Proceedings of the 18th Conference on Natural Language Processing (KONVENS 2022)

Bye, Bye, Maintenance Work?
Using Model Cloning to Approximate the Behavior of Legacy Tools

Piush Aggarwal **Torsten Zesch**
Computational Linguistics
CATALPA - Center for Advanced Technology-Assisted Learning and Predictive Analytics
FernUniversität in Hagen
{piush.aggarwal,torsten.zesch}@fernuni-hagen.de

Abstract

A lot of NLP tools are not maintained anymore, but might still provide some unique functionality. We investigate whether such legacy tools could be replaced by a neural network that closely imitates the original behavior. For this purpose, we propose *model cloning* that can be performed by solely looking at the output of the original model, which makes the cloning possible also for black-box systems. Using a single neural architecture for cloning legacy models, caries other benefits like ease-of-use, continued maintenance, and expected speed increase. As a proof-of-concept, we clone 9 models from 5 POS tagger implementations of different complexity. The cloned models all learn to perform POS tagging on par with the legacy models, but seem not to learn the specific tagging patterns of individual legacy models.

1 Introduction

End-to-end neural models are increasingly used to build NLP tools (Tao et al., 2022; Wolf et al., 2020; Qi et al., 2020; Akbik et al., 2019; Han et al., 2019) However, legacy tools are still being used in production and for research purposes, as they might provide a unique functionality that cannot be easily replaced. Such legacy tools are often not maintained anymore and increasingly hard to use. Or outright dangerous, as the Log4Shell vulnerability[1] has turned some legacy Java tools into unmanageable security risks. They might only work with a specific OS version or with an outdated version of the programming language. Or the required models have to be secretly traded between researchers, as the official download ceased to exist. For some very important tools, it might be possible to port them to the latest technology and keep them available, but the bulk of legacy tools will soon be gone.

[1] https://en.wikipedia.org/wiki/Log4Shell

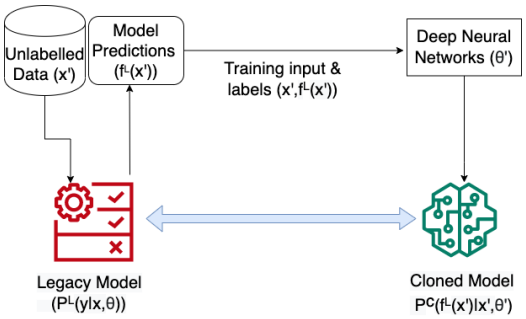

Figure 1: The model cloning process.

We argue that a possible solution is to clone the legacy models into a state-of-the-art neural model. We consider here a situation where the original training data is not available. Otherwise, we could simply retrain the model. The legacy models might also include hard-coded heuristics or dictionaries that are not reflected in the training data itself. We thus propose to apply the legacy model on plain text and then use the results to train a new model.[2]

In this paper, we choose POS tagging as a proof-of-concept use case to illustrate the potential properties of model cloning. We choose 9 different POS models from 5 legacy tools and clone their behavior into BiLSTM-CRF networks (Huang et al., 2015; Graves et al., 2013). We make all of the generated cloned models and our experimental code publicly available.[3]

2 Model Cloning

Under *model cloning*, we understand the process of copying the behavior of a legacy model by only looking at its output. Figure 1 gives an overview of the cloning process, where we select a *legacy model* ($P^L(y|x,\theta)$) which is trained on data (x, y) (unknown to us) is fed with unlabeled data (x'). To-

[2] Cloning might be restricted by the license of the legacy model.

[3] https://github.com/aggarwalpiush/model_cloning

gether with predictions ($f^L(x')$)) generated by the model these data-label pairs are use to train a deep neural network. After optimized training, the generated model ($P^c(f^L(x')|x', \theta')$) is called *cloned model*. Here θ and θ' represent model parameters.

3 Experimental Setup

To illustrate the potential properties of model cloning, we use *POS Tagging* as an example task. We apply the above mentioned model cloning architecture to classical POS taggers and evaluate how closely we can copy there behavior.

POS Taggers Table 1 lists the pre-neural legacy POS-taggers used in our experiments. We use the DKPro core framework (Eckart de Castilho and Gurevych, 2014) version of the following taggers: We use Java-based NLP4J (or ClearNLP) (Choi and Palmer, 2012), Hepple (Hepple, 2000), Mate tagger (Björkelund et al., 2010), OpenNLP[4] and Stanford (Toutanova et al., 2003).

