson, and Ivan Vulić. 2020. Efficient intent detection with dual sentence encoders. *CoRR*, abs/2003.04807.

Hongshen Chen, Xiaorui Liu, Dawei Yin, and Jiliang Tang. 2017. A survey on dialogue systems: Recent advances and new frontiers. *CoRR*, abs/1711.01731.

Qian Chen, Zhu Zhuo, and Wen Wang. 2019. BERT for joint intent classification and slot filling. *CoRR*, abs/1902.10909.

Alice Coucke, Alaa Saade, Adrien Ball, Théodore Bluche, Alexandre Caulier, David Leroy, Clément Doumouro, Thibault Gisselbrecht, Francesco Caltagirone, Thibaut Lavril, et al. 2018. Snips Voice Platform: An embedded spoken language understanding system for private-by-design voice interfaces. *arXiv preprint arXiv:1805.10190*, pages 12–16.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186.

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 SIGDIAL*, pages 207–219.

Jamie Fraser, Ioannis Papaioannou, and Oliver Lemon. 2018. Spoken conversational AI in video games: Emotional dialogue management increases user engagement. In *Proceedings of IVA*.

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 NAACL-HLT*, pages 753–757.

Anuj Kumar Goyal, Angeliki Metallinou, and Spyros Matsoukas. 2018. Fast and scalable expansion of natural language understanding functionality for intelligent agents. In *Proceedings of NAACL-HLT*, pages 145–152.

Charles T. Hemphill, John J. Godfrey, and George R. Doddington. 1990. The ATIS Spoken Language Systems Pilot Corpus. In *Proceedings of the Workshop on Speech and Natural Language*, HLT '90, pages 96–101.

Matthew Henderson, Iñigo Casanueva, Nikola Mrkšić, Pei-Hao Su, Tsung-Hsien Wen, and Ivan Vulić. 2019a. ConveRT: Efficient and accurate conversational representations from transformers. *CoRR*, abs/1911.03688.

Matthew Henderson, Ivan Vulić, Daniela Gerz, Iñigo Casanueva, Paweł Budzianowski, Sam Coope, Georgios Spithourakis, Tsung-Hsien Wen, Nikola Mrkšić, and Pei-Hao Su. 2019b. Training neural response selection for task-oriented dialogue systems. In *Proceedings of ACL*, pages 5392–5404.

Samuel Humeau, Kurt Shuster, Marie-Anne Lachaux, and Jason Weston. 2020. Poly-encoders: Transformer architectures and pre-training strategies for fast and accurate multi-sentence scoring. In *Proceedings of ICLR*.

Aaron Jaech, Larry P. Heck, and Mari Ostendorf. 2016. Domain adaptation of recurrent neural networks for natural language understanding. In *Proceedings of INTERSPEECH*, pages 690–694.

Rahul Jha, Alex Marin, Suvamsh Shivaprasad, and Imed Zitouni. 2018. Bag of experts architectures for model reuse in conversational language understanding. In *Proceedings of NAACL-HLT*, pages 153–161.

Kunho Kim, Rahul Jha, Kyle Williams, Alex Marin, and Imed Zitouni. 2019. Slot tagging for task oriented spoken language understanding in human-to-human conversation scenarios. In *Proceedings of CoNLL*, pages 757–767.

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.

Liliana Laranjo, Adam G. Dunn, Huong Ly Tong, Ahmet Baki Kocaballi, Jessica Chen, Rabia Bashir, Didi Surian, Blanca Gallego, Farah Magrabi, Annie Y.S. Lau, and Enrico Coiera. 2018. Conversational agents in healthcare: A systematic review. *Journal of the American Medical Informatics Association*, 25(9):1248–1258.

Bing Liu, Gökhan Tür, Dilek Hakkani-Tür, Pararth Shah, and Larry P. Heck. 2018. Dialogue learning with human teaching and feedback in end-to-end trainable task-oriented dialogue systems. In *Proceedings of NAACL-HLT*, pages 2060–2069.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. In *Proceedings of ACL*, pages 1064–1074.

Shikib Mehri, Evgeniia Razumovskaia, Tiancheng Zhao, and Maxine Eskenazi. 2019. Pretraining methods for dialog context representation learning. In *Proceedings of ACL*, pages 3836–3845.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. 2013. On the difficulty of training recurrent neural networks. In *Proceedings of ICML*, pages 1310–1318.

