DIET: Lightweight Language Understanding for Dialogue Systems

Tanja Bunk[1][*]   **Daksh Varshneya**[1][†]   **Vladimir Vlasov**[1][‡]   **Alan Nichol**[§]

Rasa

Abstract

Large-scale pre-trained language models have shown impressive results on language understanding benchmarks like GLUE and Super-GLUE, improving considerably over other pre-training methods like distributed representations (GloVe) and purely supervised approaches. We introduce the Dual Intent and Entity Transformer (DIET) architecture, and study the effectiveness of different pre-trained representations on intent and entity prediction, two common dialogue language understanding tasks. DIET advances the state of the art on a complex multi-domain NLU dataset and achieves similarly high performance on other simpler datasets. Surprisingly, we show that there is no clear benefit to using large pre-trained models for this task, and in fact DIET improves upon the current state of the art even in a purely supervised setup without any pre-trained embeddings. Our best performing model outperforms fine-tuning BERT and is about six times faster to train.

1 Introduction

Two common approaches to data-driven dialogue modeling are the end-to-end and the modular systems. Modular approaches like POMDP-based dialogue policies (Williams and Young, 2007) and Hybrid Code Networks (Williams et al., 2017) use separate natural language understanding (NLU) and generation (NLG) systems. The dialogue policy itself receives the output from the NLU system and chooses the next system action, before the NLG system generates a corresponding response. In the end-to-end approach user input is directly fed into

the dialogue policy to predict the next system utterance. Recently these two approaches have been combined in Fusion Networks (Mehri et al., 2019).

In the context of dialogue systems, natural language understanding typically refers to two subtasks: intent classification and entity recognition.

Goo et al. argue that modeling these sub-tasks separately can suffer from error propagation and hence a single multi-task architecture should benefit from mutual enhancement between two tasks.

Recent work has shown that large pre-trained language models yield the best performance on challenging language understanding benchmarks (see section 2). However, the computational cost of both pre-training and fine-tuning such models is considerable (Strubell et al., 2019).

Dialogue systems are not only developed by researchers, but by many thousands of software developers worldwide. Facebook's Messenger platform alone supports hundreds of thousands of third party conversational assistants (Johnson, 2018). For these applications it is desirable that models can be trained and iterated upon quickly to fit into a typical software development workflow. Furthermore, since many of these assistants operate in languages other than English, it is important to understand what performance can be achieved *without* large-scale pre-training.

In this paper, we propose DIET (Dual Intent and Entity Transformer), a new multi-task architecture for intent classification and entity recognition. One key feature is the ability to incorporate pre-trained word embeddings from language models and combine these with sparse word and character level n-gram features in a plug-and-play fashion. Our experiments demonstrate that even without pre-trained embeddings, using only sparse word and character level n-gram features, DIET improves upon the current state of the art on a complex NLU dataset. Moreover, adding pre-

---

[*]t.bunk@rasa.com

[†]d.varshneya@rasa.com

[‡]vladimir@rasa.com

[§]alan@rasa.com

[1]authors have equally contributed

trained word and sentence embeddings from language models further improves the overall accuracy on all tasks. Our best performing model significantly outperforms fine-tuning BERT and is six times faster to train. Documented code to reproduce these experiments is available online at `https://github.com/RasaHQ/DIET-paper`.

