ADAPTERS FOR RESOURCE-EFFICIENT DEPLOYMENT OF NLU MODELS

Jan Nehring, Nils Feldhus, Akhyar Ahmed

German Research Center for Artificial Intelligence
jan.nehring@dfki.de

Abstract: Modern Transformer-based language models such as BERT are huge and, therefore, expensive to deploy in practical applications. In environments such as commercial chatbot-as-a-service platforms that deploy many NLP models in parallel, less powerful models with a smaller number of parameters are an alternative to transformers to keep deployment costs down, at the cost of lower accuracy values. This paper compares different models for Intent Detection concerning their memory footprint, quality of Intent Detection, and processing speed. Many task-specific Adapters can share one large transformer model with the Adapter framework. The deployment of 100 NLU models requires 1 GB of memory for the proposed BERT+Adapter architecture, compared to 41.78 GB for a BERT-only architecture.

1 Introduction

Natural Language Understanding (NLU) is an essential component of dialog systems (DS). The NLU converts an unstructured user utterance into structured information: It comprises a) Intent Detection (ID), where the dialog system classifies a user utterance into a predefined list of intents. Based on this classification, the system can understand what the user wants to say. The other part of NLU is b) slot filling/entity recognition (ER), in which the dialog system fills in specific slots that belong to an intent.

Transformer-based models [1] currently show the best results in ID [2]. Transformers models are huge. A standard BERT model [3] for ID on the HWU64 dataset [4] occupies 438.01 MB of memory in the Hugging Face implementation [5]. Hosting these large models is expensive and uses large amounts of computational power, which is no longer adequate in times of climate change and scarce resources [6]. Therefore, smaller architectures were proposed, such as DIET [7] or ConveRT [8], which use substantially less memory (64.51 MB for DIET in our HWU64 example), at the cost of lower NLU accuracy.

We apply Adapters [9, 10] to ID. Adapters use a general-purpose pre-trained BERT model and introduce a small number of additional parameters (6.16 MB for our HWU64 example in the AdapterHub implementation [11]). During training, Adapters freeze the parameters of the original BERT model and train only the additional parameters. Using this approach, we propose a resource-efficient method to deploy multiple ID models: Instead of deploying multiple BERT models, we only need a single, shared BERT model and one Adapter for each downstream NLU application. The deployment of, for example, 100 ID models using this new framework requires 1,05 GB (438.01 MB for the BERT model + 100 × 6.16 MB for the Adapters) instead of 43.8 GB (100 × 438.01 MB size of the BERT model).

Our Adapter approach for deployment shows its strength in environments where many models are deployed in parallel. We propose its use in chatbot-as-a-service platforms such as

Google Dialogflow[1], SAP Conversational AI[2] and Amazon Lex[3]. These systems offer chatbots as a service, where users can create their own chatbots. Each of these user-generated chatbots uses its own NLU model, so these systems host a large number of NLU models in parallel. Using our approach, this industry can save costs, resources, and energy.

In our experiments, we investigate how different models for ID trade model size and processing speed for ID quality. We want to help practitioners to choose the suitable model for their use case. We publish the source code of our experiments on GitHub[4] under the open Apache v2 license.

2 Background

The recent increase in research on dialog systems was a catalyst for research in NLU. Systems like the Dual Intent and Entity Transformer (DIET) [7] or the Dual Sentence Encoders [8, 12] focus on lightweight models. Smaller models are competitive with large-scale models in terms of performance and are much faster in training and inference. Full-size Transformer models are used for NLU also and achieve a stronger performance than the efficient architectures [2, 7, 8]. To our knowledge, example-driven intent prediction observers [13] is the current state-of-the-art for ID, but for better comparability, we chose a standard BERT architecture over this approach for our experiments.