Cloned Model Sequence labeling tasks such as POS-tagging are most promisingly taken care by linear statistical models (e.g. Conditional Random Fields (CRF) (Lafferty et al., 2001)) and neural network (NN) based models such as LSTM, BI-LSTM, etc. In our work, we use BILSTM-CRF based DNN architecture (Huang et al., 2015) for generating cloned models, where for a selected token in the text statement, a BILSTM layer carry the input text features from both direction of the sentence (Graves et al., 2013) as well as CRF layer provide sentence level tag information. We use a untrained embedding layer of 300 size input to 300 units of BILSTM cells followed by single layer of fully connected neural network having 13 units (number of classes). Model's raw predictions (pre-normalized) is used to generate CRF transition matrices which are input to a RNN cell to generate the final prediction. Negative log likelihood of CRF-layer output is used as loss function with Adam (Kingma and Ba, 2014) as an optimizer.

Note that for our proof-of-concept experiment, the actual architecture in the cloned model only needs to be powerful enough to simulate the original behavior. However, other architectures might be able to learn the same behavior from less data or reflect the behavior more closely.

[4] opennlp.apache.org

Tagger	Modelname	Domain	abbr.
Hepple	-	-	hp
Mate	Conll2009	mixed	mt
NLP4J	Ontonotes	news	on
	Mayo	medical	ma
OpenNLP	Maxent	unknown	mx
	Perceptron	unknown	pp
Stanford	csls-left3w	news	st1
	fast	unknown	st2
	wsj-0-18-csls	news	st3

Table 1: POS-taggers' models considered for cloning process.

Unlabelled Data Based on the model cloning process described in Figure 1, we use the known unlabeled data for training and labeled test data for evaluation. Note that all the labels are normalized and mapped to standard coarse grained universal tag-set (Das and Petrov, 2011). As an input to legacy models, we use web text of 1 Million sentences from news-wire platforms downloaded from the Leipzig Corpus Collection (Goldhahn et al., 2012). Before prediction, each sentence was tokenized using NLP4j's tokenizer (Choi and Palmer, 2012). We ignore the tags 'apos', '^' and 'X' in our experiments, as they are not easily mapped to coarse-grained labels for comparison.

Labeled Test Data As we also want to evaluate the objective tagging quality of the cloned models, we evaluate on a corpus with gold tags, following the setup in Horsmann et al. (2015). For evaluation, we consider formal writings, e.g. news articles, travel reports and how-to's which overlap the same domain with the known unlabeled data. We use three subsections of the GUM (Zeldes, 2017) and Brown (Francis and Kucera, 1964) corpus. Details of the corpora are provided in Table 3.

Model Training To generate the cloned models, we use the DELTA framework[5] (Han et al., 2019). We use a batch size of 36,864 for only single epoch cycle with a dropout rate of 0.5 and 0.001 as learning rate. Since our objective is to investigate how well we can learn the output of the taggers, we do not initialize the network with word embeddings to avoid any other external dependency than the training data. To generate the predictions labels, we use a 64 bit Intel(R) Xeon(R) Gold 5120 CPU @ 2.20GHz machine. For the training of cloned

[5] github.com/didi/delta

Tagger	ERROR Brown	GUM-News	GUM-Voyage	GUM-HowTo	tokens ($\times 10^3$ per sec) Cloned	Legacy	Δ
Mate (mt)	.05	.05	.05	.05	186.6	4.5	+182.1
Hepple (hp)	.04	.03	.03	.04	213.8	227.6	-13.8
OpenNLP (mx)	.04	.03	.04	.04	183.5	40.4	+143.1
OpenNLP (pp)	.06	.05	.07	.07	190.9	193.1	-2.2
Stanford (st3)	.04	.03	.04	.03	196.6	16.3	+180.3
Stanford (st1)	.04	.03	.04	.04	211.4	15.1	+196.3
Stanford (st2)	.04	.04	.04	.04	214.2	9.1	+205.1
NLP4J (ma)	.06	.04	.04	.05	208.2	26.7	+181.5
NLP4J (on)	.06	.03	.04	.04	198.9	14.9	+184.0
Average	.05	.04	.04	.04	200.5	60.9	+139.6

Table 2: The cloned models performance evaluated on labeled test data. ERROR is calculated by substracting Weighted F1 metric from 1. Δ provide tagging speed comparison with respect to legacy models.

