

# KI-PROGRAMMIERUNG

Ein neues Kapitel in der Software-Entwicklung?



MAIBORNWOLFF



**Marius Wichtner**  
Lead Software Engineer

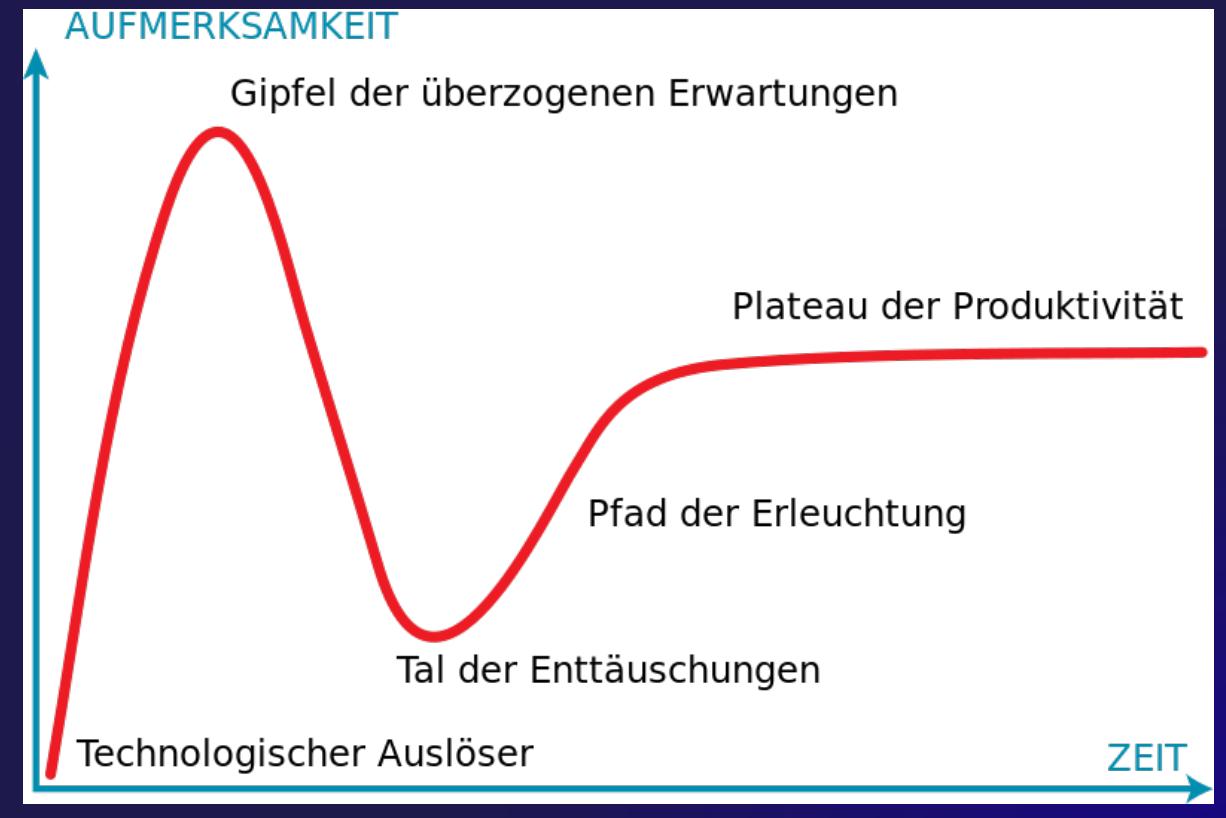




Gerald Ford, 1973



## Gartner Hype Cycle





Quelle: Google Trends



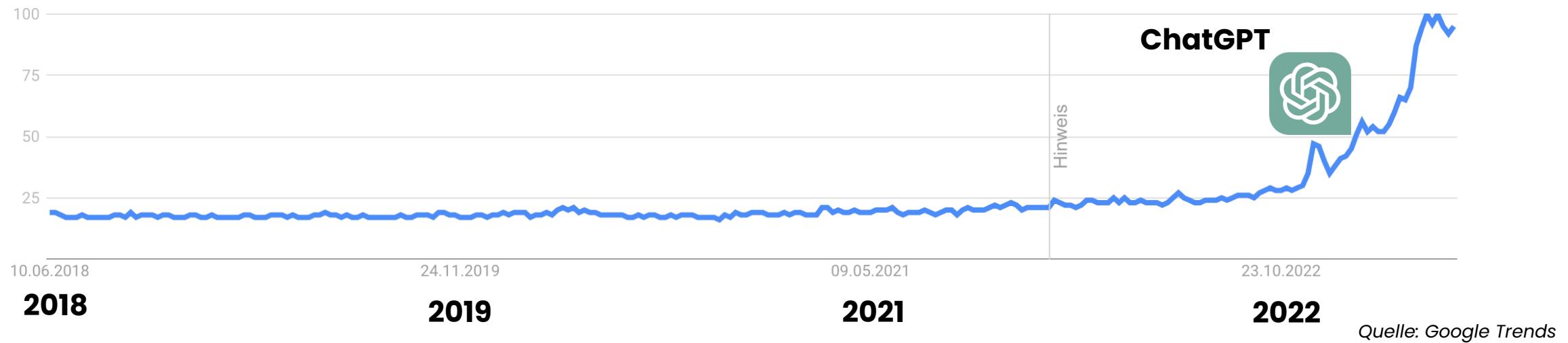
## Trend: "Cloud Computing"



Quelle: Google Trends



## Trend: "AI"

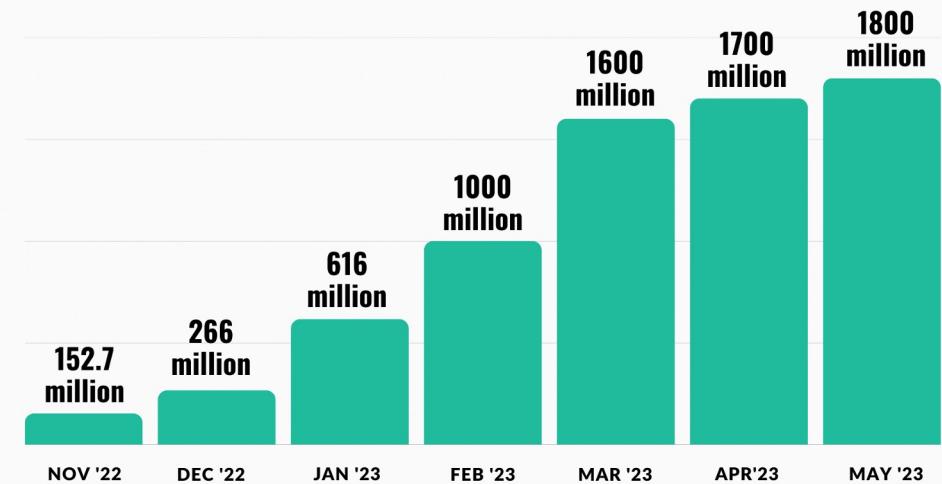


## Time to Reach 100M Users

Months to get to 100 million global Monthly Active Users



## ChatGPT Website All Time Views



Quelle: demandsage

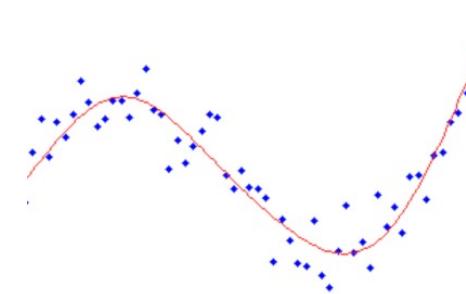
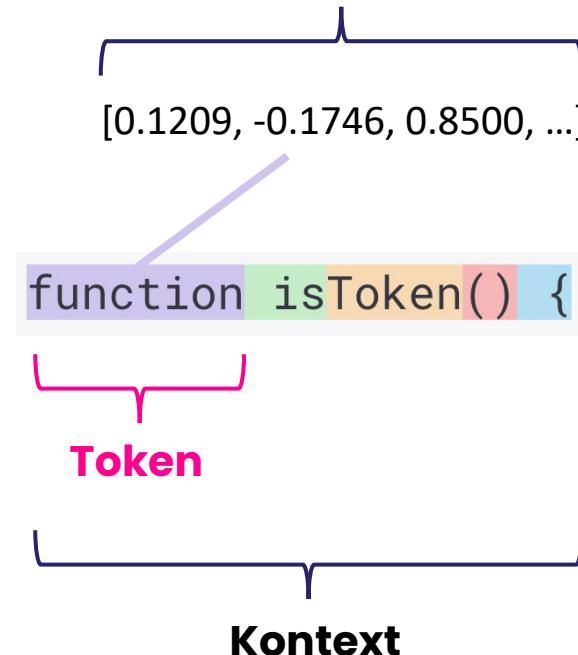