Instead of replacing the original model for a smaller one, Adapters [9, 10] are a lightweight addition to transformer models. An Adapter is a small set of parameters inserted between the original model's layers, usually a pre-trained transformer model. The Adapter adds a classification layer on top of the original model for a text classification task. During training, the parameters of the original Transformer model are frozen, and only the parameters of the Adapter are modified. The performance measured in accuracy or F1-score is similar to full fine-tuning on most tasks [14]. In dialog systems research, Adapters were used for the tasks dialog state tracking, response retrieval [15], and neural end-to-end dialog [16], but these works did not investigate the resource efficiency.

3 Experiments

3.1 Datasets

We use the dataset ATIS [17], Banking77 [18], CLINC150[19] and HWU64 [4] for our experiments. All datasets are datasets for NLU evaluation. Each sample consists of a user utterance which is annotated with a single intent and 0 or more entities. We did not use the entities in our work. In addition, each intent belongs to a domain. ATIS and Banking77 span one domain only, while the others cover multiple domains. Originally we wanted to use the DialoGLUE benchmark [2] which comprises several NLU datasets, including HWU64. However, the DialoGLUE version of HWU64 contains only 11k user utterances. We contacted the author, who confirmed that this is a bug. Therefore, we did not use the DialoGLUE benchmark. We split the datasets into train, valid, and test partitions.

[1] https://cloud.google.com/dialogflow

[2] https://cai.tools.sap

[3] https://aws.amazon.com/lex/

[4] https://github.com/jnehring/ESSV2023-Adapters-for-Resource-Efficient-Deployment-of-NLU-models

3.2 Models

We conduct our experiments on six models. The BERT model is the standard BERT architecture [3] with a sequence classification head. The DistilBERT model [20] reduces the size of a standard BERT model through knowledge distillation [21, 22] and is an example of a lightweight transformer architecture. The Dual Intent and Entity Transformer (DIET) [7] is an example of a lightweight model for ID. DIET uses only two instead of the usual 12 encoder layers, making DIET another example of a lightweight transformer architecture. To evaluate Adapter-based approaches, we use the Adapter versions of BERT and DistilBERT, which we call BERT+Adapter and DistilBERT+Adapter. We use the AdapterFusion architecture [23], implemented by AdapterHub [11]. Further, we use a C-Support SVM model, which is very memory efficient, but has less predictive strength. For reasons of brevity we have to omit the details how we trained the models and refer to our published source codes for further information.

4 Results

4.1 Model size

Model	Size n Models	Size 1 Model	Size 100 Models
BERT	$n \times 438.01$	438.01	43,801
BERT+Adapter	$n \times 6.16 + 438.01$	444.17	1,054
DistilBERT	$n \times 265.49$	265.49	26,549
DistilBERT+Adapter	$n \times 4.36 + 265.49$	269,85	**701**
DIET	$n \times 64.51$	64.51	6,451
SVM	$n \times 7.19$	**7.19**	719

Table 1 – Model sizes in MB, *n* denotes the number of models. The smallest model is highlighted in bold.

Table 1 shows the sizes of the individual models in MB. SVM is the most memory efficient model, while transformers use the most memory. Transformer models BERT, Distilbert and DIET have the largest memory footprint, while the adapter-based approaches BERT+Adapter and Distilbert+Adapter have similar memory footprints as the SVM when 100 models are deployed in parallel. SVM differs from the other models because it uses normal RAM memory, instead of GPU memory.

4.2 Training and inference time

Table 2 shows SPS, the number of samples processed per second, during training and inference. The SVM-based models train and predict considerably faster than the other models. The adapter-based models train considerably faster than non-adapter-based models, because they train less parameters. On the other hand, adapter-based models predict slightly slower than their non-adapter counterparts, because they have slightly more parameters. DIET is much slower; we argue that this is due to the our implementation, because we use a different machine learning framework for DIET.