Prajit Ramachandran, Barret Zoph, and Quoc V. Le. 2017. Searching for activation functions. *CoRR*, abs/1710.05941.

Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, Raghav Gupta, and Pranav Khaitan. 2019. Towards scalable multi-domain conversational agents: The schema-guided dialogue dataset. *arXiv preprint arXiv:1909.05855*.

Antoine Raux, Brian Langner, Alan W. Black, and Maxine Eskénazi. 2003. LET's GO: Improving spoken dialog systems for the elderly and non-natives. In *Proceedings of EUROSPEECH*.

Gökhan Tür, Dilek Z. Hakkani-Tür, and Larry P. Heck. 2010. What is left to be understood in atis? In *Proceedings of SLT*, pages 19–24.

Tsung-Hsien Wen, Yishu Miao, Phil Blunsom, and Steve J. Young. 2017a. Latent intention dialogue models. In *Proceedings of ICML*, pages 3732–3741.

Tsung-Hsien Wen, David Vandyke, Nikola Mrkšić, Milica Gašić, Lina M. Rojas-Barahona, Pei-Hao Su, Stefan Ultes, and Steve Young. 2017b. A network-based end-to-end trainable task-oriented dialogue system. In *Proceedings of EACL*, pages 438–449.

Jason Williams. 2012. A critical analysis of two statistical spoken dialog systems in public use. In *Proceedings of SLT*.

Jason D. Williams. 2014. Web-style ranking and SLU combination for dialog state tracking. In *Proceedings of SIGDIAL*, pages 282–291.

Qizhe Xie, Zihang Dai, Eduard H. Hovy, Minh-Thang Luong, and Quoc V. Le. 2019. Unsupervised data augmentation. *CoRR*, abs/1904.12848.

Steve Young. 2010. Still talking to machines (cognitively speaking). In *Proceedings of INTERSPEECH*, pages 1–10.

Steve J. Young. 2002. Talking to machines (statistically speaking). In *Proceedings of INTERSPEECH*.

Tiancheng Zhao, Kaige Xie, and Maxine Eskénazi. 2019. Rethinking action spaces for reinforcement learning in end-to-end dialog agents with latent variable models. In *Proceedings of NAACL-HLT*, pages 1208–1218.

A DSTC8 Datasets: Data Extraction and Statistics

As discussed in §3, we extract span annotated data sets from the Schema Guided Dialog Dataset (SGDD) in four different domains. SGDD is a multi-domain data set with each domain consisting of several sub-domains. As the data set has been built for transfer learning from one domain to another, many sub-domains only exist in either the training or development data sets. We are interested in single-domain dialog, and therefore chose datasets from four different domains of the original dataset: (1) *bus and coach booking*, (2) *buying tickets for events*, (3) *property viewing* and *renting cars*. We select these domains due to their high number of conversations and their large variety of slots (e.g. *area of city to view an apartment*, *type of event to attend*, *time/date of coach to book*). For each of these domains, we chose their first sub-domain[6], and took all turns from conversations that stay within this sub-domain. For the requested slots feature, we check for when the system action of the turn prior contains a `REQUEST` action. The training and development split is kept the same for all extracted turns. Table 5 shows the resulting data set sizes for each sub-domain. We are releasing these filtered single-domain data sets, along with the code to create them from the original SGDD data.

[6] We refer to them by their corresponding ID in the original data set: *Buses_1*, *Events_1*, *Homes_1*, *RentalCars_1*

Sub-domain	Train Size	Dev Size	Slots
Buses_1	1133	377	from_location (169/54), leaving_date (165/57), to_location (166/52)
Events_1	1498	521	city_of_event (253/82), date (151/33), subcategory (56/26)
Homes_1	2064	587	area (288/86), visit_date (237/62)
RentalCars_1	874	328	dropoff_date (112/42), pickup_city (116/48), pickup_date (120/43), pickup_time (119/43)

Table 5: Statistics of the used data sets extracted from the DSTC8 schema-guided dialog dataset. We also report the number of examples in the train and development sets for each slot in parentheses.