# Large Language Model (LLM) Basics

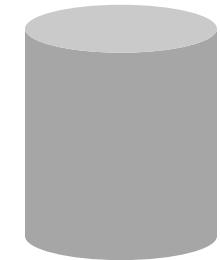
```
>'>).insertAfter(a)),l.call(this,a,c),a.css({this.vertical=new m(a,b),this.horizontal=new lPosition(),this.bar.addClass("scrolling"),rolling.bind(this),q>this,finishScrollMode:function(a){var b=this,d=this.drag.bindOn(),c.off("mousemove",d),b.bar.removeClass("entHeight"),this.context.scrollHeight>this.height*100+"%"},this}),drag:{value:function(a){text.s>Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat,
```

**Sequentielle  
Kontextabhängige  
Daten**

## Hochdimensionaler Vektor (Embedding)



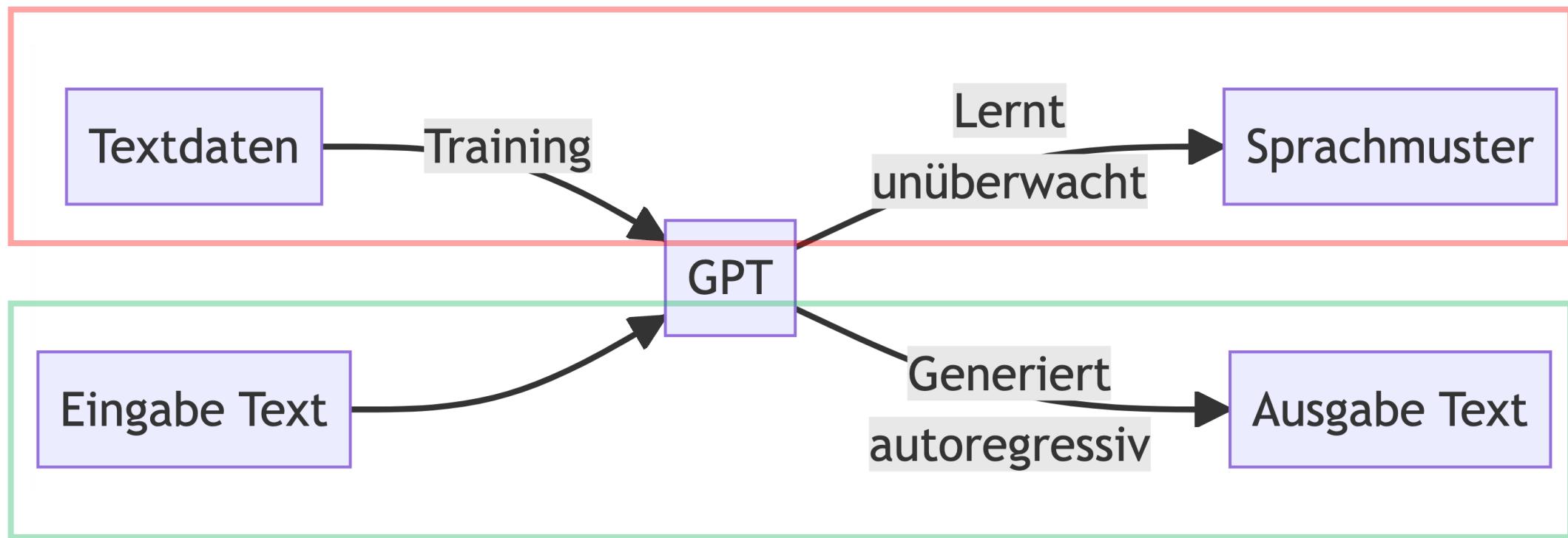
**Neuronales Netz**



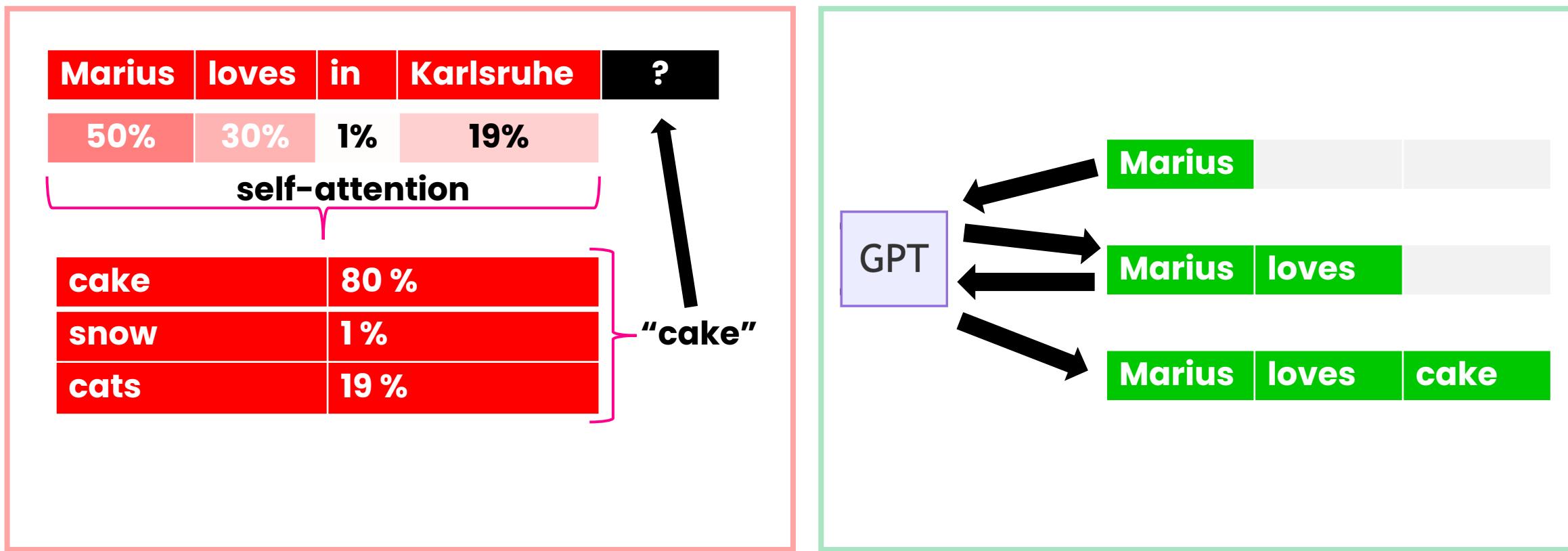
**Modell**



# Generative Pretrained Transformer (GPT)



# Generative Pretrained Transformer (GPT)



# ChatGPT != GPT

Step 1

**Collect demonstration data and train a supervised policy.**

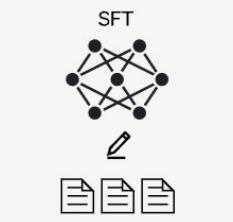
A prompt is sampled from our prompt dataset.



A labeler demonstrates the desired output behavior.



This data is used to fine-tune GPT-3.5 with supervised learning.



Step 2

**Collect comparison data and train a reward model.**

A prompt and several model outputs are sampled.



A labeler ranks the outputs from best to worst.



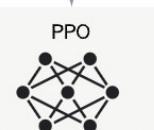
This data is used to train our reward model.



Step 3

**Optimize a policy against the reward model using the PPO reinforcement learning algorithm.**

A new prompt is sampled from the dataset.



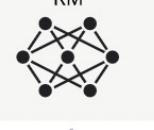
The PPO model is initialized from the supervised policy.



The policy generates an output.

