



COBURG
UNIVERSITY
of applied sciences and arts



(How well) Can ChatGPT Program?

Jochen L. Leidner Michael Reiche

Workshop on *AI for AI Learning* Held at ECAI 2023
Kraków, Poland - 2023-09-30



Outline

- Introduction
- Study Design
- Dataset Scope & Sample
- Results
- Discussion & Limitations
- Summary, Conclusion and Future Work
- *Postscript: But Is it a Real Problem?*



Introduction

- Background
 - Large (billion+ parameter, billion+ input token), pre-trained neural language models happened (Google BERT, Open AI GPT-2/3/4 Las & ChatGPT application, Meta Llama etc.)
- Motivation
 - As part of project VoLL-KI, a project between 3 German universities (Bamberg, Erlangen and Coburg), we seek to develop technology that can improve effectiveness of learning for students of artificial intelligence.
- Research Questions
 - Can ChatGPT assist us in assisting students when they learning how to program?
 - **(How well) can ChatGPT program? [this paper]**
 - Can students equipped with ChatGPT (and little to no machine learning knowledge) build a classifier?
 - **What Do students learn that use ChatGPT when learning to program? [Reiche and Leidner, at this workshop]**



Study Design

- Scope
 - Collect a set of programming questions (made up by first author)
 - Ask LM for answers
 - (Human) language of the question: English
 - Programming languages: most common ones (that the authors know)
 - Typical practical problems (not “brain teasers” or tests of boundary knowledge)
- Method
 - Curate set of questions
 - Write program to query OpenAI’s API (model: using 2023-05-23 snapshot of ChatGPT and GPT-Turbo3.5)
 - Ask all questions and database answers on 2023-07-20
 - Evaluate correctness manually using a crude ordinary scoring method (0..2 points): wrong, partially correct, correct
- Dataset
 - N=105 questions



Dataset Scope

- Programming languages

Python	C++	Java	C
JavaScript	Julia	Kotlin	Bash/sh
Prolog	Scheme/LISP	Rust	FORTRAN
COBOL	R	Matlab	

- Question types

No	Question Type	Example
1	Syntax	<i>How do I formulate X in language Y</i>
2	Idiom	<i>How do I achieve to get X done in language Y</i>
3	Explain	<i>What does code X do?</i>
4	Diagnose	<i>Why does code X not work?</i> <i>Why does code X not do Y?</i>

- Computing concepts

abstraction	ACID	anonymous function
divide and conquer	domain-specific language	generator function
immutable object	linked list	mailbox
recursion	REST	reusability
singleton class	save data to a file	window

- Error types

syntax error	wrong type	lifetime error
scope error	logical error	forgot to initialize
unallocated memory	index out of bounds	null pointer error



Dataset Sample

- In Julia, how can I view the compiled code for a function?
- In Rust, how can I design functions to return errors systematically?
- In Rust, how do I implement what would be a class in Java or C++?
- In Rust, how do I make a struct printable?
- Show me a Kotlin class for singly linked lists.
- In Rust, how do I implement what would be a class in Java or C++?
- In Rust, how do I make a struct printable?
- In Rust, which library is best for fast trie lookup in RAM?
- In Rust, which library provides efficient B-tree storage on disk?
- In Kotlin, how does the minimum code of an Android mobile app look like?
- In Kotlin, what is the syntax for anonymous functions?
- What does this C++ code do? ...
- What does this Python code do? ...
- What does the following Python code do? ...
- How can I test in Python whether a CUDA GPU is present?
- Explain recursion.
- Explain the difference between transient and persistent.
- In databases, explain the ACID acronym.
- Show me an SQL query that computes aggregate statistics about a table.
- If I have an SQL table defined by "CREATE TABLE t(...)", how can I insert a new author only if he or she does not already exist?
- In FORTRAN 95, how can I multiply two 2x2 matrices of integers A and B?
- Can I separate a function's declaration from its implementation in FORTRAN?
- Can I separate a function's declaration from its implementation in C++?
- Can I separate a function's declaration from its implementation in Java?
- Explain tail recursion to me.