Corpus	Tokens (x10^3)	Tagset	Sent Len ($\mu \pm \sigma$)
Brown	1,018	Brown	20.2 ± 13.1
GUM-News	8	PTB-TT	23.0 ± 12.5
GUM-Voyage	7	PTB-TT	22.0 ± 13.4
GUM-HowTo	11	PTB-TT	15.6 ± 9.9

Table 3: News domain labeled test data. Here, PTB-TT denotes penn tree bank with extended tree tagger tagset.

models, an additional 24 GB memory size Quadro RTX 6000 GPU is used.

4 Results

Table 2 shows how closely the cloned models were able to mirror the behavior of the legacy models. For that purpose, we treat the legacy results as the gold standard and report the ERROR, i.e. how much the cloned models deviates from it. We find that on average cloned models are able to approximate the behavior of legacy POS taggers with an error of 4 points. This value is statistically significant (based on McNemar Test (Dietterich, 1998) with $p < 0.05$), which means that our cloned models are significantly different from the legacy models.

Error Analysis The heatmap in Figure 2 shows where we find the major differences between legacy and cloned model. We only show results for the Stanford (st1) model, but the other models perform similarly. One source of mismatch are verb/noun and adj/noun confusions in both directions, which seems to indicate that the model has not learned the actual behavior of the legacy model. An error category that stands out is where the cloned model assigns a *NOUN* tag to what should have been *PUNCT* within the legacy model. For example in the sequence *Annapolis , Jan. 7 (special)*, the token the closing parenthesis is tagged as a noun by all cloned models.

Tagging Quality When the cloned model deviates from exactly mirroring the behavior of the legacy model, it could (i) assign a wrong tag when the legacy model was wrong, (ii) correct a mistake by the legacy model, or (iii) assign a wrong tag when also the legacy model was wrong (this last case would be neutral in term of tagging quality). To test what effect is dominating here, we also evaluate legacy models and their cloned versions on the gold labels of our evaluation corpus. We find that cloned models are either on par with legacy models or up to 2 percent points worse (in terms of average F_1). This shows that differences in behavior between legacy and cloned models are relevant for the task performance and result in worse tagging quality.

Tagging Speed To measure the tagging speed, we choose a single server setup for both legacy as well as cloned models. We only measure pure tagging speed and exclude model loading time, because when tagging a lot of text the one-time cost to load the model does not matter that much. Table 2 shows that cloned models are either much faster or on par with legacy tools. Projecting in the future, the neural models will get faster, while the legacy models are unlikely to benefit from using GPUs and improved library speed.

5 Related Work

Model cloning can be seen as a kind of *model extraction attack*, where copying a model has been investigated under the aspect of being a threat to a service's underlying business model (Yuan et al., 2022; Tramèr et al., 2016). In this scenario, an ad-

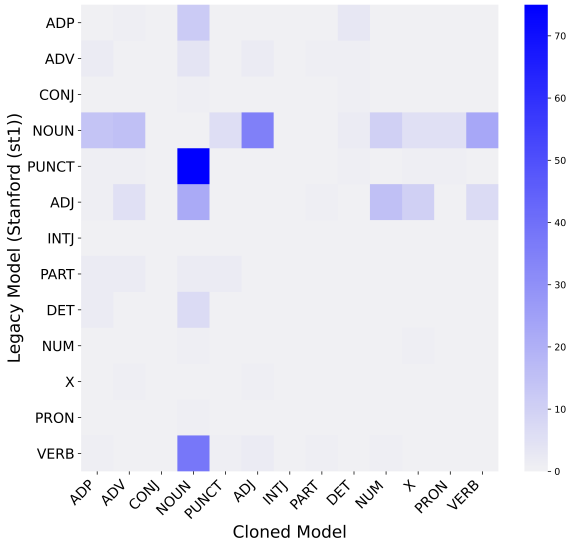

Figure 2: Heatmap illustrating failures of the cloned model to reproduce tags assigned by the Stanford legacy model (st1).