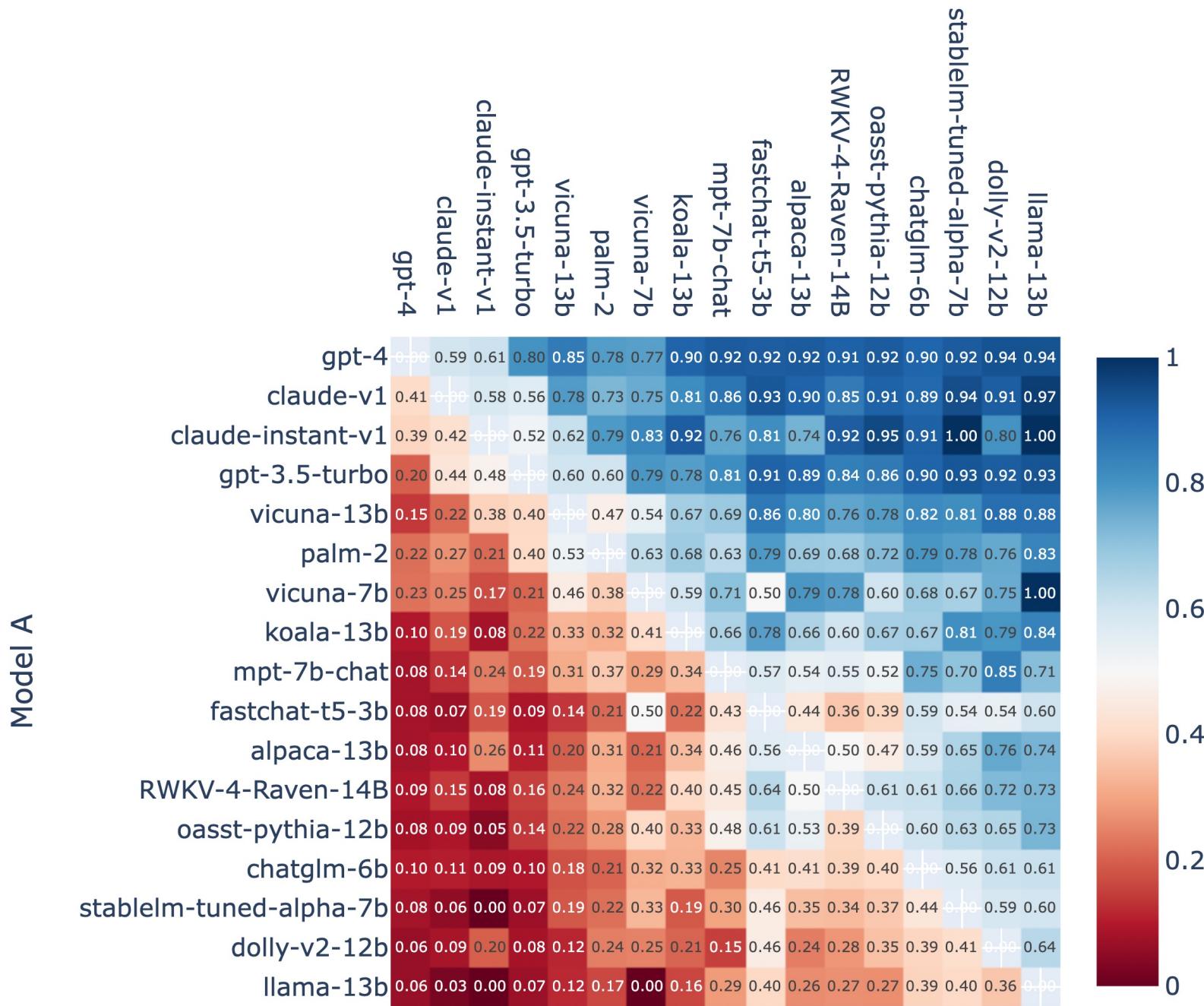


The reward model calculates a reward for the output.



The reward is used to update the policy using PPO.

Quelle: OpenAI





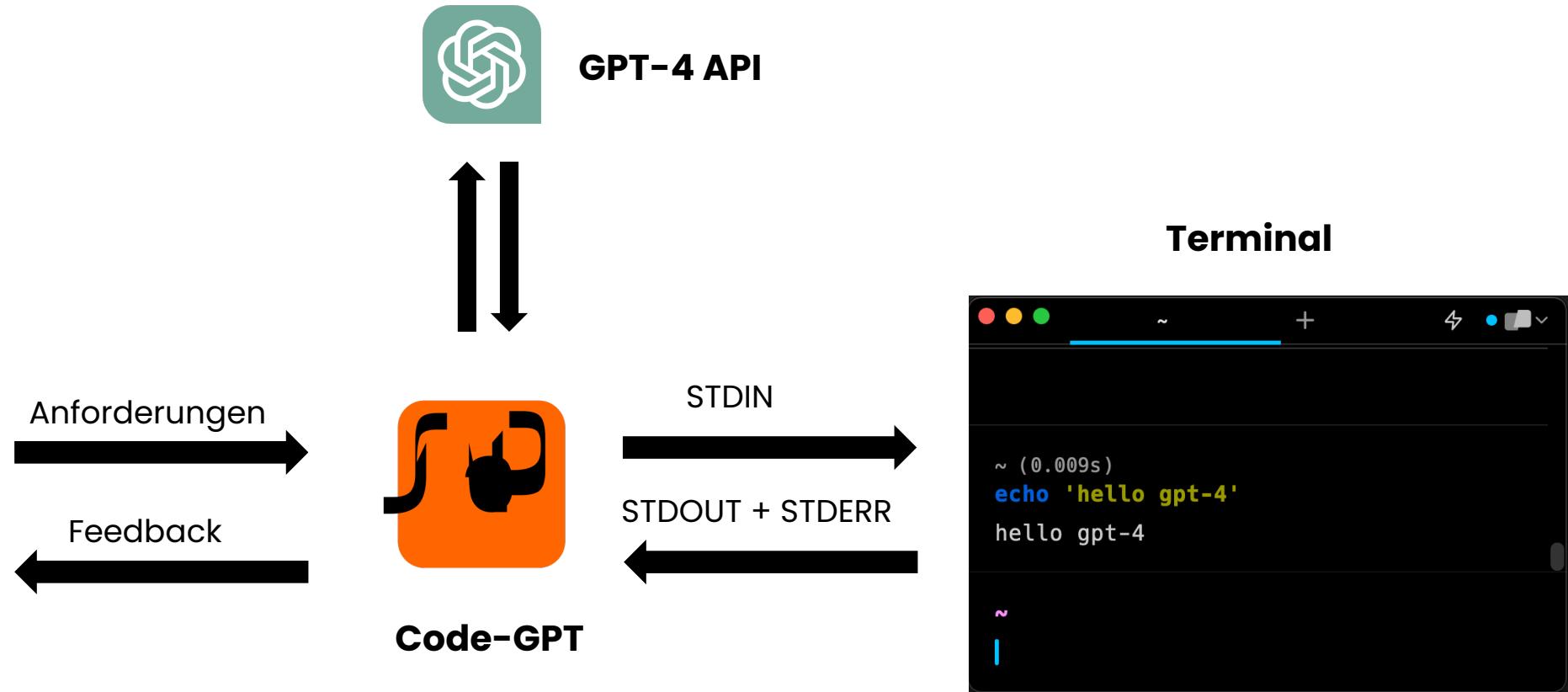
“...GPT-4 can use tools (including itself)...”