2 Related Work

2.1 Transfer learning of dense representations

Top performing models (Liu et al., 2019a; Zhang et al., 2019) on language understanding benchmarks like GLUE (Wang et al., 2018) and SuperGLUE (Wang et al., 2019) benefit from using dense representations of words and sentences from large pre-trained language models like ELMo (Peters et al., 2018), BERT (Devlin et al., 2018), GPT (Radford, 2018) etc. Since these embeddings are trained on large scale natural language text corpora, they generalize well across tasks and can be transferred as input features to other language understanding tasks with or without fine-tuning (Peters et al., 2018; Sun et al., 2019a; Lee and Hsiang, 2019; Adhikari et al., 2019; Klein and Nabi, 2019). Different fine-tuning strategies have also been proposed for effective transfer learning across tasks (Howard and Ruder, 2018; Sun et al., 2019b). However, Peters et al. (2019) show that fine-tuning a large pre-trained language model like BERT may not be optimal for every downstream task. Moreover, these large scale language models are slow, expensive to train and hence not ideal for real-world conversational AI applications (Henderson et al., 2019b). To achieve a more compact model, Henderson et al. (2019b) pre-train a word and sentence level encoder on a large scale conversational corpus from Reddit (Henderson et al., 2019a). The resultant sentence level dense representations, when transferred (without fine-tuning) to a downstream task of intent classification, perform much better than embeddings from BERT and ELMo. We further investigate this behaviour for the task of joint intent classification and entity recognition. We also study the impact of using sparse representations like word level one-hot encodings and character level n-grams along with dense representations transferred from large pre-trained language models.

2.2 Joint Intent Classification and Named Entity Recognition

In recent years a number of approaches have been studied for training intent classification and named entity recognition (NER) in a multi-task setup. Zhang and Wang (2016) proposed a joint architecture composed of a Bidirectional Gated Recurrent Unit (BiGRU). The hidden state of each time step is used for entity tagging and the hidden state of last time step is used for intent classification. Liu and Lane (2016); Varghese et al. (2020) and Goo et al. (2018) propose an attention-based Bidirectional Long Short Term Memory (BiLSTM) for joint intent classification and NER. Haihong et al. (2019) introduce a co-attention network on top of individual intent and entity attention units for mutual information sharing between each task. Chen et al. (2019) propose Joint BERT which is built on top of BERT and is trained in an end to end fashion. They use the hidden state of the first special token `[CLS]` for intent classification. The entity labels are predicted using the final hidden states of other tokens. A hierarchical bottom-up architecture was proposed by Vanzo et al. (2019) composed of BiLSTM units to capture shallower representations of semantic frames (Baker et al., 1998). They predict dialogue acts, intents and entity labels from representations learnt by individual layers stacked in a bottom-up fashion. In this work, we adopt a similar transformer-based multi-task setup for DIET and also perform an ablation study to observe its effectiveness compared to a single task setup.

3 DIET Architecture

A schematic representation of our architecture is illustrated in Figure 1. DIET consists of several key parts.

Featurization Input sentences are treated as a sequence of tokens, which can be either words or subwords depending on the featurization pipeline. Following Devlin et al. (2018), we add a special classification token __CLS__ to the end of each sentence. Each input token is featurized with what we call sparse features and/or dense features. Sparse features are token level one-hot encodings and multi-hot encodings of character n-grams ($n \leq 5$). Character n-grams contain a lot of redundant information, so to avoid overfitting we apply dropout to these sparse features. Dense features can be any pre-trained word embeddings: ConveRT (Hen-
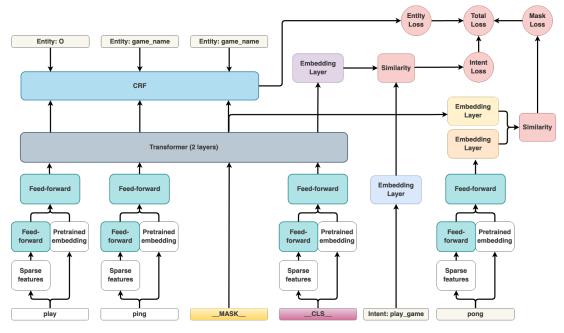
Figure 1: A schematic representation of the DIET architecture. The phrase "play ping pong" has the intent `play_game` and entity `game_name` with value "ping pong". Weights of the feed-forward layers are shared across tokens.

derson et al., 2019b), BERT (Devlin et al., 2018) or GloVe (Pennington et al., 2014). Since ConveRT is also trained as a sentence encoder, when using ConveRT we set the initial embedding for `__CLS__` token as the sentence encoding of the input sentence obtained from ConveRT.[1] This adds extra contextual information for the complete sentence in addition to information from individual word embeddings. For out-of-the-box pre-trained BERT, we set it to the corresponding output embedding of the BERT `[CLS]` token and for GloVe, to the mean of the embeddings of the tokens in a sentence. Sparse features are passed through a fully connected layer with shared weights across all sequence steps to match the dimension of the dense features. The output of the fully connected layer is concatenated with the dense features from pre-trained models.