B Experimental Results on RESTAURANTS-8K and DSTC8: F_1 Scores for Each Slot

Slot	Fraction	Span-ConveRT	V-CNN-CRF	Span-BERT
date	1	0.96	0.95	0.92
	1/2	0.95	0.94	0.90
	1/4	0.93	0.93	0.86
	1/8	0.91	0.88	0.84
	1/16	0.86	0.82	0.76
	1/32	0.83	0.70	0.62
	1/64	0.76	0.64	0.21
	1/128	0.58	0.43	0.20
first_name	1	0.97	0.93	0.92
	1/2	0.95	0.92	0.92
	1/4	0.93	0.88	0.85
	1/8	0.93	0.85	0.82
	1/16	0.81	0.65	0.53
	1/32	0.54	0.30	0.19
	1/64	0.45	0.23	0.02
	1/128	0.19	0.09	0.00
last_name	1	0.97	0.92	0.93
	1/2	0.96	0.88	0.92
	1/4	0.94	0.83	0.89
	1/8	0.90	0.78	0.72
	1/16	0.80	0.67	0.71
	1/32	0.51	0.45	0.30
	1/64	0.33	0.07	0.01
	1/128	0.24	0.04	0.00
people	1	0.96	0.95	0.91
	1/2	0.94	0.93	0.90
	1/4	0.91	0.92	0.87
	1/8	0.88	0.87	0.80
	1/16	0.83	0.79	0.79
	1/32	0.73	0.63	0.58
	1/64	0.68	0.49	0.43
	1/128	0.60	0.39	0.29
time	1	0.95	0.95	0.91
	1/2	0.93	0.94	0.89
	1/4	0.91	0.91	0.86
	1/8	0.88	0.89	0.82
	1/16	0.76	0.85	0.76
	1/32	0.62	0.76	0.67
	1/64	0.53	0.52	0.46
	1/128	0.43	0.36	0.37

Table 6: F1 scores for each slot in the Restaurants8k datastet.

Dataset	Slot	Fraction	ConveRT Reps	Vanilla Reps	BERT Reps.
Buses_1	from_location	1	0.93	0.94	0.87
		1/2	0.78	0.80	0.75
		1/4	0.82	0.77	0.72
		1/8	0.71	0.67	0.52
		1/16	0.53	0.54	0.35
	leaving_date	1	0.96	0.95	0.96
		1/2	1.00	0.88	0.95
		1/4	0.96	0.88	0.89
		1/8	0.91	0.81	0.72
		1/16	0.79	0.61	0.57
	to_location	1	0.87	0.89	0.84
		1/2	0.82	0.81	0.81
		1/4	0.82	0.65	0.79
		1/8	0.75	0.64	0.61
		1/16	0.49	0.44	0.38
Events_1	city_of_event	1	0.94	0.94	0.90
		1/2	0.92	0.91	0.85
		1/4	0.90	0.80	0.81
		1/8	0.74	0.68	0.51
		1/16	0.80	0.72	0.58
	date	1	0.90	0.88	0.89
		1/2	0.88	0.91	0.91
		1/4	0.84	0.83	0.79
		1/8	0.74	0.62	0.57
		1/16	0.77	0.53	0.68
	subcategory	1	0.90	0.94	0.58
		1/2	0.78	0.71	0.42
		1/4	0.68	0.70	0.50
		1/8	0.46	0.30	0.00
		1/16	0.40	0.31	0.00
Homes_1	area	1	0.97	0.98	0.94
		1/2	0.93	0.90	0.90
		1/4	0.93	0.87	0.86
		1/8	0.87	0.76	0.72
		1/16	0.81	0.64	0.56
	visit_date	1	0.98	0.93	0.99
		1/2	0.98	0.89	0.98
		1/4	0.98	0.88	0.89
		1/8	0.96	0.87	0.88
		1/16	0.95	0.73	0.83
RentalCars_1	dropoff_date	1	0.93	0.89	0.88
		1/2	0.89	0.87	0.72
		1/4	0.73	0.58	0.70
		1/8	0.64	0.71	0.46
		1/16	0.62	0.48	0.33
	pickup_city	1	0.88	0.84	0.86
		1/2	0.86	0.75	0.85
		1/4	0.83	0.65	0.71
		1/8	0.74	0.60	0.49
		1/16	0.53	0.15	0.10
	pickup_date	1	0.86	0.87	0.87
		1/2	0.76	0.74	0.81
		1/4	0.74	0.70	0.72
		1/8	0.71	0.53	0.58
		1/16	0.47	0.26	0.42
	pickup_time	1	0.98	0.95	0.95
		1/2	0.98	0.96	0.91
		1/4	0.95	0.81	0.84
		1/8	0.91	0.50	0.69
		1/16	0.85	0.33	0.68