versary keeps using a model, which is offered via a paid or un-paid endpoint, until enough data has been gathered to train an own model. In particular, neural network-based model extraction is a powerful approach with their ability to approximate a function that maps an input on a certain output (Yi Shi et al., 2017). Adversaries can exploit the neural network to approximate the functionalities of endpoint services and become independent after successful cloning (Takemura et al., 2020; Atli et al., 2020). Extraction attacks are not only limited to attack model functionality, but also helps in stealing model hyper-parameters which are considered confidential specially for commercial and proprietary algorithms (Wang and Gong, 2018). Neural networks such as Knockoff Nets (Orekondy et al., 2019) are able to successfully by-pass the monetary and intellectual effort and create a reasonable cloned models as little as $30. Even cloning of real time systems such as artificial human voice synthesis (Arik et al., 2018) and autonomous driving (D'Este et al., 2003; Kuefler et al., 2017) are common practices nowadays.

Other related methods are distant (Mintz et al., 2009) and weak (Hoffmann et al., 2011) supervision which are used to build huge however relatively noisy labeled training data. They not only save time and money but are also less prone to induce human errors into the dataset. The algorithms which are used to generate the labels can be correlated with cloned model that approximate the behavioral mapping of available manually annotated data. Another area related to cloning is *Bootstrapping* (Goldman and Zhou, 2000), where machine-annotated raw data is generated as an attempt to overcome the lack of human-annotated gold data.

6 Summary

Model cloning is a potential solution to ensure the continued availability of legacy tools that are not maintained anymore. As a first experiment into model cloning, we have experimented with mirroring the behavior of 9 different pre-neural POS tagging models. We find that the cloned models come close in terms of POS tagging performance, but somewhat fail to closely resemble the specific behavior of individual taggers.

Our results are limited by only experimenting with POS tagging as one example task and by using only one neural architecture. Some NLP tasks might lend themselves more easily to cloning and some neural architecture might be better suited for cloning. In future work, we thus want to improve the cloning process to better capture the specific behavior of a given model the and to extend the paradigm to other tasks beyond POS tagging.

Acknowledgments

This work was conducted in the framework of CATALPA - Center for Advanced Technology-Assisted Learning and Predictive Analytics of the FernUniversität in Hagen, Germany.

References

Alan Akbik, Tanja Bergmann, Duncan Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf. 2019. Flair: An easy-to-use framework for state-of-the-art nlp. In *NAACL 2019, 2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 54–59.

Sercan Arik, Jitong Chen, Kainan Peng, Wei Ping, and Yanqi Zhou. 2018. Neural voice cloning with a few samples. In *Advances in Neural Information Processing Systems*, pages 10019–10029.

Buse Gul Atli, Sebastian Szyller, Mika Juuti, Samuel Marchal, and N. Asokan. 2020. Extraction of Complex DNN Models: Real Threat or Boogeyman? In *Engineering Dependable and Secure Machine Learning Systems*, pages 42–57, Cham. Springer International Publishing.

Anders Björkelund, Bernd Bohnet, Love Hafdell, and Pierre Nugues. 2010. A high-performance syntactic and semantic dependency parser. In *Proceedings of the 23rd International Conference on Computational Linguistics: Demonstrations*, COLING '10, page 33–36, USA. Association for Computational Linguistics.

Richard Eckart de Castilho and Iryna Gurevych. 2014. A broad-coverage collection of portable NLP components for building shareable analysis pipelines. In *Proceedings of the Workshop on Open Infrastructures and Analysis Frameworks for HLT*, pages 1–11, Dublin, Ireland. Association for Computational Linguistics and Dublin City University.

Jinho D. Choi and Martha Palmer. 2012. Fast and robust part-of-speech tagging using dynamic model selection. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 363–367, Jeju Island, Korea. Association for Computational Linguistics.

Dipanjan Das and Slav Petrov. 2011. Unsupervised part-of-speech tagging with bilingual graph-based projections. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 600–609, Portland, Oregon, USA. Association for Computational Linguistics.

Claire D'Este, Mark O'Sullivan, and Nicholas Hannah. 2003. Behavioural cloning and robot control. In *Robotics and Applications*, pages 179–182.

Thomas G. Dietterich. 1998. Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms. *Neural Computation*, 10(7):1895–1923.

W Nelson Francis and Henry Kucera. 1964. Brown corpus. *Department of Linguistics, Brown University, Providence, Rhode Island*, 1.

Dirk Goldhahn, Thomas Eckart, and Uwe Quasthoff. 2012. Building large monolingual dictionaries at the leipzig corpora collection: From 100 to 200 languages. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012)*, pages 759–765, Istanbul, Turkey. European Languages Resources Association (ELRA).