Model	$\text{SPS}_{\text{train}}$	$\text{SPS}_{\text{inference}}$
BERT	14.64 (6.43)	822.67 (27.72)
BERT+Adapter	123.98 (59.97)	791.42 (24.63)
DIET	48.49 (8.76)	130.07 (3.76)
Distilbert	19.99 (7.19)	1252.28 (70.78)
Distilbert+Adapter	203.74 (51.67)	1227.73 (60.15)
SVM	**3738.24 (2494.69)**	**2652.72 (3617.64)**

Table 2 – Average training and inference speed in samples per second per model. Numbers in brackets show the standard deviation over the ten repetitions of the experiment. The maximum values are highlighted in bold

Model	ATIS	Banking77	CLINC150	HWU64	Mean
BERT	94.06%	89.74%	91.27%	87.77%	89.85% (2.68)
BERT+Adapter	96.42%	89.68%	93.73%	88.80%	91.24% (3.32)
DIET	95.29%	88.54%	87.97%	84.48%	88.85% (3.53)
Distilbert	**97.42%**	**92.31%**	**95.22%**	**91.67%**	**93.56% (2.39)**
Distilbert+Adapter	95.97%	89.22%	93.42%	90.32%	91.57% (2.71)
SVM	91.94%	86.36%	85.18%	83.47%	85.69% (3.53)

Table 3 – Accuracy values as percentage of the models on different datasets. The best performing models are highlighted in bold.

4.3 Quality of intent detection

Table 3 shows the accuracy values of the models for ID. Distilbert performs the best. Adapter-based approaches BERT+Adapter and Distilbert+Adapter 2nd best. SVM performs worst. It is surprising that BERT performs worse than Distilbert and the adapter-based approaches, because this contradicts previous research [22, 14].

5 Discussion

Our experiments show that Distilbert is the best model for superior ID accuracy when hosting only a few models or when GPU memory usage is not an important factor. In this case, one could also try different transformer models, such as RoBERTa [24] or ConvBERT [2], to further boost the performance. Adapter-based approaches are useful when many models are deployed in parallel and GPU memory efficiency is important. The faster training performance of adapter-based models is an additional plus for the practical work of chatbot designers. The inference time of adapter-based approaches is only slightly slower than of non-adapter-based approaches. The memory-savings of adapters when many models are deployed in parallel is striking.

Using SVMs for ID is an interesting alternative because it is very memory efficient, at the cost of less than 10% of ID performance. We do not know which technology commercial platforms such as IBM Watson Assistant or Google Dialogflow us for ID. But from our discussion with one commercial chatbot-as-a-service we learned that they do not use transformers; Other researcher [4] indicates the same because of the comparatively low the performance of their ID detection. Since they deploy many models in parallel, these platforms can boost their ID performance using Adapters, while staying memory efficient.

We claim in the introduction that using Adapters, the system can hold 1000 NLU models in parallel in 11 GB of memory. In our experiments, we did not explicitly show that. Instead, in section 4.1 we analyzed theoretically how the memory footprint grows as the number of modules grows.

6 Conclusion

In our experiments, we cannot give a clear answer that one of the models is superior to the others. However, we could quantify the tradeoff between resource efficiency and quality of ID. When resource efficiency is of utmost importance, SVM is the best model, although the Distilbert+Adapter architecture has a comparable memory footprint to SVM when 100 models are deployed in parallel. When ID quality is most important, large transformer models should be used. Smaller transformers such as DIET or Distilbert show weaker performance in ID similar compared to large transformers and have smaller model sizes.

To the best of our knowledge, our work is the first to point out that Adapters can host many NLU models in parallel more efficiently. We believe that this approach applies in any environment that hosts multiple models. We assume it is especially useful in environments where many models are not heavily used, such as a chatbots-as-a-service environment. Further, we believe it is applicable in many settings where users can generate and fine-tune their models, e.g., named entity recognition as a service with user-generated content. Finally, Adapters are not limited to models with the same task. Houlsby et al. [10] showed that they can also be used in environments where models for different tasks are deployed, e.g., one Adapter for Named Entity Recognition and another Adapter for Sentiment Analysis.

