LANGUAGE MODEL IS ALL YOU NEED: NATURAL LANGUAGE UNDERSTANDING AS QUESTION ANSWERING

Mahdi Namazifar, Alexandros Papangelis, Gokhan Tur, Dilek Hakkani-Tür

Amazon Alexa AI

ABSTRACT

Different flavors of transfer learning have shown tremendous impact in advancing research and applications of machine learning. In this work we study the use of a certain family of transfer learning, where the target domain is mapped to the source domain. Specifically we map Natural Language Understanding (NLU) problems to Question Answering (QA) problems and we show that in low data regimes this approach offers significant improvements compared to other approaches to NLU. Moreover, we show that these gains could be increased through sequential transfer learning across NLU problems from different domains. We show that our approach could reduce the amount of required data for the same performance by up to a factor of 10.

Index Terms— Transfer Learning, Question Answering, Natural Language Understanding

1. INTRODUCTION

Transferring the knowledge that machine learning models learn from a source domain to a target domain, which is known as transfer learning (Figure 1a) [1, 2], has shown tremendous success in Natural Language Processing (NLP) [3, 4, 5, 6, 7].

One of the most prominent advantages of transfer learning is manifested in low data regimes. As the models become increasingly complex, in most cases this complexity comes with requirements for larger training data which makes transferring the learning from a high data domain to a low data domain very impactful. In this work we focus on the type of transfer learning in which the target domain is first mapped to the source domain. Next a model is trained on the source domain. Then the transfer of knowledge is done through fine-tuning of this model on the mapping of the target domain (to the source domain), as shown in Figure 1b. As an example of this transfer learning paradigm in NLP, decaNLP [8] could be mentioned where 10 NLP tasks are mapped to the Question Answering (QA) problem, in which given a context the model should find the answer to a question.

In this work, we map Natural Language Understanding (NLU) problems to the QA problem. Here NLU refers to determining the intent and value of slots in an utterance [9]. For instance in "show cheap Italian restaurants" intent could be *inform* and the value for slot *cuisine* is "Italian" and for slot *price range* is "cheap". More specifically in our approach to which we refer as QANLU, we build slot and intent detection questions and answers based on the NLU annotated data. QA models are first trained on QA corpora and then fine-tuned on questions and answers created from NLU annotated data. In this approach transfer learning happens through transferring knowledge of finding the answer to a question given a context, that is acquired by the model during the training of the QA model, to finding the value of an intent or a slot in text input. The main contribution of this work is not the mapping of NLU to QA (as it has been studied in the past), but is the study of transfer learning that comes as a result of this mapping. Through our computational results we show that QANLU in low data regimes and few-shot settings significantly outperforms the sentence classification and token tagging approaches for intent and slot detection tasks, as well as the newly introduced "IC/SF few-shot" approach [10] for NLU. We also show that QANLU sets a new state of the art performance on slot detection on the Restaurants-8k dataset [11]. Furthermore, we show that augmenting the QA corpora with questions and answers created based on NLU annotated data improves the performance of QA models. Throughout this work we use span selection based QA models built on top of transformer-based language models [6]. That being said, our approach is quite generic and could be extended to any type of QA system.