Transformer To encode context across the complete sentence, we use a 2 layer transformer (Vaswani et al., 2017) with relative position attention (Shaw et al., 2018). The transformer architecture requires its input to be the same dimension as the transformer layers. Therefore, the

concatenated features are passed through another fully connected layer with shared weights across all sequence steps to match the dimension of the transformer layers, which in our experiments is 256.

Named entity recognition A sequence of entity labels $\boldsymbol{y}_{\text{entity}}$ is predicted through a Conditional Random Field (CRF) (Lafferty et al., 2001) tagging layer on top of the transformer output sequence $\boldsymbol{a}$ corresponding to an input sequence of tokens.

$$L_{\text{E}} = L_{\text{CRF}}(\boldsymbol{a}, \boldsymbol{y}_{\text{entity}}), \qquad (1)$$

where $L_{\text{CRF}}(.)$ denotes negative log-likelihood for a CRF (Lample et al., 2016).

Intent classification The transformer output for `__CLS__` token a_{CLS} and intent labels y_{intent} are embedded into a single semantic vector space $h_{\text{CLS}} = E(a_{\text{CLS}})$, $h_{\text{intent}} = E(y_{\text{intent}})$, where $h \in \mathbb{R}^{20}$. We use the dot-product loss (Wu et al., 2017; Henderson et al., 2019c; Vlasov et al., 2019) to maximize the similarity $S_{\text{I}}^{+} = h_{\text{CLS}}^{T} h_{\text{intent}}^{+}$ with the target label y_{intent}^{+} and minimize similarities $S_{\text{I}}^{-} = h_{\text{CLS}}^{T} h_{\text{intent}}^{-}$ with negative samples y_{intent}^{-}.

$$L_{\text{I}} = -\left\langle S_{\text{I}}^{+} - \log\left(e^{S_{\text{I}}^{+}} + \sum_{\Omega_{\text{I}}^{-}} e^{S_{\text{I}}^{-}}\right)\right\rangle, \quad (2)$$

where the sum is taken over the set of negative samples Ω_{I}^{-} and the average $\langle.\rangle$ is taken over all

---

[1]Sentence embeddings from ConveRT are 1024-dimensional and word embeddings are 512-dimensional. To overcome this dimension mismatch, we use a simple trick of tiling the word embeddings to extra 512 dimensions and get 1024-dimensional word embeddings. This keeps the neural architecture the same for different pre-trained embeddings.

examples.

At inference time, the dot-product similarity serves as a ranker over all possible intent labels.

Masking Inspired by the masked language modelling task (Taylor, 1953; Devlin et al., 2018), we add an additional training objective to predict randomly masked input tokens. We select at random 15% of the input tokens in a sequence. For a selected token, in 70% of cases we substitute the input with the vector corresponding to the special mask token __MASK__, in 10% of cases we substitute the input with the vector corresponding to a random token and in the remaining 20% we keep the original input. The output of the transformer a_{MASK} for each selected token y_{token} is fed through a dot-product loss (Wu et al., 2017; Henderson et al., 2019c; Vlasov et al., 2019) similar to the intent loss.

$$L_{\text{M}} = -\left\langle S_{\text{M}}^{+} - \log\left(e^{S_{\text{M}}^{+}} + \sum_{\Omega_{\text{M}}^{-}} e^{S_{\text{M}}^{-}}\right)\right\rangle, \quad (3)$$

where $S_{\text{M}}^{+} = h_{\text{MASK}}^{T} h_{\text{token}}^{+}$ is the similarity with the target label y_{token}^{+} and $S_{\text{M}}^{-} = h_{\text{MASK}}^{T} h_{\text{token}}^{-}$ are the similarities with negative samples y_{token}^{-}, $h_{\text{MASK}} = E(a_{\text{MASK}})$ and $h_{\text{token}} = E(y_{\text{token}})$ are the corresponding embedding vectors $h \in \mathbb{R}^{20}$; the sum is taken over the set of negative samples Ω_{M}^{-} and the average $\langle . \rangle$ is taken over all examples.