*Quelle: Sparks of Artificial General Intelligence: Early experiments with GPT-4*





Marius



<https://github.com/2mawi2/code-gpt-demo>





**GPT:** What software would you like me to create for you?



**Marius:** I need a password checker



**GPT:** What programming language should I use ?



**Marius:** Python !



**GPT:** What are the requirements for a valid password?



**Marius:** At least 6 digits, one letter and one number



**GPT:** > echo "def..." > password\_checker.py

```
def is_valid_password(password):
    if len(password) < 6:
        return False
    has_digit = False
    has_letter = False
    for char in password:
        if char.isdigit():
            has_digit = True
        elif char.isalpha():
            has_letter = True
    return has_digit and has_letter

password = input('Enter a password: ')

if is_valid_password(password):
    print('Valid password')
else:
    print('Invalid password')
```





**GPT:** I have created a Python script called password\_checker.py. Let's run the script to test it.



**GPT:** > python3 password\_checker.py



**Marius:** password -> "Invalid password"  
thisIsALongPassword!3 -> "Valid password"





**GPT:** Would you like to make any further changes?



**Marius:** I require unit tests



**GPT:** > echo "import unit..." > test\_password\_checker.py

```
import unittest

class TestPasswordChecker(unittest.TestCase):

    def test_valid_password(self):
        self.assertTrue(is_valid_password('a1b2c3'))

    def test_invalid_password(self):
        self.assertFalse(is_valid_password('abc'))
        self.assertFalse(is_valid_password('123456'))
        self.assertFalse(is_valid_password('abcdef'))

if __name__ == '__main__':
    unittest.main()
```





GPT:

```
> python3 -m unittest test_password_checker.py  
-----  
NameError: name 'is_valid_password' is not defined
```



GPT: An import is missing. Let me fix that and run the tests again.



GPT:

```
> echo "import unittest" > test_password_checker.py
```



GPT:

```
> python3 -m unittest test_password_checker.py  
-----  
OK
```





**GPT:** Are you satisfied with the password checker?



**Marius:** If user admin -> at least 10 digits



**GPT:** How should the program determine if the user is an admin?



**Marius:** I will pass a boolean

- Adapts code



GPT: > echo "def is\_valid\_password..." > password\_checker.py

- Adapts tests



GPT: > echo "from password\_checker impo..." > test\_password\_checker.py

- Runs tests



GPT: > python3 -m unittest test\_password\_checker.py

```
def is_valid_password(password, is_admin=False):
    if is_admin:
        min_length = 10
    else:
        min_length = 6

    if len(password) < min_length:
        return False

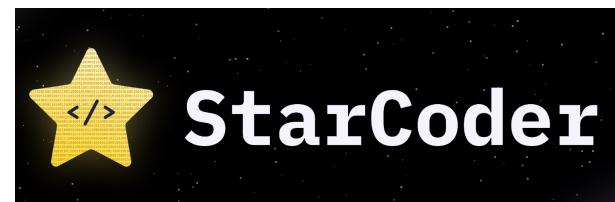
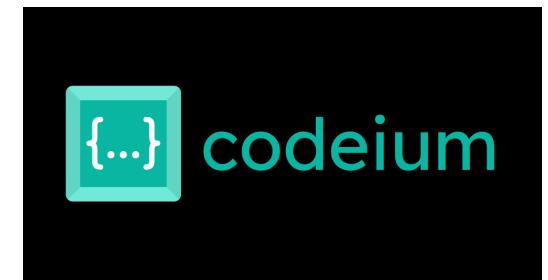
    has_digit = False
    has_letter = False
    for char in password:
        if char.isdigit():
            has_digit = True
        elif char.isalpha():
            has_letter = True

    return has_digit and has_letter
```





## Copilots



# Autocomplete

```
3 ~ def has_length(password, length):  
4 |     return len(password) >= length  
5  
6 def is_secure(password):  
7 |     return has_length(password, 8) and has_number(password)  
8  
9 ~ def has_number(password):  
10 |    return any(char.isdigit() for char in password)
```



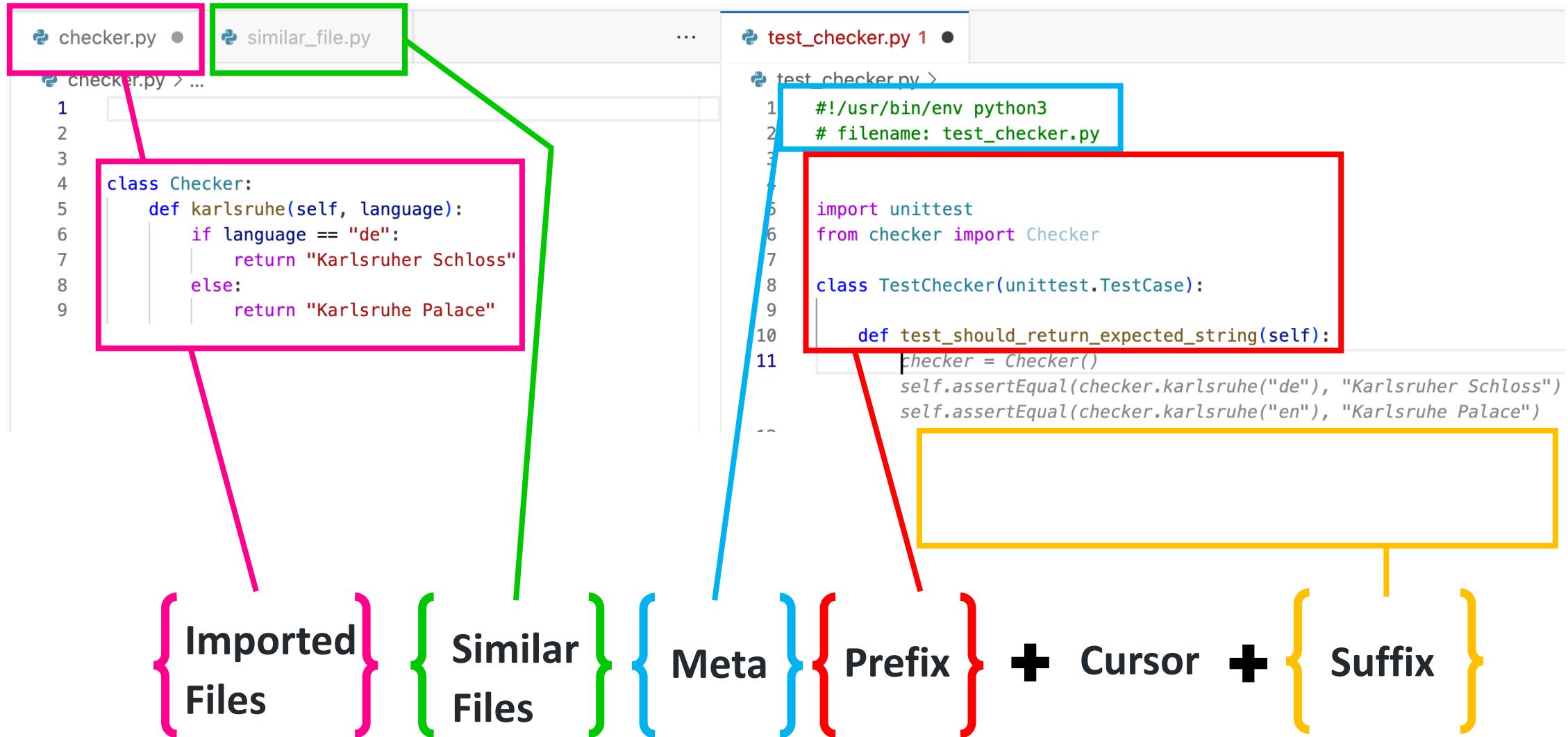
# Autocomplete

```
1  """
2  user json
3  {
4  |     "password": "string",
5  |     "username": "string",
6  }
7  """
8  def has_lower_case_password(user):
9  |     return any(c.islower() for c in user['password'])

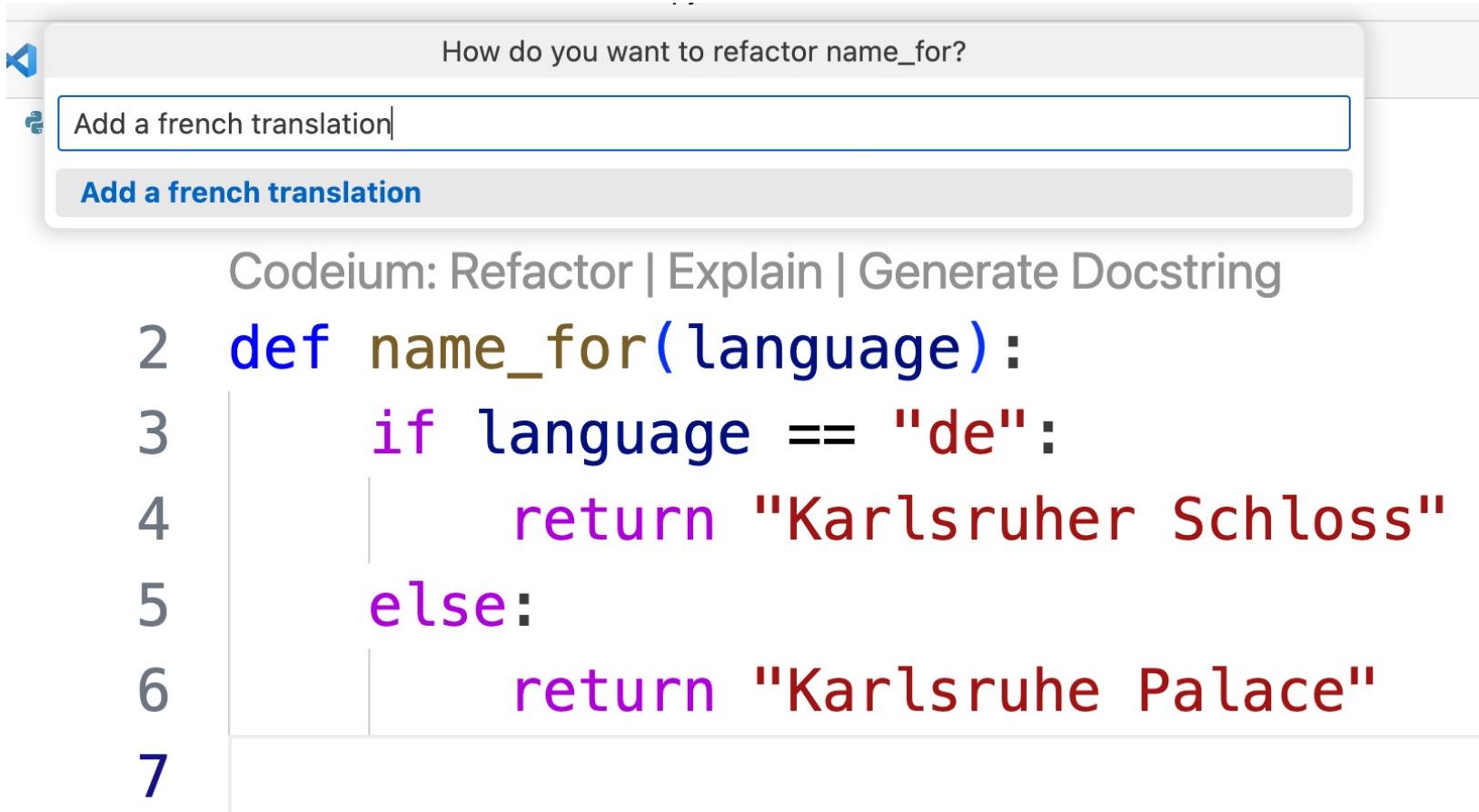
11 import re
12 def has_lower_case(password):
13 |     # Use regex
14 |     return re.search("[a-z]", password) is not None
```