A follow up work to this article could transfer this approach to other application areas. Also, we leave the examination of slot filling, the other task of NLU, for adapter-based models for future research.

Acknowledgments

The authors would like to thank the Federal Ministry for financial support for Economics and Technology (BMWi) as part of the SPEAKER project (funding reference: 01MK20011R).

References

[1] Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin: *Attention is all you need*. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, vol. 2017-Decem, pp. 5999–6009. Curran Associates, Inc., 2017. URL `https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf`. 1706.03762.

[2] Mehri, S., M. Eric, and D. Hakkani-Tur: *DialoGLUE: A natural language understanding benchmark for task-oriented dialogue*. *ArXiv*, 2020. URL `https://arxiv.org/abs/2009.13570`. 2009.13570.

[3] Devlin, J., M.-W. Chang, K. Lee, and K. 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 (Long and Short Papers)*, pp. 4171–4186. Association for Computational Linguistics, Minneapolis, Minnesota, 2019. doi:10.18653/v1/N19-1423. URL `https://aclanthology.org/N19-1423`.

[4] Liu, X., A. Eshghi, P. Swietojanski, and V. Rieser: *Benchmarking Natural Language Understanding Services for building Conversational Agents*. In *Proceedings of the Tenth International Workshop on Spoken Dialogue Systems Technology (IWSDS).*

Springer, Ortigia, Siracusa (SR), Italy, 2019. URL https://arxiv.org/abs/1903.05566.

[5] Wolf, T., L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. Le Scao, S. Gugger, M. Drame, Q. Lhoest, and A. Rush: *Transformers: State-of-the-art natural language processing*. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 38–45. Association for Computational Linguistics, Online, 2020. doi:10.18653/v1/2020.emnlp-demos.6. URL https://aclanthology.org/2020.emnlp-demos.6.

[6] Strubell, E., A. Ganesh, and A. McCallum: *Energy and policy considerations for deep learning in NLP*. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 3645–3650. Association for Computational Linguistics, Florence, Italy, 2019. doi:10.18653/v1/P19-1355. URL https://aclanthology.org/P19-1355.

[7] Bunk, T., D. Varshneya, V. Vlasov, and A. Nichol: *DIET: Lightweight language understanding for dialogue systems*. *arXiv*, 2020. 2004.09936.

[8] Henderson, M., I. Casanueva, N. Mrkšić, P.-H. Su, T.-H. Wen, and I. Vulić: *ConveRT: Efficient and accurate conversational representations from transformers*. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 2161–2174. Association for Computational Linguistics, Online, 2020. doi:10.18653/v1/2020.findings-emnlp.196. URL https://aclanthology.org/2020.findings-emnlp.196.

[9] Rebuffi, S.-A., H. Bilen, and A. Vedaldi: *Learning multiple visual domains with residual adapters*. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, vol. 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/file/e7b24b112a44fdd9ee93bdf998c6ca0e-Paper.pdf.

[10] Houlsby, N., A. Giurgiu, S. Jastrzebski, B. Morrone, Q. De Laroussilhe, A. Gesmundo, M. Attariyan, and S. Gelly: *Parameter-efficient transfer learning for NLP*. In K. Chaudhuri and R. Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, vol. 97 of *Proceedings of Machine Learning Research*, pp. 2790–2799. PMLR, 2019. URL https://proceedings.mlr.press/v97/houlsby19a.html.

[11] Pfeiffer, J., A. Rücklé, C. Poth, A. Kamath, I. Vulić, S. Ruder, K. Cho, and I. Gurevych: *AdapterHub: A framework for adapting transformers*. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 46–54. Association for Computational Linguistics, Online, 2020. doi:10.18653/v1/2020.emnlp-demos.7. URL https://aclanthology.org/2020.emnlp-demos.7.