We hypothesize that adding a training objective for reconstructing masked input should act as a regularizer as well as help the model learn more general features from text and not only discriminative features for classification (Yoshihashi et al., 2018).

Total loss We train the model in multi-task fashion by minimizing the total loss L_{total}.

$$L_{total} = L_I + L_E + L_M \quad (4)$$

The architecture can be configured to turn off any of the losses in the sum above.

Batching We use a balanced batching strategy (Vlasov et al., 2019) to mitigate class imbalance (Japkowicz and Stephen, 2002) as some intents can be more frequent than others. We also increase our batch size throughout training as another source of regularization (Smith et al., 2017).

4 Experimental Evaluation

In this section we first describe the datasets used in our experiments, then we describe the experimental setup, followed by an ablation study to understand the effectiveness of each component of the architecture.

4.1 Datasets

We used three datasets for our evaluation: NLU-Benchmark, ATIS, and SNIPS. The focus of our experiments is the NLU-Benchmark dataset, since it is the most challenging of the three. The state of the art on ATIS and SNIPS is already close to 100% test set accuracy, see Table 5.

NLU-Benchmark dataset The NLU-Benchmark dataset (Liu et al., 2019b), available online[2], is annotated with scenarios, actions, and entities. For example, "schedule a call with Lisa on Monday morning" is annotated with the scenario calendar, the action set_event, and the entities [event_name: *a call with Lisa*] and [date: *Monday morning*]. The intent label is obtained by concatenating the scenario and action labels (e.g. calendar_set_event). The dataset has 25,716 utterances which cover multiple home assistant tasks, such as playing music or calendar queries, chit-chat, and commands issued to a robot. The data is split into 10 folds. Each fold has its own train and test set of respectively 9960 and 1076 utterances.[3] Overall 64 intents and 54 entity types are present.

ATIS ATIS (Hemphill et al., 1990) is a well-studied dataset in the field of NLU. It comprises annotated transcripts of audio recordings of people making flight reservations. We used the same data split as Chen et al. (2019), originally proposed by Goo et al. (2018) and available online[4]. The training, development, and test sets contain 4,478, 500 and 893 utterances. The training dataset has 21 intents and 79 entities.

SNIPS This dataset is collected from the Snips personal voice assistant (Coucke et al., 2018). It contains 13,784 training and 700 test examples. For fair comparison, we used the same data split as Chen et al. (2019) and Goo et al. (2018). 700

		Intent	**Entities**
HERMIT	F1	87.55±0.63	84.74±1.18
	R	87.70±0.64	82.04±2.12
	P	87.41±0.63	**87.65±0.98**
sparse + ConveRT†	F1	**90.18±0.53**	**86.04±1.01**
	R	**90.18±0.53**	**86.13±0.99**
	P	**90.18±0.53**	85.95±1.42

Table 1: Results from HERMIT (Vanzo et al., 2019) and from our best performing configuration of DIET on the NLU-Benchmark dataset. Our best performing model uses word and character level sparse features and combines them with embeddings from ConveRT. The model does not use a mask loss (indicated by the †).

		Intent	**Entities**
single-task: intent classification	F1	90.90±0.19	-
	R	90.90±0.19	-
	P	90.90±0.19	-
single-task: entity recognition	F1	-	82.57±1.41
	R	-	81.85±1.87
	P	-	83.32±1.51

Table 2: Training DIET on just a single task, i.e. intent classification or entity recognition, on the NLU-Benchmark dataset.

examples from the training set are used as development set. The data can be found online[4]. The SNIPS dataset contains 7 intents and 39 entities.