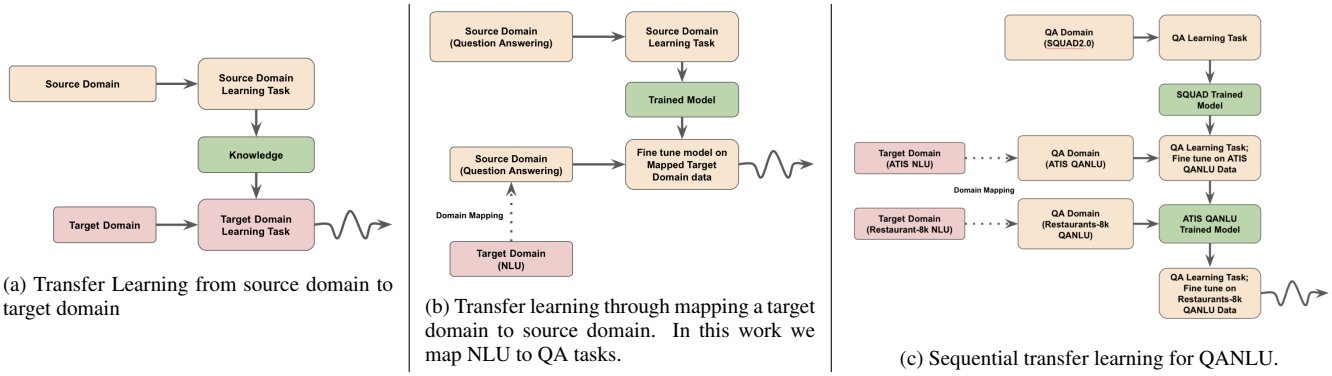
2. RELATED WORKS

Framing NLP tasks as QA has been studied in the past. For instance [8] maps 10 NLP tasks (excluding intent and slot detection) into QA and trains a single model for all of them. However, this work does not explore the task of intent and slot classification. In a similar line of reasoning, [12] poses the Dialogue State Tracking (DST) task as machine reading comprehension (MRC), formulated as QA. [13] builds on that work achieving competitive DST results with full data and in few-shot settings. [14] also explores DST as QA, using candidate values for each slot in the question (similar to the Multiple-Choice setting of [13]) achieving slightly better results than [13]. We propose a method that is conceptually similar but focuses on low-resource applications and does not require designing and training of a new model architecture or extensive data pre-processing, achieving strong results in slot and intent detection with an order of magnitude less data. Here we do not discuss all intent or slot detection methods. However, some notable few-shot NLU works include [15, 16, 17, 11, 15], and we compare against their results when appropriate. Other interesting approaches that do not require training include priming pre-trained language models, e.g. [18].

3. QUESTION ANSWERING FOR NATURAL LANGUAGE UNDERSTANDING (QANLU)

3.1. Slot Detection

Consider a set of text records $T = \{t_1, t_2, ..., t_n\}$ in which each record is annotated for the set of slots $S = s_1, s_2, ..., s_m$. Also for each slot s_j consider a set of questions $Q_{s_j} = \{q_{s_j 1}, q_{s_j 2}, ..., q_{s_j k_j}\}$ that could be asked about s_j given any text record t_i. The following is an example of such a setting:



(a) Transfer Learning from source domain to target domain

(b) Transfer learning through mapping a target domain to source domain. In this work we map NLU to QA tasks.

(c) Sequential transfer learning for QANLU.

Fig. 1:

S : {`cuisine`, `price range`, `area`}, t_i : *"Show cheap Italian restaurants"*
`cuisine`: "Italian", `price range`: "cheap", `area`: ""
Q : $\{Q_{\texttt{cuisine}}, Q_{\texttt{price range}}, Q_{\texttt{area}}\}$

where

$Q_{\texttt{cuisine}}$: {"what cuisine was mentioned?",
"what type of food was specified?"}
$Q_{\texttt{price range}}$: {"what price range?"}
$Q_{\texttt{area}}$: {"what part of town was mentioned?", "what area?"}

Given T, S, and Q it is straightforward to create the set of all the possible questions and their corresponding answers for each t_i as the context for the questions:

Context:	*"Show cheap Italian restaurants"*	
	what cuisine was mentioned?	"Italian"
	what type of food was specified?	"Italian"
	what price range?	"cheap"
	what part of town was mentioned?	""
	what area?	""

We experiment with different ways of creating the set Q. This set could be handcrafted, i.e. for each slot we create a set of questions separately, or created using templates such as "what _____ was mentioned?" where the blank is filled with either the slot name or a short description of the slot, if available.

3.2. Intent Detection

For intent detection we add "yes. no." at the beginning of the context and for each intent we create a question like "is the intent asking about _____?" where the blank is filled with the intent. The answer to these questions are "yes" or "no" from the segment that was added to the beginning of the context depending on whether the intent is in the context or not.