Table 7: F1 scores for all of the slots in the DSTC8 single-domain experiments

C Quantitative Error Analysis of Span-ConveRT and V-CNN-CRF on RESTAURANTS-8K

We divide the errors into four categories:

1. The model predicted no span when there was a span present.
2. The model predicted a span when no span was present.
3. The model predicted a span which does not overlap the label span.
4. The model predicted a span which overlaps label span.

When training on the full training set (Figure 3), there is little difference in error breakdown between Span-ConveRT and V-CNN-CRF. This suggests the behavior of these models is similar when trained in a high-data setting, but improvements made by Span-ConveRT are on all fronts.

When trained on a 16th of the dataset (Figure 4), the difference between the models becomes more pronounced. Most notably, the Span-ConveRT model produces a greater proportion of type 4 errors compared to the V-CNN-CRF model on every slot. This suggests that the errors Span-ConveRT makes, although not precisely correct with its span prediction, are more likely to yield a span that could parse to a correct value. For example, consider the sentence “a table for 8pm this evening”. The correct span for the slot *time* is ”8pm”, but if a model erroneously predicts “8pm this evening” (a span which overlaps the label span) it will still parse to the same time as the label span.

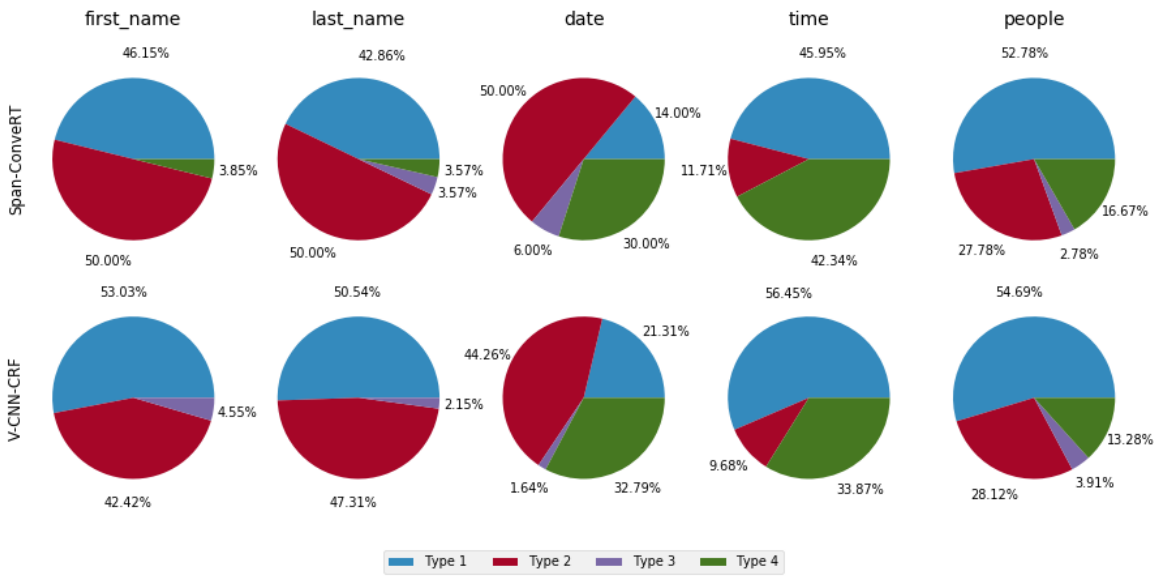

Figure 3: Breakdown of errors made on the test set of RESTAURANTS-8K after training on the entire train set.