4.2 Experimental Setup

Our model is implemented in Tensorflow (Abadi et al., 2016). We used the first fold of the NLU-Benchmark dataset to select hyperparameters. We randomly took 250 utterances from the training set as a development set for that purpose. We trained our models over 200 epochs on a machine with 4 CPUs, 15 GB of memory and one NVIDIA Tesla K80. We used Adam (Kingma and Ba, 2014) for optimization with an initial learning rate of 0.001. The batch size increased incrementally from 64 to 128 (Smith et al., 2017). Training our model on the first fold of the NLU-Benchmark dataset takes around one hour. At inference time we need around 80ms to process one utterance.

4.3 Experiments on NLU-Benchmark dataset

The NLU-Benchmark dataset contains 10 folds, each with a separate train and test set. To obtain the overall performance of our model on this dataset we followed the approach of Vanzo et al. (2019): train 10 models independently, one for each fold and take the average as the final score. Micro-averaged precision, recall and F1 score are used as metrics. True positives, false positives, and false negatives for intent labels are calculated as in any other multi-class classification task. An entity counts as true positive if there is an overlap between the predicted and the gold span and their labels match.

Table 1 shows the results of our best performing model on the NLU-Benchmark dataset. Our best performing model uses sparse features, i.e. one-hot encodings at the token level and multi-hot encod-

ings of character n-grams ($n \leq 5$). These sparse features are combined with dense embeddings from ConveRT (Henderson et al., 2019b). Our best performing model does not use a mask loss (described in Section 3 and indicated by † in the table). We outperform HERMIT on intents by over 2% absolute. Our micro-averaged F1 score on entities (86.04%) is also higher than HERMIT (84.74%). HERMIT reports a similar precision value on entities, however, our recall value is much higher (86.13% compared to 82.04%).

4.4 Ablation Study on NLU-Benchmark dataset

We used the NLU-Benchmark dataset to evaluate different components of our model architecture as it covers multiple domains and has the most number of intents and entities of the three datasets.

Importance of joint training In order to evaluate if the two tasks, i.e. intent classification and named entity recognition, benefit from being optimized jointly or not, we trained models for each of the tasks individually. Table 2 lists the results of just training a single task with DIET. The results show that the performance of intent classification slightly decreases when trained jointly with entity recognition (90.90% vs 90.18%). It should be noted that the best performing configuration for single task training for intent classification corresponds to using embeddings from ConveRT with no transformer layers[5]. However, the micro-averaged F1 score of entities drops from 86.04% to 82.57% when entities are trained separately. Inspecting the NLU-Benchmark dataset, this is likely due to strong correlation between particular intents and the presence of specific entities. For example, almost every utterance that

---

[5]This result is in line with the results reported in Casanueva et al. (2020)

belongs to the `play_game` intent has an entity called `game_name`. Also, the entity `game_name` only occurs together with the intent `play_game`. We believe that this result further brings out the importance of having a modular and configurable architecture like DIET in order to handle trade-off in performance across both tasks.

Importance of different featurization components and masking As described in Section 3 embeddings from different pre-trained language models can be used as dense features. We trained multiple variants to study the effectiveness of each: only sparse features, i.e. one-hot encodings at the token level and multi-hot encodings of character n-grams ($n \leq 5$), and combinations of those together with ConveRT, BERT, or GloVe. Additionally, we trained each combination with and without the mask loss. The results presented in Table 3 show F1 scores for both intent classification and entity recognition and indicate multiple observations: DIET performance is competitive when using sparse features together with the mask loss, without any pre-trained embeddings. Adding a mask loss improves performance by around 1% absolute on both intents and entities. DIET with GloVe embeddings is also equally competitive and is further enhanced on both intents and entities when used in combination with sparse features and mask loss. Interestingly, using contextual BERT embeddings as dense features performs worse than GloVe. We hypothesize that this is because BERT is pre-trained primarily on prose and hence requires fine-tuning before being transferred to a dialogue task. The performance of DIET with ConveRT embeddings supports this, since ConveRT was trained specifically on conversational data. ConveRT embeddings with the addition of sparse features achieves the best F1 score on intent classification and it outperforms the state of the art on both intent classification and entity recognition by a considerable margin of around 3% absolute. Adding a mask loss seems to slightly hurt the performance when used with BERT and ConveRT as dense features.