# Autocomplete



# Chat



How do you want to refactor name\_for?

Add a french translation

Add a french translation

Codeium: Refactor | Explain | Generate Docstring

```
2 def name_for(language):
3     if language == "de":
4         return "Karlsruher Schloss"
5     else:
6         return "Karlsruhe Palace"
7
```



Refactor function: name\_for (checker.py:2:0-:6:33)

Add a french translation

Fri, Jun 9, 2023, 5:02 pm

MW

## Proposed Change

Replacing code at checker.py:2-6

 Apply Diff

Copy Code

```
def name_for(language):
    if language == "de":
        return "Karlsruher Schloss"
+   elif language == "fr":
+       return "Château de Karlsruhe"
    else:
        return "Karlsruhe Palace"
```

I added a new conditional statement to check if the input parameter is "fr". If it is, the function returns the translation for French. Otherwise, it returns the original translation in English.



Fri, Jun 9, 2023, 5:02 pm



Clear conversation

Type your question here...

Hit ⇧ + Enter to type a new line.

Submit Chat

# Chat

Accept  | Reject 

<<<<<<<< ✨ Codeium AI Suggestion >>>>>>

```
def name_for(language):
    if language == "de":
        return "Karlsruher Schloss"
+   elif language == "fr":
+       return "Château de Karlsruhe"
    else:
        return "Karlsruhe Palace"
```

<<<< bot-5daae50e-5854-46d6-aa7a-cd1eb12d107b

```
final class User {
    public String name;

    public User(String name) {
        this.name = name;
    }
}

class UserBuilder {
    private String name;

    public UserBuilder() {
    }

    public UserBuilder withName(String name) {
        this.name = name;
        return this;
    }

    public User build() {
        return new User(name);
    }
}
```

```
class App {
    fun bestCity(userName: String): String {
        return if (userName == "Marius") {
            "Frankfurt"
        } else {
            "Karlsruhe"
        }
    }
}

class AppTest {
    @Test
    fun `should return best city for Marius`() {
        assertEquals("Frankfurt", App().bestCity("Marius"))
    }

    @Test
    fun `should return best city for other users`() {
        assertEquals("Karlsruhe", App().bestCity("other"))
    }
}
```

```
3 var cities = listOf("Frankfurt", "Berlin", "Karlsruhe")
4
5 fun formatSortedByLengthWithOrderPrefix(cities: List<String>): String {
6     return cities
7         .sortedBy { it.length }
8         .mapIndexed { index, city -> "${index + 1}. $city" }
9         .joinToString("\n")
10
11 fun main(args: Array<String>) {
12     println(formatSortedByLengthWithOrderPrefix(cities))
13 }
```

```
> Task :app:run
1. Berlin
2. Frankfurt
3. Karlsruhe
```

BUILD SUCCESSFUL in 607ms

```
- name: MY_S3_ACCESSKEY
  valueFrom:
    secretKeyRef:
      name: my-s3-bucket
      key: s3_access_key
- name: MY_S3_SECRETKEY
  valueFrom:
    secretKeyRef:
      name: my-s3-bucket
      key: s3_secret_key
```

restartPolicy: OnFailure





## Technische Grenzen



Leistung



Kontext



Genauigkeit





## Rechtliche Blocker



Copywrite



Datenschutz



Tim Davis  
@DocSparse

GitHub Copilot - javastuff - Visual Studio Code

File Go Run Terminal Help

Untitled-1.cpp 5 • void cs\_gaxpy (const cs \*A, const double Untitled-1 • // sparse matrix add Untitled-2 • ...

updateDaysBetween... 5

Terminal

```
#include "cs.h"
/* C = alpha*A + beta*B */
cs *cs_add (const cs *A, const cs *B, double alpha, double beta)
{
    csi p, j, nz = 0, anz, *Cp, *Ci, *Bp, m, n, bnz, *w, values ;
    double *x, *Bx, *Cx ;
    cs *C ;
    if (!CS_CSC (A) || !CS_CSC (B)) return (NULL) ;           /* check inputs */
    if (A->m != B->m || A->n != B->n) return (NULL) ;
    m = A->m ; anz = A->p [A->n] ;
    n = B->n ; Bp = B->p ; Bx = B->x ; bnz = Bp [n] ;
    w = cs_malloc (m, sizeof (csi)) ;                         /* get workspace */
    values = (A->x != NULL) && (Bx != NULL) ;
    x = values ? cs_malloc (m, sizeof (double)) : NULL ;      /* get workspace */
    C = cs_salloc (m, n, anz + bnz, values, 0) ;             /* allocate result*/
    if (!C || !w || (values && !x)) return (cs_done (C, w, x, 0)) ;
    Cp = C->p ; Ci = C->i ; Cx = C->x ;
    for (j = 0 ; j < n ; j++)
    {
        Cp [j] = nz ;                                     /* column j of C starts here */
        nz = cs_scatter (A, j, alpha, w, x, j+1, C, nz) ; /* alpha*A(:,j)*/
        nz = cs_scatter (B, j, beta, w, x, j+1, C, nz) ; /* beta*B(:,j) */
        if (values) for (p = Cp [j] ; p < nz ; p++) Cx [p] = x [Ci [p]] ;
    }
    Cp [n] = nz ;                                     /* finalize the last column of C */
    cs_srealloc (C, 0) ;                            /* remove extra space from C */
    return (cs_done (C, w, x, 1)) ;                  /* success; free workspace, return C */
}
```