[12] Cer, D., Y. Yang, S. yi Kong, N. Hua, N. Limtiaco, R. S. John, N. Constant, M. Guajardo-Cespedes, S. Yuan, C. Tar, Y.-H. Sung, B. Strope, and R. Kurzweil: *Universal sentence encoder*. *arXiv*, 2018. 1803.11175.

[13] MEHRI, S. and M. ERIC: *Example-driven intent prediction with observers.* In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 2979–2992. Association for Computational Linguistics, Online, 2021. doi:10.18653/v1/2021.naacl-main.237. URL https://aclanthology.org/2021.naacl-main.237.

[14] PETERS, M. E., S. RUDER, and N. A. SMITH: *To tune or not to tune? adapting pretrained representations to diverse tasks.* In *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*, pp. 7–14. Association for Computational Linguistics, Florence, Italy, 2019. doi:10.18653/v1/W19-4302. URL https://aclanthology.org/W19-4302.

[15] HUNG, C.-C., A. LAUSCHER, S. P. PONZETTO, and G. GLAVAŠ: *DS-TOD: Efficient domain specialization for task oriented dialog. arXiv*, 2021. URL https://arxiv.org/pdf/2110.08395.pdf. 2110.08395.

[16] MADOTTO, A., Z. LIN, Y. BANG, and P. FUNG: *The adapter-bot: All-in-one controllable conversational model. arXiv*, 2020. URL https://arxiv.org/abs/2008.12579.

[17] HEMPHILL, C. T., J. J. GODFREY, and G. 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. URL https://aclanthology.org/H90-1021.

[18] CASANUEVA, I., T. TEMČINAS, D. GERZ, M. HENDERSON, and I. VULIĆ: *Efficient intent detection with dual sentence encoders.* In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, pp. 38–45. Association for Computational Linguistics, Online, 2020. doi:10.18653/v1/2020.nlp4convai-1.5. URL https://aclanthology.org/2020.nlp4convai-1.5.

[19] LARSON, S., A. MAHENDRAN, J. J. PEPER, C. CLARKE, A. LEE, P. HILL, J. K. KUMMERFELD, K. LEACH, M. A. LAURENZANO, L. TANG, and J. MARS: *An evaluation dataset for intent classification and out-of-scope prediction.* In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 1311–1316. Association for Computational Linguistics, Hong Kong, China, 2019. doi:10.18653/v1/D19-1131. URL https://aclanthology.org/D19-1131.

[20] SANH, V., L. DEBUT, J. CHAUMOND, and T. WOLF: *DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter.* In *5th Workshop on Energy Efficient Machine Learning and Cognitive Computing @ NeurIPS 2019.* 2019. URL http://arxiv.org/abs/1910.01108. 1910.01108.

[21] BUCILU, C., R. CARUANA, and A. NICULESCU-MIZIL: *Model compression.* In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '06, p. 535–541. Association for Computing Machinery, New York, NY, USA, 2006. doi:10.1145/1150402.1150464. URL https://doi.org/10.1145/1150402.1150464.

[22] HINTON, G., O. VINYALS, and J. DEAN: *Distilling the knowledge in a neural network.* 2015. URL http://arxiv.org/abs/1503.02531. Cite arxiv:1503.02531Comment: NIPS 2014 Deep Learning Workshop.

[23] PFEIFFER, J., A. KAMATH, A. RÜCKLÉ, K. CHO, and I. GUREVYCH: *AdapterFusion: Non-destructive task composition for transfer learning.* In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pp. 487–503. Association for Computational Linguistics, Online, 2021. doi:10.18653/v1/2021.eacl-main.39. URL `https://aclanthology.org/2021.eacl-main.39`.

[24] LIU, Y., M. OTT, N. GOYAL, J. DU, M. JOSHI, D. CHEN, O. LEVY, M. LEWIS, L. ZETTLEMOYER, and V. STOYANOV: *Roberta: A robustly optimized bert pretraining approach. ArXiv*, abs/1907.11692, 2019.