3.3. Question Answering Model

In this work we use span detection based QA models that are built on top of transformers [19] as are described in [6]. We also use the SQuAD2.0 [20] data format for creating questions and answers, as well as the corpus for the source domain (QA). Note that in converting annotated NLU data to questions and answers in QANLU, since for each text record we ask all the questions for all the slots (whether they appear in the text or not), many of the questions are not answerable. As was discussed earlier, we use pre-trained QA models that are trained on SQuAD2.0 (the green box in Figure 1b) and fine-tune them with the questions and answers that are created from the NLU tasks. We also study how in a sequential transfer learning style we can improve the performance of NLU through QANLU (Figure 1c).

4. COMPUTATIONAL RESULTS

In this section we present our computational results for QANLU. Our experiments are done on the ATIS [21, 22] and Restaurants-8k [11] datasets. All of the experiments are implemented using Huggingface [23], and we also use pre-trained language models and QA models provided by Huggingface and fine-tune them for our QA data. We base our experiments mainly on pre-trained DistilBERT [24] and ALBERT [25] models.

4.1. ATIS

The ATIS dataset is an NLU benchmark that provides manual annotations for utterances inquiring a flight booking system. Since the original ATIS dataset does not have a validation set, we use the split of the original training set into training and validation that is proposed in [26]. For each slot in ATIS we create a question set and for each record in ATIS we create the set of questions and answers based on all the question sets and the slot and intent annotation of the record, according to the approach described in Section 3. In the first set of experiments we study how our QANLU approach compares to the widely used joint token and sentence classification [9] in few-shot settings using different stratification in sampling of the training records for the few-shot setting. Table 2 summarizes the results. In this table we report F1 scores for both slots and intent detection tasks. The reason why we use F1 scores for intent detection is that in the ATIS dataset each record could have more than one intent. Each value in Table 2 is an average over 5 runs with different random seeds. Each row in this table represents one sample of the ATIS training data. The set of rows titled "$\mathcal{N}$ uniform samples" are sampled uniformly with samples of sizes of 10, 20, 50, 100, 200, and 500 ATIS records. The set of rows titled "$\mathcal{N}$ samples per slot" are sampled such that each sample includes at least $\mathcal{N}$ instances for any of the slots, where $\mathcal{N}$ is 1, 2, 5, or 10. The set of rows titled "$\mathcal{N}$ samples per intent" are sampled such that each intent appear in at least $\mathcal{N}$ instances, where $\mathcal{N}$ is 1, 2, 5, or 10. The numbers in parenthesis in front of $\mathcal{N}$ represent the number of ATIS records in the sample. For each ATIS record we have 179 questions and answers for intents and slots.

In Table 2 we report performance of models based on both

DistilBERT and ALBERT. For QANLU we fine-tune a QA model trained on SQuAD2.0 data ("distilbert–base–uncased–distilled–squad"[2] for DistilBERT and "twmkn9/albert–base–v2–squad2"[2] for ALBERT) on our questions and answers for ATIS samples. We Also train joint intent and token classification models for the ATIS training samples based on pre-trained DistilBERT and ALBERT models ("distilbert–base–uncased"[2] and "albert–base–v2"[2])[3]. We compare the results of QANLU models with the classification based models (noted as QANLU and Cls in the table, respectively). It is clear that QANLU models outperform classification based models, often by a wide margin. For instance for the ALBERT based model, for the case where there is at least 1 sample per slot the QA based model outperforms the classification based model by 26% (86.37 vs 68.26). It is notable that the gap between the two approaches narrows as the number of samples increases, with the exception of intent detection for the uniform sample with only 10 samples. In a closer look at this sample, the intent for all the records is the same ("atis_flight" which is the intent for 74% of the ATIS training set) and that could explain why the models almost always predict the same value for the intent.

The fact that for both DistilBERT and ALBERT based models we see that the QANLU significantly outperforms the intent and slot classification models in few-shot settings indicates that the performance improvements are likely stemmed from transfer learning from reading comprehension that is learned in the QA task.