- Explain the difference between transient and persistent.
- In databases, explain the ACID acronym.
- Show me an SQL query that computes aggregate statistics about a table.
- If I have an SQL table defined by "CREATE TABLE t(...)", how can I insert a new author only if he or she does not already exist?
- In FORTRAN 95, how can I multiply two 2x2 matrices of integers A and B?
- Can I separate a function's declaration from its implementation in FORTRAN?
- Can I separate a function's declaration from its implementation in C++?
- Can I separate a function's declaration from its implementation in Java?
- Explain tail recursion to me.
- Show me all bugs and deficiencies in this C code:

```
void nb_net_init(void) { nb_init_timers(); nb_net_state = malloc(sizeof(nb_net_state)); nb_net_state->num_conn = 0; }
```
- What is wrong in the following Python code: ... ? Explain all errors or bugs to me.
- Show me all errors or bugs in the following C function: ... How does it look like when it is corrected?
- Show me a set of C functions for creating a (singly) linked list, inserting data to a linked list, deleting an item from a linked list and freeing a linked list.



Preliminary Results

- Quantitative findings:
 - Our question dataset was processed in 12:26 Minutes wall-clock time (including server-side processing and network round-trips) on a MacBook Air 10 (2021) with ARM M1 processor.(Note this is not very meaningful, as the processing was done on OpenAI's servers.)
 - Score: a total **158 points** (out of 210 reachable) or **78.24%**
 - For each of the N=105 questions, a **mean of 1.7785 points** was reached [0..2]
- Qualitative findings:
 - Given that foundational language model models like GPT-3 & co. were not primarily developed for code, the results are impressive: the responses are often impeccable.
 - However, it is unhelpful that the system projects a confidence also and in particular when the results are wrong and incomplete.
 - Partially correct answers dominate, no guarantee that code snippets check for errors or even compile



Qualitative Evaluation (1/2)

► Human Question I:

What is divide and conquer? Explain it to me using a chess game as an example.

► Machine Answer I:

Divide and conquer is a programming paradigm that involves breaking down a problem into smaller subproblems, solving them independently, and then combining the solutions to solve the original problem. It is often used to solve complex problems more efficiently. In the domain of writing a chess game, an example of using divide and conquer could be implementing the move validation logic. Instead of checking the validity of a move for the entire chessboard, the board can be divided into smaller sections, such as rows or columns. Each section can then be validated independently, reducing the complexity of the problem. For instance, when validating a move for a specific piece, the program can divide the chessboard into rows and check if the move is valid within that row. Similarly, it can divide the board into columns and validate the move within each column. By breaking down the problem into smaller sections, the move validation process becomes more manageable and efficient.

► Human Question II:

What is the difference between String and StringBuffer in Java?

► Machine Answer II:

The main difference between StringBuffer and String in Java is that StringBuffer is mutable, meaning its value can be changed, while String is immutable, meaning its value cannot be changed once it is created. In other words, when you modify a StringBuffer object, it does not create a new object, but instead modifies the existing object. This can be useful when you need to perform a lot of string manipulation operations, as it avoids creating multiple new objects. On the other hand, when you modify a String object, it creates a new object with the modified value, leaving the original object unchanged. This can be less efficient when performing a lot of string manipulation operations, as it involves creating multiple new objects. Another difference is that StringBuffer is synchronized, which means it is thread-safe and can be used in multi-threaded environments without any issues. String, on the other hand, is not synchronized. In summary, if you need to perform a lot of string manipulation operations or if you are working in a multi-threaded environment, StringBuffer is a better choice. If you don't need to modify the string and want to ensure immutability, String is the way to go.



Qualitative Evaluation (2/2)

► Machine Answer III:

You can load a file containing whitespace-separated integers into an array in Python by following these steps:

```
1. Open the file using the 'open()' function and specify  
the file path and mode. For example, to open a file named  
"numbers.txt" in read mode, you can use: (...) file =  
open("numbers.txt", "r") (...)
```

- Code in the response works
- Well-explained
- But: hypothetical “file not found” situation is not handled
 - Many online Websites (StackExchange answers) may also lack this level of sophistication.
 - On those, human discussants may contest incomplete answers once recognised, but not guaranteed (volunteer crowd)