Comparison with fine-tuned BERT Following Peters et al. (2019), we evaluate the effectiveness of incorporating BERT inside the featurization pipeline of DIET and fine-tuning the entire model. Table 4 shows DIET with frozen ConveRT embeddings as dense features and word, char level sparse features outperforms fine-tuned BERT on en-

tity recognition while performing on par for intent classification. This result is especially important because fine-tuning BERT inside DIET on all 10 folds of NLU-Benchmark dataset takes 60 hours, compared to 10 hours for DIET with embeddings from ConveRT and sparse features.

4.5 Experiments on ATIS and SNIPS

In order to compare our results to the results presented in Chen et al. (2019), we use the same evaluation method as Chen et al. (2019) and Goo et al. (2018). They report the accuracy for intent classification and micro-averaged F1 score for entity recognition. Again, true positives, false positives, and false negatives for intent labels are obtained as in any other multi-class classification task. However, an entity only counts as a true positive if the prediction span exactly matches the gold span and their label match, a stricter definition than that of Vanzo et al. (2019). All experiments on ATIS and SNIPS were run 5 times. We take the average over the results from those runs as final numbers.

To understand how transferable the hyperparameters of DIET are, we took the best performing model configurations of DIET on the NLU-Benchmark dataset and evaluated them on ATIS and SNIPS. The intent classification accuracy and named entity recognition F1 score on the ATIS and SNIPS dataset are listed in Table 5.

Due to the stricter evaluation method we tagged our data using the BILOU tagging schema (Ramshaw and Marcus, 1995). The use of the BILOU tagging schmea is indicated by the * in Table 5.

Remarkably, using only sparse features and no pre-trained embeddings whatsoever, DIET achieves performance within 1-2% of the Joint BERT model. Using the hyperparameters from the best performing model on the NLU-Benchmark dataset, DIET achieves results competitive with Joint BERT on both ATIS and SNIPS.

5 Conclusion

We introduced DIET, a flexible architecture for intent and entity modeling. We studied its performance on multiple datasets, and showed that DIET advances the state of the art on the challenging NLU-Benchmark dataset. Furthermore we extensively study the effectiveness of using embeddings from various pre-training methods. We find that there is no single set of embeddings which is al-

sparse	dense	mask loss	Intent	Entities
✓	✗	✗	87.10±0.75	83.88±0.98
✓	✗	✓	88.19±0.84	85.12±0.85
✗	GloVe	✗	89.20±0.90	84.34±1.03
✓	GloVe	✗	89.38±0.71	84.89±0.91
✗	GloVe	✓	88.78±0.70	85.06±0.84
✓	GloVe	✓	89.13±0.77	86.04±1.09
✗	BERT	✗	87.44±0.92	84.20±0.91
✓	BERT	✗	88.46±0.88	85.26±1.01
✗	BERT	✓	86.92±1.09	83.96±1.33
✓	BERT	✓	87.45±0.67	84.64±1.31
✗	ConveRT	✗	89.76±0.98	**86.06±1.38**
✓	ConveRT	✗	**90.18±0.53**	86.04±1.01
✗	ConveRT	✓	90.15±0.68	85.76±0.80
✓	ConveRT	✓	89.47±0.74	86.04±1.29

Table 3: Comparison of different featurization and architecture components on NLU-Benchmark dataset. The three columns on the left indicate whether sparse features are used or not, what kind of dense features are used, if any, and whether the model was trained with a mask loss or not. The reported numbers are micro-averaged F1 scores.