GitHub Copilot x

```
142 =====
143 =====
144 Accept Solution
145 cs_add (const cs *A, const cs *B, double alpha, double beta)
146 {
147     int p, j, *Cp, *Ci, m, n, [anz, *Bp, *Bi, bnz, *w, *x, nz0 ;
148     double *Cx, *Bx, *x ;
149     cs *C ;
150     if (!CS_CSC (A) || !CS_CSC (B)) return (NULL) ;           // check inputs
151     m = A->m ; n = A->n ; anz = A->p [n] ;
152     Bp = B->p ; Bi = B->i ; Bx = B->x ; bnz = Bi [n] ;
153     if (m != B->m || n != B->n) return (NULL) ;           // check dimensions
154     | = cs_salloc (m, n, anz + bnz, A-> != NULL, 0) ;       // alloc result
155     w = cs_malloc (n, sizeof (int)) ;                         // get workspace
156     x = (A->x != NULL) ? cs_malloc (n, sizeof (double)) : NULL ; // get workspace
157     if (!C || !w || !x) return (cs_done (C, w, x, 0)) ;       // out of memory
158     Cp = C->p ; Ci = C->i ; Cx = C->x ;
159     for (j = 0 ; j < n ; j++)
160     {
161         Cp [j] = Cx ;                                     // column j of C starts here
162         nz = cs_scatter (A, j, alpha, w, x, j 1, C, nz) ; // alpha*A(:,j)
163         nz = cs_scatter (B, j, beta, w, x, j[1], C, nz) ; // beta*B(:,j)
164     }
165     Cp [nn = nz] ;                                     // finalize the last column
166     cs_srealloc (C, 0) ;                            // remove extra space from C
167     return (cs_done (C, w, x, 1)) ;                  // success; free workspace, r
168 }
169
170 // sparse matrix axpy
171 cs_axpy (const cs *A, double alpha, zconst cs *B, double beta)
172 {
173     int
174 =====
175 Accept Solution
176 cs_add+ (const cs *], const cs *B, double alpha, double beta)
177 {
178     int p, j, *Cp, *Ci, m, n, anz, *Bp, bnz, *w ;
179     double *Cx, *Bx ;
180     cs *C ;
181     if (!CS_CSC (A) || !CS_CSC (B)) return (NULL) ;           // check inputs
182     m = A->m ; anz = A->p [A->n] ;
183     n = A->n ; Bp = B->p ; bnz = Bp [B->n] ;
```

Quelle: Twitter

```

// sparse matrix transpose in CSC format
// (9 lines of comments omitted here)
#include "cs.h"
cs *cs_transpose (const cs *A, int values)
{
    int p, q, j, *ATp, *ATi, *w, m, n, *Ap, *Ai ;
    double *ATx, *Ax ;
    cs *AT ;
    if (!CS_CSC (A)) return (NULL) ;      // check inputs
    m = A->m ; n = A->n ; Ap = A->p ; Ai = A->i ; Ax = A->x ;
    AT = cs_salloc (n, m, Ap [n], values && Ax != NULL, 0) ; // allocate result
    w = cs_calloc (m, sizeof (int)) ;           // get workspace
    if (!AT || !w) return (cs_done (AT, NULL, w, 0)) ; // out of memory
    ATp = AT->p ; ATI = AT->i ; ATx = AT->x ;
    for (p = 0 ; p < Ap [n] ; p++) w [Ai [p]]++ ;           // row counts
    cs_cumsum (ATp, w, m) ;                         // row pointers
    for (j = 0 ; j < n ; j++)
    {
        for (p = Ap [j] ; p < Ap [j+1] ; p++)
        {
            ATI [q = w [Ai [p]]++] = j ;           // AT(i,j) is the pth entry in A
            if (values) ATx [q] = Ax [p] ;
        }
    }
    return (cs_done (AT, NULL, w, 1)) ;      // success; free workspace, return result
}

```

# Save bandwidth by using a torrent to distribute more efficiently #73

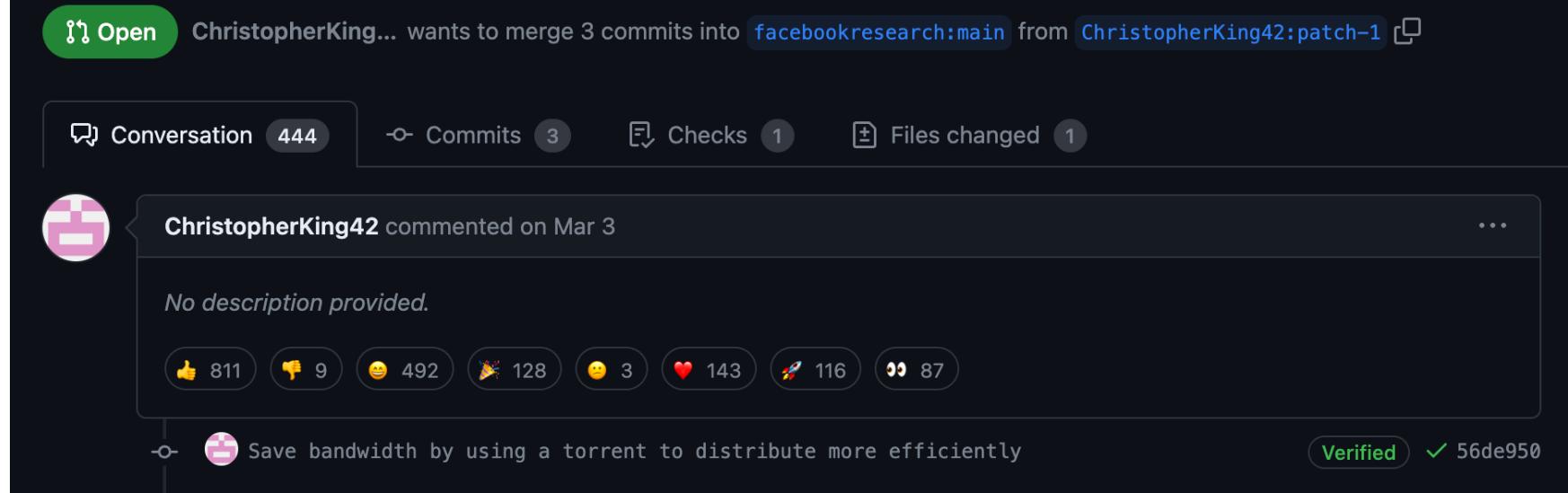
[↑ Open](#) ChristopherKing... wants to merge 3 commits into `facebookresearch:main` from `ChristopherKing42:patch-1`