Work in Progress - Discussion & Limitations

- Pros:
 - Easy to conduct the experiment
 - Main strength is main weakness: plenty of experimental control, as not real student data (we knew the questions and what to expect as an answer, so we could probe broadly many corners of ChatGPT's programming knowledge)

- Cons:
 - Small dataset:
 - data set overall size is okay but small number of questions per question type and per language
 - need **more code example questions** (esp. from **real students**, but with their consent)
 - ▶ **open dataset**; all invited to contribute 🐞

<https://github.com/Information-Access-Research-Group-IARG/Prorogramm-for-ChatGPT>



- Single-rater, no inter-annotator or intra-annotator agreement measurement
- Scoring scheme crude: would be nice to categorise error types in a granular fashion



Summary, Conclusions and Future Work

- **Summary**

- We presented a study and dataset for evaluating the programming knowledge captured by language models like GPT-3.

- **Conclusions**

- **Our first finding is that the overall knowledge level is suprising: nearly 100% of the questions were at least partially correctly answered.**

- **Our second finding is that most questions are only partially correctly answererd.**

This poses two problems: (1) if answers are used by uncritically pasting them into production code, unsafe, buggy and outright dangerous software will be the result, and people may even die as a consequence. (2) In teaching, model answers given must be correct; students may take legal action if they do not receive credit for solution they were taught, even if these were wrong/incomplete.

- **Future work**

- The database of questions should be amended (more code snippet + error description type questions, more programming languages, more questions per programming language, questions asked in human languages other than English etc.).
- Results should be re-run after several years to test if “the Web” and language models trained on it, improve over time (in this narrow space).
- Other language models should be tried with our (open) dataset, notably GPT-4 and open models like Llama.
- We only looked at the top-1 output of the LM. Experiments with various temperatures and multiple results would be interesting (thanks to an anonymous reviewer for pointing this out).
- ***We need to explore whether or how to best integrate the powerful-yet-imperfect capability responsibly into student-facing support solutions.***



Postscript: But Is It a Real Problem?

- Does the expressed concern constitute a problem with real “copy & paste” from Websites (and by implication with language models trained on such content)?



Postscript: But Is It a Real Problem?

- YES!

The most copied StackOverflow snippet of all time
is flawed!

by Andreas Lundblad, 2019-12-02

<https://programming.guide/worlds-most-copied-so-snippet.html>





Baltes and Diehl (2019: 11)

Usage and Attribution of Stack Overflow Code Snippets in GitHub Projects

11

Table 1 RQ 1 – Phase 1: Ten most frequently referenced code snippets from SO Java answers; one asterisk: link was broken and referred to a question, we selected two referenced snippets; two asterisks: snippet based on external resource, but adapted.

Rank	Answer ID	Description	Type	Ext. Availability
1	3758880	human readable byte size	method	blog, no license
2	5445161**	read InputStream to String	method	blog, no license
3	9855338**	convert byte array to hex String	method	other SO post
4	26196831	Android: RecyclerView onClick	class	none
5	7696791*	Android: close soft keyboard	snippet	none
6	140861	hex dump String to byte array	class	none
7	2581754	sort Map<Key, Value> by values	class	none
8	5599842	format file size as MB, GB, etc.	method	none
9	326440	create Java String from file cont.	method	none
10	3145655	Android: get current location	class	none

encing Java files as possible, while taking care that it does not become too generic, leading to false positives.

Table 1 lists the ten most frequently referenced Java answers in the table. The first column shows the rank, the second column shows the answer ID, the third column shows a short description of the snippet, the fourth column shows the type of the snippet, and the fifth column shows the external availability of the snippet.

Usage and attribution of Stack Overflow code snippets in GitHub projects

Sebastian Baltes¹ · Stephan Diehl¹

Published online: 1 October 2018
© Springer Science+Business Media, LLC, part of Springer Nature 2018