Sally A. Goldman and Yan Zhou. 2000. Enhancing supervised learning with unlabeled data. In *Proceedings of the Seventeenth International Conference on Machine Learning*, ICML '00, pages 327–334, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Alex Graves, Abdel-rahman Mohamed, and Geoffrey E. Hinton. 2013. Speech recognition with deep recurrent neural networks. *CoRR*, abs/1303.5778.

Kun Han, Junwen Chen, Hui Zhang, Haiyang Xu, Yiping Peng, Yun Wang, Ning Ding, Hui Deng, Yonghu Gao, Tingwei Guo, Yi Zhang, Yahao He, Baochang Ma, Yulong Zhou, Kangli Zhang, Chao Liu, Ying Lyu, Chenxi Wang, Cheng Gong, Yunbo Wang, Wei Zou, Hui Song, and Xiangang Li. 2019. DELTA: A DEep learning based Language Technology plAtform. *arXiv e-prints*.

Mark Hepple. 2000. Independence and Commitment: Assumptions for Rapid Training and Execution of Rule-based POS Taggers. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, pages 278–277, Hong Kong. Association for Computational Linguistics.

Raphael Hoffmann, Congle Zhang, Xiao Ling, Luke Zettlemoyer, and Daniel S Weld. 2011. Knowledge-based weak supervision for information extraction of overlapping relations. In *Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies*, pages 541–550.

Tobias Horsmann, Nicolai Erbs, and Torsten Zesch. 2015. Fast or Accurate ? – A Comparative Evaluation of PoS Tagging Models. In *Proceedings of the International Conference of the German Society for Computational Linguistics and Language Technology (GSCL-2015)*, pages 22–30, Essen, Germany.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional LSTM-CRF models for sequence tagging. *CoRR*, abs/1508.01991.

Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *International Conference on Learning Representations*.

A. Kuefler, J. Morton, T. Wheeler, and M. Kochenderfer. 2017. Imitating driver behavior with generative adversarial networks. In *2017 IEEE Intelligent Vehicles Symposium (IV)*, pages 204–211.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, page 282–289, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 1003–1011.

Tribhuvanesh Orekondy, Bernt Schiele, and Mario Fritz. 2019. Knockoff Nets: Stealing Functionality of Black-Box Models. In *CVPR*.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A Python Natural Language Processing Toolkit for Many Human Languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*.

Tatsuya Takemura, Naoto Yanai, and Toru Fujiwara. 2020. Model Extraction Attacks on Recurrent Neural Networks. *Journal of Information Processing*, 28:1010–1024.

Zhihua Tao, Chunping Ouyang, Yongbin Liu, Tonglee Chung, and Yixin Cao. 2022. Multi-head attention graph convolutional network model: End-to-end entity and relation joint extraction based on multi-head attention graph convolutional network. *CAAI Transactions on Intelligence Technology*.

Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. 2003. Feature-rich Part-of-speech Tagging with a Cyclic Dependency Network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, pages 173–180, Stroudsburg, PA, USA. Association for Computational Linguistics.

Florian Tramèr, Fan Zhang, Ari Juels, Michael K. Reiter, and Thomas Ristenpart. 2016. Stealing Machine Learning Models via Prediction APIs. In *Proceedings of the 25th USENIX Conference on Security Symposium*, SEC'16, pages 601–618, Berkeley, CA, USA. USENIX Association.

B. Wang and N. Gong. 2018. Stealing Hyperparameters in Machine Learning. In *2018 IEEE Symposium on Security and Privacy (SP)*, pages 36–52.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi 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 Rush. 2020. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Yi Shi, Y. Sagduyu, and A. Grushin. 2017. How to steal a machine learning classifier with deep learning. In *2017 IEEE International Symposium on Technologies for Homeland Security (HST)*, pages 1–5.

Xiaoyong Yuan, Leah Ding, Lan Zhang, Xiaolin Li, and Dapeng Oliver Wu. 2022. ES Attack: Model Stealing Against Deep Neural Networks Without Data Hurdles. *IEEE Transactions on Emerging Topics in Computational Intelligence*, pages 1–13.

Amir Zeldes. 2017. The GUM Corpus: Creating Multilayer Resources in the Classroom. *Lang. Resour. Eval.*, 51(3):581–612.