In this set of experiments we used handcrafted questions for each slot. One could argue that creating questions for slots is as difficult or perhaps more difficult as getting data annotated specifically for intents and slots. To see if we can detour the manual question creation process we also experimented with questions that were created using frames based on a brief description of each slot as well as using the tokenized slots names. These frame based questions could be easily created for free by running some simple scripts. The experimental results show no significant degradation in the performance of QANLU models trained on frame based questions. In another set of experiments we compare QANLU with another few-shot approach (few-shot IC/SF) proposed in [10]. We use the exact same split of the ATIS dataset that is created in that paper. Results are in Table 1. The few-shot IC/SF results (43.10) are

	Few-shot IC/SF	QANLU
F1 score	43.10	**68.69**

Table 1: QANLU vs Few-shot IC/SF [10] Slot detection F1. 43.10 is reported in Table 5 of [10].

average of multiple runs of a BERT model first pre-trained on the training set, and then fine-tuned on a "support" set sampled from the test set, and then evaluated on a "query" set also sampled from the test set. We used the exact same training set that used in that work and trained a BERT (base size) based QANLU model on the training set. We then directly evaluated that model on the exact same test set created in [10], without any fine-tuning on a support set. The resulting F1 score (68.98) is 60% higher than what is reported [10].

4.2. Restaurants-8k

4.2.1. QANLU for Restaurants-8k

The Restaurants-8k dataset [11] is a set of annotated utterances coming from actual conversations in the restaurant booking domain. The dataset only contains the user side utterances and slot (5 in total) annotations. The system side of the conversations are missing, but given the set of slots that are annotated at every user turn, using simple frames we can build a full context for token classification and QANLU approaches.

The rest of data preparation process is identical to what we described in Section 3.1. We take both uniform and stratified samples of the training data to create few-shot settings for training QANLU models, and compare the results with token classification models. The QANLU model is again a QA model trained on SQuAD2.0 ("distilbert–base–uncased–distilled–squad"[2]) that we fine-tune on the sampled training sets. The token classification model is built on top of "distilbert–base–uncased"[2]. The results are captured in the curves "QANLU (SQ→R8k)" (SQ stands for SQuAD2.0 and R8k stands for Restaurants-8k) and "Cls" (stands for token classification and similar to the ATIS case is based on [9] without the sentence classification head) in Figure 2. We discuss the results in the next subsection.

4.2.2. Sequential Transfer Learning from ATIS to Restaurants-8k

In another set of experiments we study whether QANLU would enable transfer learning from one NLU domain to another. This is referred to as sequential transfer learning in the literature. For this purpose we fine-tune a QANLU model that was trained on the entire ATIS training set, on samples of Restaurants-8k dataset. We compare the performance of the resulting model with QANLU first trained on SQuAD2.0 and then fine-tuned on Restaurants-8k samples, as well as the token classification model.

4.2.3. Restaurants-8k Results

In Figure 2 the curve QANLU (SQ → ATIS → r8k) is the squential transfer learning model based on "distilbert–base–uncased–distilled–squad"[2] model (DistilBERT base model trained on SQuAD2.0). From the figure we can see that except for 10 and 20 uniform samples, for all the samples fine-tuning of SQuAD2.0 QA models on Restaurants-8k results in significantly higher F1 scores compared to the token classification approach. For uniform samples of size 10 and 20 the QANLU model (trained on SQuAD2.0 and fine-tuned on Restaurants-8k samples) performs poorly. Our intuition on the reason behind this poor performance is the small number of questions and answers for these samples (15 per record), and most likely it is not sufficient for the model to learn how to handle NLU style questions. On the other hand for the sequential transfer learning QANLU model (SQ→ATIS→R8k column of Figure 2) we see that the model outperforms both the token classification model and the QANLU model trained on SQuAD2.0 and fine-tuned on Restaurants-8k samples by a wide margin (in some cases by over 50%). These numbers are also shown in Figure 2. This suggests that perhaps using QA as the canonical problem where NLU problems from different domains could be mapped to, could facilitate transfer learning across these NLU problems specially in few-shot settings. Also note that when the entire data is used for training the performance difference vanishes (96.98 for SQ→ R8k, 96.43 for SQ→ ATIS →R8k, and 95.94 for Cls), which suggests that the QANLU approach is as strong as the state of the art outside of few-shot settings. Also Figure 3 shows a comparison between QANLU and Span-ConveRT [11] in few-shot settings. The few-shot F1 scores of Span-ConveRT on Restaurants-8k are borrowed from Table 3 of [11]. In these experiment in order to match the settings of Span-ConverRT we do not create the previous turn for the