Conversation 444 Commits 3 Checks 1 Files changed 1

ChristopherKing42 commented on Mar 3  
*No description provided.*

811 9 492 128 3 143 116 87

Save bandwidth by using a torrent to distribute more efficiently ✓ 56de950



2 README.md

... ... @@ -1,7 +1,7 @@

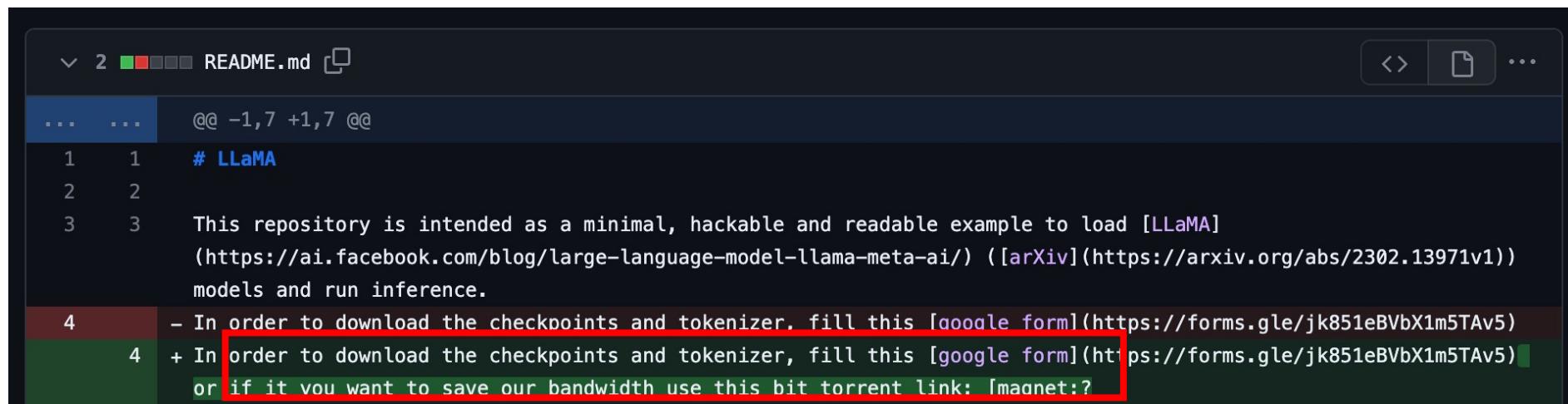
1 1 # LLaMA

2 2

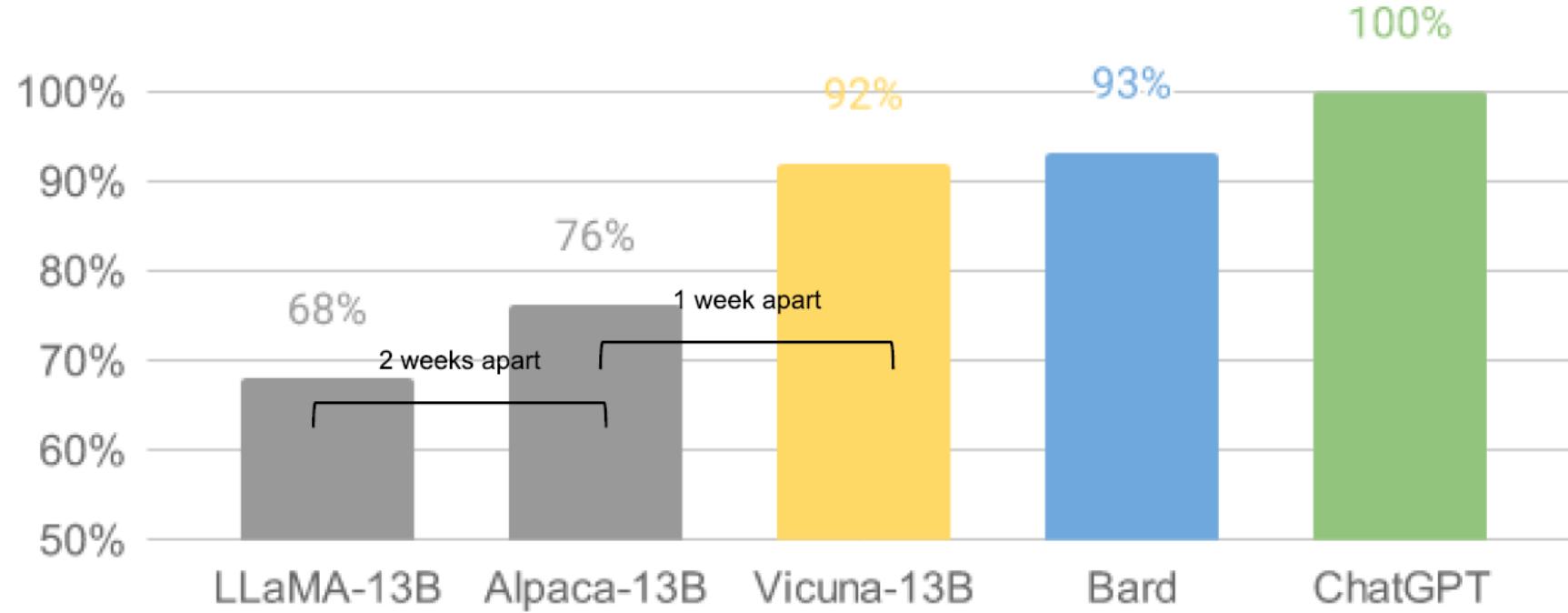
3 3 This repository is intended as a minimal, hackable and readable example to load [LLaMA]  
(<https://ai.facebook.com/blog/large-language-model-llama-meta-ai/>) ([arXiv](<https://arxiv.org/abs/2302.13971v1>))  
models and run inference.

4 - In order to download the checkpoints and tokenizer, fill this [google form](<https://forms.gle/jk851eBVbX1m5TAv5>)

4 + In order to download the checkpoints and tokenizer, fill this [google form](<https://forms.gle/jk851eBVbX1m5TAv5>) or if you want to save our bandwidth use this bit torrent link: [magnet:?]