Abstract

Stack Overflow (SO) is the most popular question-and-answer website for software developers, providing a large amount of copyable code snippets. Using those snippets raises maintenance and legal issues. SO's license (CC BY-SA 3.0) requires attribution, i.e., referencing the original question or answer, and requires derived work to adopt a compatible license. While there is a heated debate on SO's license model for code snippets and the required attribution, little is known about the extent to which snippets are copied from SO without proper attribution. We present results of a large-scale empirical study analyzing the usage and attribution of non-trivial Java code snippets from SO answers in public GitHub (GH) projects. We followed three different approaches to triangulate an estimate for the ratio of unattributed usages and conducted two online surveys with software developers to complement our results. For the different sets of projects that we analyzed, the ratio of projects containing files with a reference to SO varied between 3.3% and 11.9%. We found that at most 1.8% of all analyzed repositories containing code from SO used the code in a way compatible with CC BY-SA 3.0. Moreover, we estimate that at most a quarter of the copied code snippets from SO are attributed as required. Of the surveyed developers, almost one half admitted copying code from SO without attribution and about two thirds were not aware of the license of SO code snippets and its implications.



A StackExchange Question...

A Long Long Time Ago...

Back in 2010 I was sitting in my office and doing what I wasn't supposed to be doing: code golfing and chasing reputation on Stack Overflow.

The following question got my attention: How do you print a byte count in a human readable format? That is, how do you format something like 123,456,789 bytes as "123.5 MB".


10


3

How to convert byte size into human readable format in java?

How to convert byte size into human-readable format in Java? Like 1024 should become "1 Kb" and 1024*1024 should become "1 Mb".

I am kind of sick of writing this utility method for each project. Are there any static methods in Apache Commons for this?

[java](#) [formatting](#) [apache commons](#)

[link](#) | [flag](#)

 **unwind**
46.7k • 4 = 50 • 125

 **limuhin**
296 • 7
100% accept rate

edited Sep 21 at 9:01

asked Sep 21 at 8:42

The good ol' 2010 look, courtesy the Wayback Machine.



...Lundblad's 2010 Initial Answer

```
public static String humanReadableByteCount(long bytes, boolean si) {  
    int unit = si ? 1000 : 1024;  
    if (bytes < unit) return bytes + " B";  
    int exp = (int) (Math.log(bytes) / Math.log(unit));  
    String pre = (si ? "KMGTPE" : "KMGTPE").charAt(exp-1) + (si ? "" : "i");  
    return String.format("%.1f %sB", bytes / Math.pow(unit, exp), pre);  
}
```



A Problem

- But: Does *not* work for numbers near closest rounded bigger number e.g. 998998 bytes should be rounded up to be shown as “1 MB” (Lundblatt, *ibid.*):

The solution works all the way up until it approaches 1 MB. When given 999,999 bytes as input, the result (in SI mode) is “1000.0 kB”. While it is true that 999,999 is closer to $1,000 \times 1000^1$ than it is to 999.9×1000^1 , the 1,000 “significand” is out of range according to spec. The correct result is “1.0 MB”.

FWIW, all 22 answers posted, including the ones using Apache Commons and Android libraries, had this bug (or a variation of it) at the time of writing this article.

So how do we fix this? First of all, we note that the exponent (`exp`) should change from ‘k’ to ‘M’ as soon as the number of bytes is closer to $1 \times 1,000^2$ (1 MB) than it is to 999.9×1000^1 (999.9 k). This happens at 999,950. Similarly, we should switch from ‘M’ to ‘G’ when we pass 999,950,000 and so on.

To achieve this we calculate this threshold and bump `exp` if `bytes` is larger. (For the binary case, this threshold is not an integer, so we need to ceil the result.)



...and Lundblad's 2023 Corrected Version (From This Week)

```
// From: https://programming.guide/worlds-most-copied-so-snippet.html
public static strictfp String humanReadableByteCount(long bytes, boolean si) {
    int unit = si ? 1000 : 1024;
    long absBytes = bytes == Long.MIN_VALUE ? Long.MAX_VALUE : Math.abs(bytes);
    if (absBytes < unit) return bytes + " B";
    int exp = (int) (Math.log(absBytes) / Math.log(unit));
    long th = (long) Math.ceil(Math.pow(unit, exp) * (unit - 0.05));
    if (exp < 6 && absBytes >= th - ((th & 0xFFFF) == 0xD00 ? 51 : 0)) exp++;
    String pre = (si ? "kMGTPE" : "KMGTPE").charAt(exp - 1) + (si ? "" : "i");
    if (exp > 4) {
        bytes /= unit;
        exp -= 1;
    }
    return String.format("%.1f %sB", bytes / Math.pow(unit, exp), pre);
}
```