[2]Model acquired from `www.huggingface.co/models`

[3]We also tried these models fine-tuned on SQuAD2.0, but they didn't perform as well on the intent and token classification tasks

		Intent				Slot			
		DistilBERT		ALBERT		DistilBERT		ALBERT	
	$\mathcal{N}$	QANLU	Cls	QANLU	Cls	QANLU	Cls	QANLU	Cls
$\mathcal{N}$ uniform samples	10	71.80	71.78	72.18	71.78	**67.23**	61.60	**64.24**	54.78
	20	**83.95**	77.80	**83.28**	75.36	**78.53**	56.70	**74.53**	51.67
	50	**86.07**	78.93	**86.32**	73.90	**83.84**	76.61	**80.26**	74.04
	100	**93.08**	87.91	**92.14**	80.20	**85.69**	80.34	**83.13**	77.50
	200	**94.30**	90.97	**96.78**	85.02	**91.24**	85.32	**89.57**	83.63
	500	**96.40**	95.45	**96.77**	90.62	**92.31**	91.15	**91.18**	86.69
$\mathcal{N}$ samples per slot (Total)	1 (75)	**88.72**	86.47	**90.91**	84.93	**88.47**	76.24	**86.37**	68.26
	2 (136)	**91.68**	84.91	**92.11**	82.42	**90.77**	84.42	**90.17**	79.49
	5 (302)	**94.34**	93.74	**95.52**	87.47	**93.11**	91.38	**87.82**	86.50
	10 (549)	**97.10**	96.19	**94.21**	92.73	**94.11**	93.93	**92.27**	91.68
$\mathcal{N}$ samples per intent (Total)	1 (17)	**40.32**	27.91	**54.49**	25.73	**62.57**	55.38	**62.22**	51.05
	2 (33)	**78.24**	47.20	**62.22**	23.52	**75.39**	65.09	**74.99**	61.01
	5 (81)	**86.49**	74.08	**89.36**	41.28	**84.40**	80.25	**82.70**	71.83
	10 (152)	91.23	91.16	**90.13**	68.93	**88.37**	83.40	**86.32**	78.25
All	N/A (4478)	98.23	98.37	97.59	97.90	95.70	95.80	94.48	95.37

Table 2: QANLU vs. intent and token classification (Cls) [9] for ATIS in few-shot settings. Each row is associated with a different sampling size and strategy of ATIS data. Values in bold represent statistically significant difference at p-value 0.05. Note that QANLU performs significantly better (in some cases by more the 20%) compared to joint intent and slot classification.

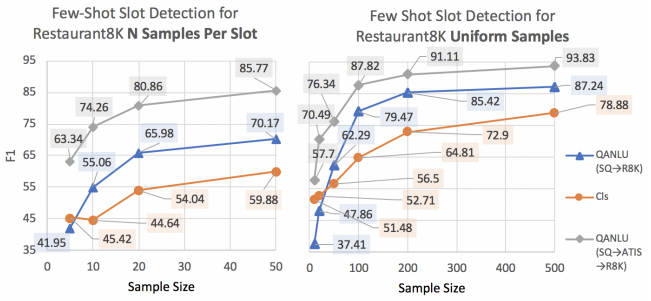


Fig. 2: Slot detection with QANLU vs token classification. SQ→ R8k indicates QANLU first trained on SQuAD2.0 and the fine-tuned on samples of Restaurants-8k. SQ→ATIS→ R8k is QANLU first trained on SQuAD2.0, then fine-tuned on entire ATIS, and then fine-tuned on samples of Restaurants-8k (sequential transfer learning). Cls is for the token classification approach. Numbers associated with each point are F1 scores.