		Intent	Entities
Fine-tuned BERT	F1	89.67±0.48	85.73±0.91
	R	89.67±0.48	84.71±1.28
	P	89.67±0.48	**86.78±1.02**
sparse + ConveRT[†]	F1	**90.18±0.53**	**86.04±1.01**
	R	**90.18±0.53**	**86.13±0.99**
	P	**90.18±0.53**	85.95±1.42

Table 4: Comparison of best performing feature set for DIET against fine-tunable BERT inside DIET on the NLU-Benchmark dataset. The best performing feature set for DIET contains sparse features combined with embeddings from ConveRT (not fined-tuned) without a mask loss (indicated by the †). Fine-tuning BERT with DIET takes 60 hours as compared to just 10 hours for DIET with sparse and ConveRT features.

ways best across different datasets, highlighting the importance of a modular architecture. Furthermore we show that word embeddings from distributional models like GloVe are competitive with embeddings from large-scale language models, and that in fact without using any pre-trained embeddings, DIET can still achieve competitive performance, outperforming state of the art on NLU-Benchmark. Finally, we also show that the best set of pre-trained embeddings for DIET on NLU-Benchmark outperforms fine-tuning BERT inside DIET and is six times faster to train.

References

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Gregory S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian J. Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Józefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Gordon Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul A. Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda B. Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2016. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *CoRR*, abs/1603.04467.

Ashutosh Adhikari, Achyudh Ram, Raphael Tang, and Jimmy Lin. 2019. Docbert: BERT for document classification. *CoRR*, abs/1904.08398.

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*, COLING '98, pages 86–90, Stroudsburg, PA, USA. Association for Computational Linguistics.

Iigo Casanueva, Tadas Teminas, Daniela Gerz, Matthew Henderson, and Ivan Vuli. 2020. Efficient intent detection with dual sentence encoders.

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

	ATIS		SNIPS	
	Intent	Entities	Intent	Entities
Joint BERT	**97.90**	**96.10**	**98.60**	**97.00**
sparse + ConveRT[†*]	96.59	95.08	98.03	94.79
sparse + GloVe[*]	96.31	94.99	97.50	94.84
sparse[*]	96.61	95.37	97.71	95.10

Table 5: Results of Joint BERT (Chen et al., 2019) and different feature sets for DIET on the ATIS and SNIPS datasets. Reported numbers are accuracy for intents and micro-average F1 score for entities. The ∗ indicates that the data was annotated using the BILOU tagging schema. † implies that no mask loss was used.

Doumouro, Thibault Gisselbrecht, Francesco Caltagirone, Thibaut Lavril, Maël Primet, and Joseph Dureau. 2018. Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces. *CoRR*, abs/1805.10190.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

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, New Orleans, Louisiana. Association for Computational Linguistics.

Ee Haihong, Peiqing Niu, Zhongfu Chen, and Meina Song. 2019. A novel bi-directional interrelated model for joint intent detection and slot filling. *CoRR*, abs/1907.00390.

Charles T. Hemphill, John J. Godfrey, and George R. Doddington. 1990. 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*.

Matthew Henderson, Pawel Budzianowski, Iñigo Casanueva, Sam Coope, Daniela Gerz, Girish Kumar, Nikola Mrksic, Georgios Spithourakis, Pei-Hao Su, Ivan Vulic, and Tsung-Hsien Wen. 2019a. A repository of conversational datasets. *CoRR*, abs/1904.06472.

Matthew Henderson, Iñigo Casanueva, Nikola Mrkšić, Pei-Hao Su, Ivan Vulić, et al. 2019b. Convert: Efficient and accurate conversational representations from transformers. *arXiv preprint arXiv: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. 2019c. Training neural response selection for task-oriented dialogue systems. *arXiv preprint arXiv:1906.01543*.

Jeremy Howard and Sebastian Ruder. 2018. Fine-tuned language models for text classification. *CoRR*, abs/1801.06146.

Nathalie Japkowicz and Shaju Stephen. 2002. The class imbalance problem: A systematic study. *Intelligent data analysis*, 6(5):429–449.

Khari Johnson. 2018. Facebook messenger passes 300,000 bots. venturebeat.com [Online; posted 1-May-2018].

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. Cite arxiv:1412.6980Comment: Published as a conference paper at the 3rd International Conference for Learning Representations, San Diego, 2015.

Tassilo Klein and Moin Nabi. 2019. Attention is (not) all you need for commonsense reasoning. *CoRR*, abs/1905.13497.

John Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. *CoRR*, abs/1603.01360.

Jieh-Sheng Lee and Jieh Hsiang. 2019. Patentbert: Patent classification with fine-tuning a pre-trained BERT model. *CoRR*, abs/1906.02124.

Bing Liu and Ian Lane. 2016. Attention-based recurrent neural network models for joint intent detection and slot filling. *CoRR*, abs/1609.01454.

Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2019a. Improving multi-task deep neural networks via knowledge distillation for natural language understanding. *CoRR*, abs/1904.09482.

Xingkun Liu, Arash Eshghi, Pawel Swietojanski, and Verena Rieser. 2019b. Benchmarking natural language understanding services for building conversational agents. *CoRR*, abs/1903.05566.

Shikib Mehri, Tejas Srinivasan, and Maxine Eskenazi. 2019. Structured fusion networks for dialog. *arXiv preprint arXiv:1907.10016*.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *CoRR*, abs/1802.05365.

Matthew E. Peters, Sebastian Ruder, and Noah A. Smith. 2019. To tune or not to tune? adapting pretrained representations to diverse tasks. *CoRR*, abs/1903.05987.

Alec Radford. 2018. Improving language understanding by generative pre-training.

Lance A. Ramshaw and Mitchell P. Marcus. 1995. Text chunking using transformation-based learning. *CoRR*, cmp-lg/9505040.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. 2018. Self-attention with relative position representations. *arXiv preprint arXiv:1803.02155*.

Samuel L Smith, Pieter-Jan Kindermans, Chris Ying, and Quoc V Le. 2017. Don't decay the learning rate, increase the batch size. *arXiv preprint arXiv:1711.00489*.

Emma Strubell, Ananya Ganesh, and Andrew McCallum. 2019. Energy and policy considerations for deep learning in nlp. *arXiv preprint arXiv:1906.02243*.

Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. 2019a. Videobert: A joint model for video and language representation learning. *CoRR*, abs/1904.01766.

Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2019b. How to fine-tune BERT for text classification? *CoRR*, abs/1905.05583.

Wilson L Taylor. 1953. cloze procedure: A new tool for measuring readability. *Journalism Bulletin*, 30(4):415–433.

Andrea Vanzo, Emanuele Bastianelli, and Oliver Lemon. 2019. Hierarchical multi-task natural language understanding for cross-domain conversational ai: Hermit nlu. pages 254–263.

Akson Sam Varghese, Saleha Sarang, Vipul Yadav, Bharat Karotra, and Niketa Gandhi. 2020. Bidirectional lstm joint model for intent classification and named entity recognition in natural language understanding. In *Intelligent Systems Design and Applications*, pages 58–68, Cham. Springer International Publishing.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Vladimir Vlasov, Johannes EM Mosig, and Alan Nichol. 2019. Dialogue transformers. *arXiv preprint arXiv:1910.00486*.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. *CoRR*, abs/1905.00537.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. *CoRR*, abs/1804.07461.

Jason D Williams, Kavosh Asadi, and Geoffrey Zweig. 2017. Hybrid code networks: practical and efficient end-to-end dialog control with supervised and reinforcement learning. *arXiv preprint arXiv:1702.03274*.

Jason D Williams and Steve Young. 2007. Partially observable markov decision processes for spoken dialog systems. *Computer Speech & Language*, 21(2):393–422.

Ledell Wu, Adam Fisch, Sumit Chopra, Keith Adams, Antoine Bordes, and Jason Weston. 2017. Starspace: Embed all the things! *arXiv preprint arXiv:1709.03856*.

Ryota Yoshihashi, Wen Shao, Rei Kawakami, Shaodi You, Makoto Iida, and Takeshi Naemura. 2018. Classification-reconstruction learning for open-set recognition. *CoRR*, abs/1812.04246.

Xiaodong Zhang and Houfeng Wang. 2016. A joint model of intent determination and slot filling for spoken language understanding. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, IJCAI'16, pages 2993–2999. AAAI Press.

Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019. ERNIE: enhanced language representation with informative entities. *CoRR*, abs/1905.07129.