Acknowledgments

- We would like to thank:
 - Our anonymous reviewers, who helped us increase the quality of the paper.
 - The Free State of Bavaria, for a generous Research Professorship grant to the first author that financed this research and our travel under the *Hightech Agenda Bavaria* R&D programme.
 - The German Federal Ministry for Research and Technology BMBF for supporting Project VoLL-KI (grant 16DHBKI089), which made this research and our workshop today possible.
 - Paul Graham's news aggregator *Hacker News* for sharing Lundblad's story





BACKUP SLIDES

Breaking News: see also [Rozière et al. \(2023\)](#) from Meta

Code Llama: Open Foundation Models for Code

Baptiste Rozière¹, Jonas Gehring¹, Fabian Gloeckle^{1,†}, Sten Sootla¹, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Tal Remez, Jérémy Rapin, Artyom Kozhevnikov, Ivan Evtimov, Joanna Bitton, Manish Bhatt, Cristian Canton Ferrer, Aaron Grattafiori, Wenhan Xiong, Alexandre Défossez, Jade Copet, Faisal Azhar, Hugo Touvron, Louis Martin, Nicolas Usunier, Thomas Scialom, Gabriel Synnaeve¹

Meta AI

Abstract

We release CODE LLAMA, a family of large language models for code based on LLAMA 2 providing state-of-the-art performance among open models, infilling capabilities, support for large input contexts, and zero-shot instruction following ability for programming tasks. We provide multiple flavors to cover a wide range of applications: foundation models (CODE LLAMA), Python specializations (CODE LLAMA - PYTHON), and instruction-following models (CODE LLAMA - INSTRUCT) with 7B, 13B and 34B parameters each. All models are trained on sequences of 16k tokens and show improvements on inputs with up to 100k tokens. 7B and 13B CODE LLAMA and CODE LLAMA - INSTRUCT variants support infilling based on surrounding content. CODE LLAMA reaches state-of-the-art performance among open models on several code benchmarks, with scores of up to 53% and 55% on HumanEval and MBPP, respectively. Notably, CODE LLAMA - PYTHON 7B outperforms LLAMA 2 70B on HumanEval and MBPP, and all our models outperform every other publicly available model on MultiPL-E. We release CODE LLAMA under a permissive license that allows for both research and commercial use.¹



References (1/2)

- 1. Barke, S., James, M.B., Polikarpova, N.: Grounded Copilot: how programmers interact with code-generating models. Unpublished manuscript, ArXiv.org pre-print server, Cornell University, New York, NY, USA (2022), <https://arxiv.org/abs/2206.15000>
- 2. Bird, C., Ford, D., Zimmermann, T., Forsgren, N., Kalliamvakou, E., Lowdermilk, T., Gazit, I.: Taking flight with copilot. *Communications of the ACM* 66(6), 56–62 (2023). <https://doi.org/10.1145/3589996>
- 3. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., Amodei, D.: Language models are few-shot learners. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H. (eds.) *Advances in Neural Information Processing Systems*. vol. 33, pp. 1877–1901. Curran (2020), https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfc4967418bfb8ac142f64a-Paper.pdf
- 4. Carbonell, J.R.: AI in CAI: An artificial-intelligence approach to computer-assisted instruction. *IEEE Transactions on Man-Machine Systems* 11(4), 190–202 (1970). <https://doi.org/10.1109/TMMS.1970.299942>
- 5. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: 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. ACL, Minneapolis, MN, USA (2019). <https://doi.org/10.18653/v1/N19-1423>
- 6. Imai, S.: Is GitHub Copilot a substitute for human pair-programming? an empirical study. In: *Proceedings of the ACM/IEEE 44th International Conference on Software Engineering: Companion Proceedings*. pp. 319–321. ICSE '22, ACM, New York, NY, USA (2022). <https://doi.org/10.1145/3510454.3522684>
- 7. Koulouri, T., Lauria, S., Macredie, R.D.: Teaching introductory programming: A quantitative evaluation of different approaches. *ACM Trans. Comput. Educ.* 14(4), 1–28 (2015). <https://doi.org/10.1145/2662412>
- 8. Leinonen, J., Denny, P., MacNeil, S., Sarsa, S., Bernstein, S., Kim, J., Tran, A., Hellas, A.: Comparing code explanations created by students and large language models (2023), unpublished manuscript, arXiv cs.CY 2304.03938, Cornell University pre-print server
- 9. MacNeil, S., Tran, A., Hellas, A., Kim, J., Sarsa, S., Denny, P., Bernstein, S., Leinonen, J.: Experiences from using code explanations generated by large language models in a web software development e-book. In: *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1*. pp. 931–937. SIGCSE 2023, Association for Computing Machinery, New York, NY, USA (2023). <https://doi.org/10.1145/3545945.3569785>