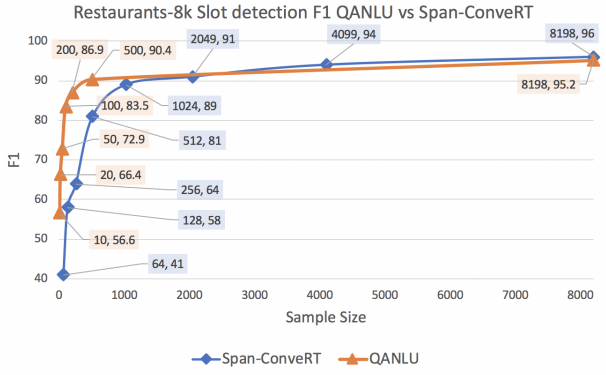


Fig. 3: QANLU compared to Span-ConveRT [11] in few-shot settings. The numbers associated with each point are the sample size and F1, respectively.

context, hence the difference between QANLU numbers in Figure 3 compared to Figure 2. From this figure it is notable that with 20 data points QANLU reaches the higher performance than Span-ConveRT achieves with 256 data points, which translates to a 10x reduction in the amount of data needed. Also with the entire training set QANLU performs within less than 1% of the state-of-the-art.

5. DISCUSSION

The customary feeding token embeddings of a sentence into a network and mapping the output of the network for each token onto a certain number of classes for NLU seems somewhat far from our intuition on how humans understand natural language. The main research question that we try to answer is whether all NLP problems can be efficiently and effectively mapped to one canonical problem. If the answer is yes, could that canonical problem be QA? In this work we scratch the surface on these questions, in that we showcase the strength of transfer learning that happens in this paradigm in learning from few examples for intent and slot detection. But our experiments were limited to span detection QA problem and SQuAD2.0 QA data. Future works will include going beyond this configuration and also expanding across different NLP problems. Measuring how much transfer of knowledge could be achieved across different NLP tasks would be interesting to know. Another future direction could be studying how the questions for QANLU could be generated automatically based on the context.

	SQuAD2.0 (2 epochs)	SQuAD2.0 + ATIS (2 epochs = 9k steps	SQuAD2.0 (9k steps)
"bert-base-cased"	70.07	74.29	65.42
"distilbert-base-uncased"	55.58	60.26	57.03
"albert-base-v2"	78.05	79.26	76.44

Table 3: F1 scores of QA models on original SQuAD2.0 and the augmented SQuAD2.0 with ATIS QANLU Data. Data augmentation improves the performance of QA models. SOTA F1 currently is 93.01

One interesting side product of QANLU is that the questions and answers created for NLU tasks could augment the questions and answers of the QA task (SQuAD2.0 in this work) in order to improve the QA model performance. To study this idea we used the exact training script that Huggingface provides for training QA models on the SQuAD2.0 and also the SQuAD2.0 augmented with questions and answers that we created for ATIS QANLU. The training scripts specify 2 training epochs. It could be argued that this comparison would not be fair since 2 passes over the augmented data means a lot more optimization steps since there are many more questions and answers in the augmented data. To account for this we also run the training on the original SQuAD2.0 data for the same number of optimization steps as it takes to run 2 epochs on the augmented data (9000 steps). The results (QA F1 on the validation set) are shown in Table 3. As the numbers show training the same models on the augmented data significantly improves the performance of the final QA model on the Development set of SQuAD2.0. We believe this result could be an indication that we can not only transfer from QA to other NLU tasks, we can also improve QA through data augmentation by mapping NLU problems to QA.

6. REFERENCES

[1] Chuanqi Tan, Fuchun Sun, Tao Kong, Wenchang Zhang, Chao Yang, and Chunfang Liu, "A survey on deep transfer learning," in *Artificial Neural Networks and Machine Learning – ICANN 2018*, Věra Kůrková, Yannis Manolopoulos, Barbara Hammer, Lazaros Iliadis, and Ilias Maglogiannis, Eds., 2018, pp. 270–279.