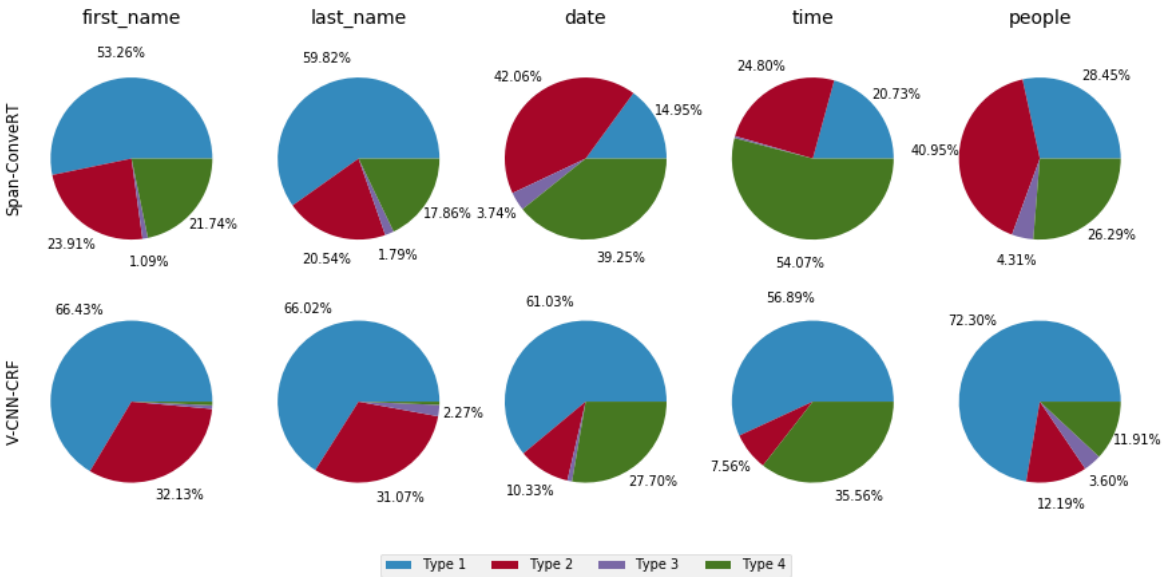

Figure 4: Breakdown of errors made on the test set of RESTAURANTS-8K after training on a 16th of the train set.

D Qualitative Error Analysis of Span-ConveRT and V-CNN-CRF on RESTAURANTS-8K

As an accompaniment to the quantitative results, we provide a brief qualitative analysis of errors in the best performing models. Considering only the *first name* slot, we collect the errors made on the test set that are exclusive to each model. That left 10 errors for Span-ConveRT and 50 for V-CNN-CRF. Along with our analysis based on the full set of 60 errors, we provide a random sample of 5 errors from each model in Tables 8 and 9.

A large portion of the errors exclusively made by V-CNN-CRF were predictions of spans where no name was mentioned. Many words that are not standard to the domain of restaurant booking were, often confidently, wrongly predicted as names. For example, in Table 9 we show that the words "bloody", "web", "animal" and "spread" were all predicted as first names by the baseline model. Employing transferred conversational representations evidently lessens the likelihood of these forms of errors occurring. Also included in the table is an example where the baseline model fails to recognize a name which, when corroborated with similar occurrences in the wider set of errors, suggests that it is less likely to predict spans for out-of-vocabulary names than Span-ConveRT.

As well as backing up the conclusions formed by our numerical results, we were also interested in what ways using pretrained representations might hinder performance. With only 10 errors exclusively made by Span-ConveRT it was not possible to form any sweeping conclusions but a handful of errors suggest that the model might employ its background knowledge to reject unfamiliar first names or accept familiar ones in spite of the sentence structure suggesting otherwise. For example, in the first row of Table 8 we find that the model rejects the name "Wen" despite it being part of a fairly common exchange for this domain and in a natural place for a first name. The other examples demonstrate that the model can sometimes predict last names as first names and in spite of contextual cues suggesting otherwise, can do so over-confidently.

Probability	Text/Spans
N/A	Wen Books, for 7:15PM, I made a reservation yesterday for a party of 8
0.4447	Saul
0.9685	Adragna
0.9247	last name Prader
0.9553	Verjan

Table 8: Random sample of errors exclusively made by **Span-ConveRT** for the slot *first name*. Red text denotes incorrectly predicted spans and orange denotes true spans that were not predicted.

Probability	Text/Spans
0.8872	bloody useless
0.3939	What is their web URL?
0.3319	ok are you guys animal friendly
0.8604	My 7 friends and I can spread ourselves over two tables if necessary
N/A	Gertrudis Hayslett

Table 9: Random sample of errors exclusively made by **V-CNN-CRF** for the slot *first name*. Red text denotes incorrectly predicted spans and orange denotes true spans that were not predicted.