# Google: “We Have No Moat, And Neither Does OpenAI”

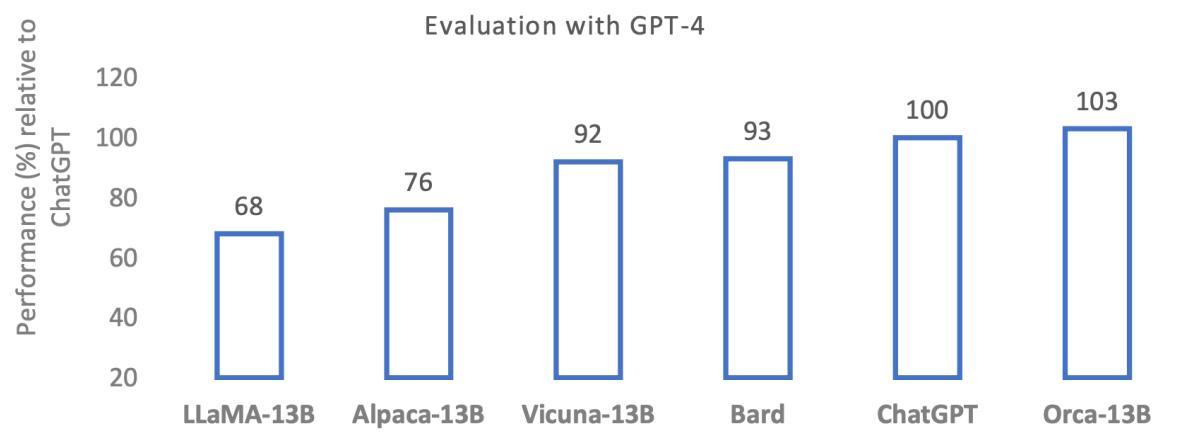


\*GPT-4 grades LLM outputs. Source: <https://vicuna.lmsys.org/>

5. Juni, 2023



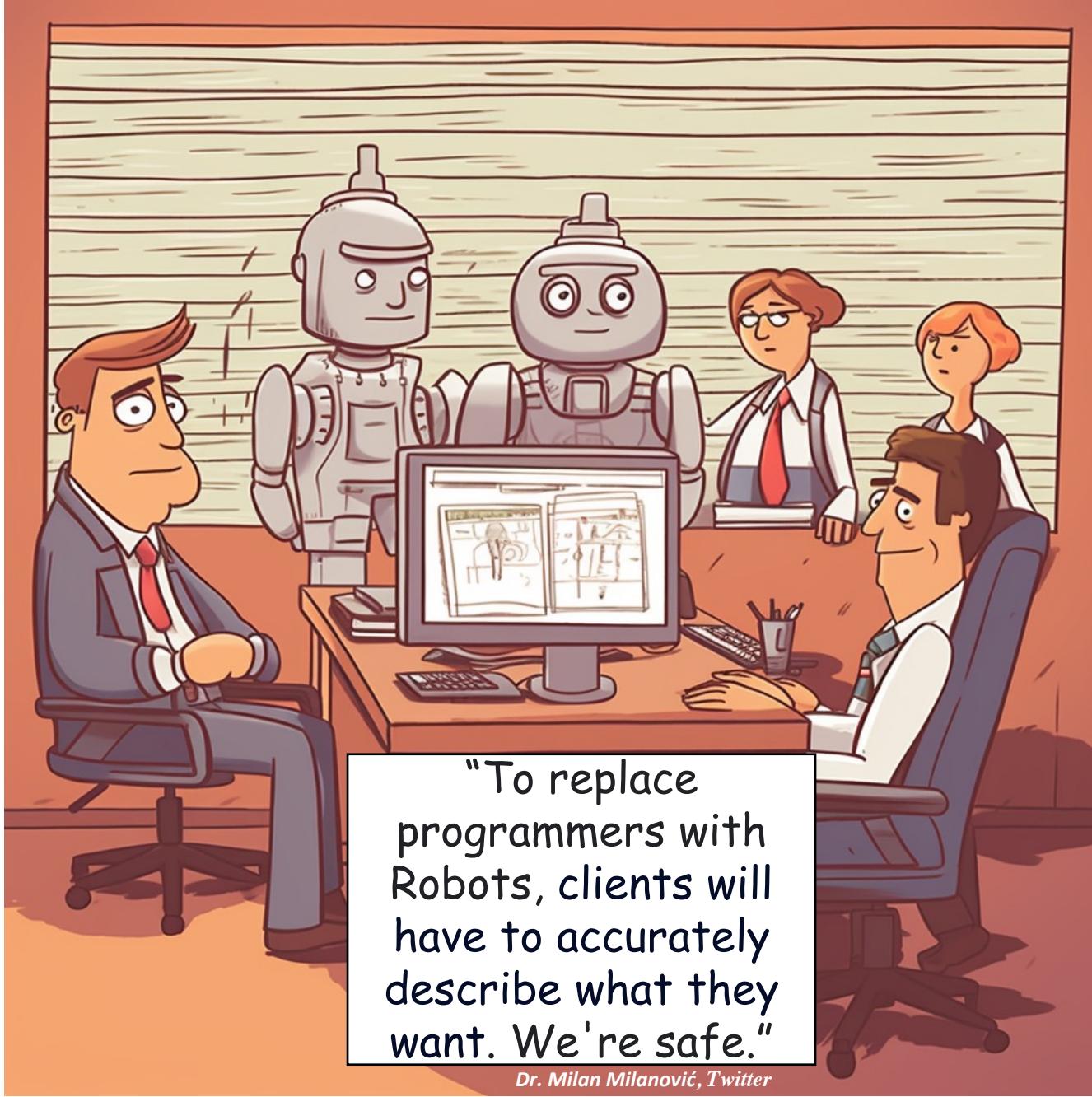
## Orca: Progressive Learning from Complex Explanation Traces of GPT-4



4. May, 2023



Model	HumanEval	MBPP
LLaMA-7B	10.5	17.7
LaMDA-137B	14.0	14.8
LLaMA-13B	15.8	22.0
CodeGen-16B-Multi	18.3	20.9
LLaMA-33B	21.7	30.2
CodeGeeX	22.9	24.4
LLaMA-65B	23.7	37.7
PaLM-540B	26.2	36.8
CodeGen-16B-Mono	29.3	35.3
StarCoderBase	30.4	49.0
code-cushman-001	33.5	45.9
StarCoder	33.6	52.7
StarCoder-Prompted	40.8	49.5



"To replace  
programmers with  
Robots, clients will  
have to accurately  
describe what they  
want. We're safe."

*Dr. Milan Milanović, Twitter*



# Effekte



**Produktivität**



**Excel-Effekt**



**Tests**



**Dokumentation**

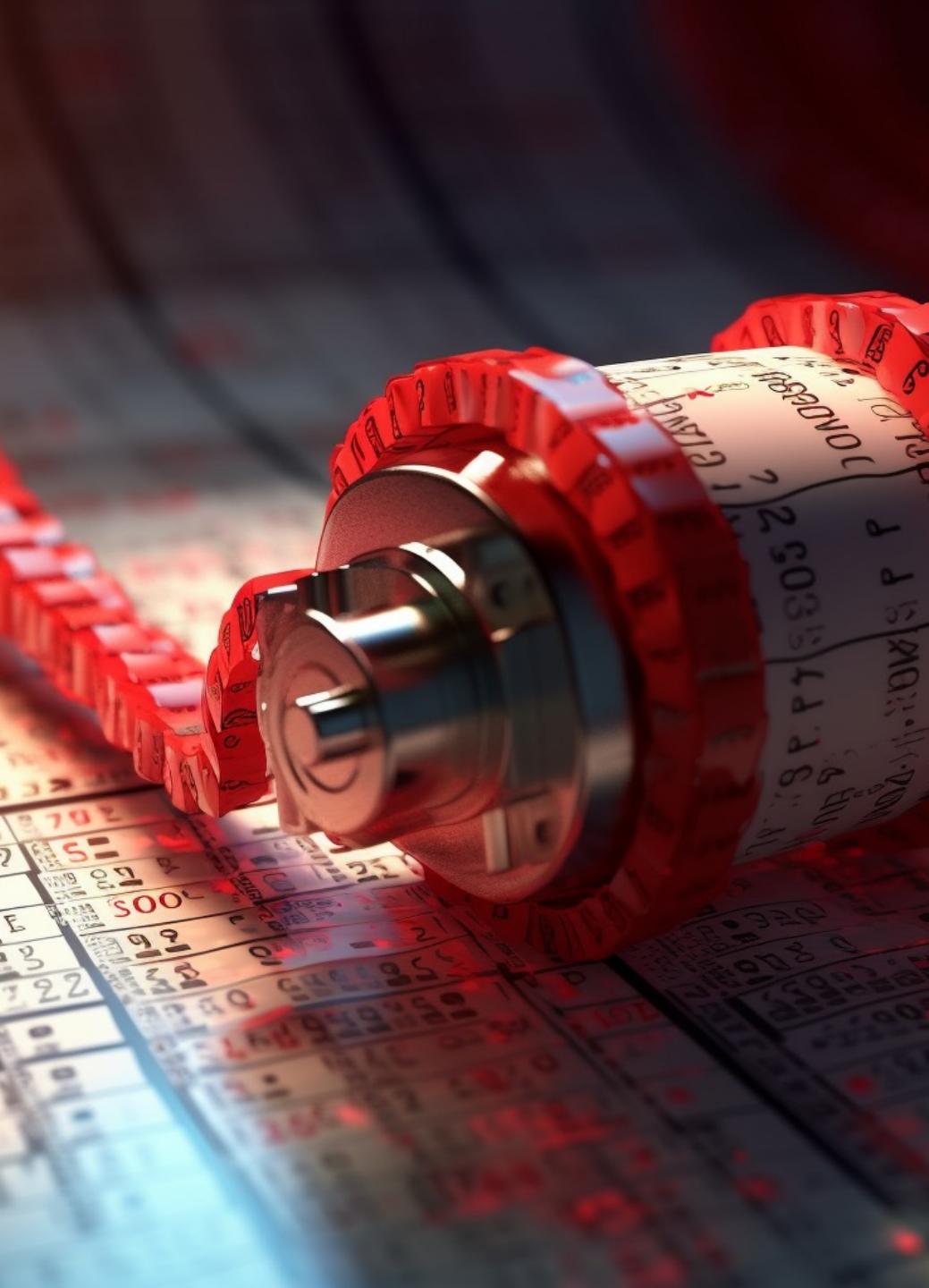


**AI-Pair-Programming**



**Tooling**





# Herausforderung



**Recruiting**



**Software Qualität**



**Kognitiver Einfluss**



**Bias**



# Quellen

<https://www.actuaries.digital/2018/09/05/history-of-ai-winters/>

<https://www.demandsage.com/chatgpt-statistics/>

[https://commons.wikimedia.org/wiki/File:Orangutan\\_using\\_precision\\_grip.jpg](https://commons.wikimedia.org/wiki/File:Orangutan_using_precision_grip.jpg)

<https://thakkarparth007.github.io/copilot-explorer/posts/copilot-internals>

[https://sinews.siam.org/Details-Page/ethical-concerns-of-code-generation-through-artificial-intelligence#disqus\\_thread](https://sinews.siam.org/Details-Page/ethical-concerns-of-code-generation-through-artificial-intelligence#disqus_thread)

<https://twitter.com/DocSparse>

<https://www.semianalysis.com/p/google-we-have-no-moat-and-neither>

<https://openai.com/blog/chatgpt>

<https://chat.lmsys.org/?arena>

[https://twitter.com/milan\\_milanovic/status/1622127889108926466?lang=en](https://twitter.com/milan_milanovic/status/1622127889108926466?lang=en)

<https://www.demandsage.com/chatgpt-statistics/>