[2] Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He, "A comprehensive survey on transfer learning," 2020.

[3] Zaid Alyafeai, Maged S. Al-shaibani, and I. Ahmad, "A survey on transfer learning in natural language processing," *ArXiv*, vol. abs/2007.04239, 2020.

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

[5] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in Neural Information Processing Systems 26*, C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, Eds., pp. 3111–3119. 2013.

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1*, June 2019, pp. 4171–4186.

[7] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, W. Li, and P. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *ArXiv*, vol. abs/1910.10683, 2019.

[8] Bryan McCann, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher, "The natural language decathlon: Multitask learning as question answering," *CoRR*, vol. 1806.08730, 2018.

[9] Qian Chen, Zhu Zhuo, and Wen Wang, "BERT for joint intent classification and slot filling," *CoRR*, vol. 1902.10909, 2019.

[10] Jason Krone, Yi Zhang, and Mona Diab, "Learning to classify intents and slot labels given a handful of examples," in *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, 2020, pp. 96–108.

[11] Sam Coope, Tyler Farghly, Daniela Gerz, Ivan Vulić, and Matthew Henderson, "Span-convert: Few-shot span extraction for dialog with pretrained conversational representations," 2020.

[12] Shuyang Gao, Abhishek Sethi, Sanchit Agarwal, Tagyoung Chung, and Dilek Hakkani-Tur, "Dialog state tracking: A neural reading comprehension approach," *arXiv preprint arXiv:1908.01946*, 2019.

[13] Shuyang Gao, Sanchit Agarwal, Tagyoung Chung, Di Jin, and Dilek Hakkani-Tur, "From machine reading comprehension to dialogue state tracking: Bridging the gap," *arXiv preprint arXiv:2004.05827*, 2020.

[14] Li Zhou and Kevin Small, "Multi-domain dialogue state tracking as dynamic knowledge graph enhanced question answering," *arXiv preprint arXiv:1911.06192*, 2019.

[15] Ankur Bapna, Gokhan Tur, Dilek Hakkani-Tur, and Larry Heck, "Towards zero-shot frame semantic parsing for domain scaling," *arXiv preprint arXiv:1707.02363*, 2017.

[16] Hemanthage S Bhathiya and Uthayasanker Thayasivam, "Meta learning for few-shot joint intent detection and slot-filling," in *Proceedings of the 2020 5th International Conference on Machine Learning Technologies*, 2020, pp. 86–92.

[17] Darsh J Shah, Raghav Gupta, Amir A Fayazi, and Dilek Hakkani-Tur, "Robust zero-shot cross-domain slot filling with example values," 2019.

[18] Andrea Madotto, "Language models as few-shot learner for task-oriented dialogue systems," *arXiv preprint arXiv:2008.06239*, 2020.

[19] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, L ukasz Kaiser, and Illia Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., pp. 5998–6008. 2017.

[20] Pranav Rajpurkar, Robin Jia, and Percy Liang, "Know what you don't know: Unanswerable questions for SQuAD," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pp. 784–789.

[21] Charles T. Hemphill, John J. Godfrey, and George R. Doddington, "The ATIS spoken language systems pilot corpus," in *Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania, June 24-27,1990*, 1990.

[22] G. Tur, D. Hakkani-Tür, and L. Heck, "What is left to be understood in atis?," in *2010 IEEE Spoken Language Technology Workshop*, 2010, pp. 19–24.

[23] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush, "Huggingface's transformers: State-of-the-art natural language processing," *ArXiv*, vol. abs/1910.03771, 2019.

[24] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf, "Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter," 2020.

[25] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut, "Albert: A lite bert for self-supervised learning of language representations," *ArXiv*, vol. abs/1909.11942, 2020.

[26] Chenwei Zhang, Yaliang Li, Nan Du, Wei Fan, and Philip S Yu, "Joint slot filling and intent detection via capsule neural networks," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2019.