References (2/2)

- 10. Reiche, M., Leidner, J.: Bridging the programming skill gap with ChatGPT: A machine learning project with business students. In: Nowacyk et al., S. (ed.) ECAI 2023 Workshops, Kraków, Poland. CCIS, Springer Nature, Cham, Switzerland (2023), Workshop on AI for AI Learning.
- 11. Robins, A., Rountree, J., Rountree, N.: Learning and teaching programming: A review and discussion. *Computer Science Education* 13(2), 137–172 (2003) LM-Assisted Learning How to Program?
- 12. Roumeliotis, K.I., Tselikas, N.D.: ChatGPT and Open-AI models: A preliminary review. *Future Internet* 15(6) (2023). <https://doi.org/10.3390/fi15060192>, <https://www.mdpi.com/1999-5903/15/6/192>
- 13. Sanderson, K.: GPT-4 is here: what scientists think. *Nature* 615(7954), 773 (2023)
- 14. Sarsa, S., Denny, P., Hellas, A., Leinonen, J.: Automatic generation of programming exercises and code explanations using large language models. In: *Proceedings of the 2022 ACM Conference on International Computing Education Research - Volume 1*. pp. 27–43. ICER 2022, Association for Computing Machinery, New York, NY, USA (2022). <https://doi.org/10.1145/3501385.3543957>
- 15. Su, Y., Wan, C., Sethi, U., Lu, S., Musuvathi, M., Nath, S.: HotGPT: How to make software documentation more useful with a large language model? In: *Proceedings of the 19th Workshop on Hot Topics in Operating Systems*. pp. 87–93. HOTOS '23, Association for Computing Machinery, New York, NY, USA (2023). <https://doi.org/10.1145/3593856.3595910>
- 16. Surameery, N.M.S., Shakor, M.Y.: Use ChatGPT to solve programming bugs. *International Journal of Information Technology & Computer Engineering* 3(01), 17–22 (2023). <https://doi.org/10.55529/ijitc.31.17.22>, <http://journal.hmjournals.com/index.php/IJITC/article/view/1679>
- 17. Vaithilingam, P., Zhang, T., Glassman, E.: Expectation vs. experience: evaluating the usability of code generation tools powered by large language models. In: *Extended Abstracts of the 2022 Conference on Human Factors in Computing Systems*. pp. 1–7 (2022), <https://dl.acm.org/doi/10.1145/3491101.3519665>
- 18. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) *Advances in Neural Information Processing Systems 30 – 31st Annual Conference on Neural Information Processing Systems, Long Beach, CA, December 4-9, 2017*. pp. 5999– 6010. (NIPS 2017), Curran Associates (2018)



About the Authors

- **Prof. Jochen L. Leidner, M.A. M.Phil. Ph.D.** is Research Professor of Artificial Intelligence in Insurance at Coburg University of Applied Sciences and Arts, where he leads the Information Access Research Group, Visiting Professor of Data Analytics at the University of Sheffield, and CEO of KnowledgeSpaces UG. He obtained Master's degrees in Computational Linguistics, English and Computer Science from Friedrich-Alexander University (FAU) Erlangen-Nuremberg and in Computer Speech, Text and Internet Technologies from the University of Cambridge and his Doctorate at the School of Informatics, College of Engineering, University of Edinburgh.
- **Michael Reiche, B.Sc. M.A.** is a doctoral candidate at Coburg University of Applied Sciences and Arts. His Ph.D. thesis work focuses on machine learning process methodology. He received a Bachelor degree in Computer Science and a Master's degree in Business, both from Coburg.

