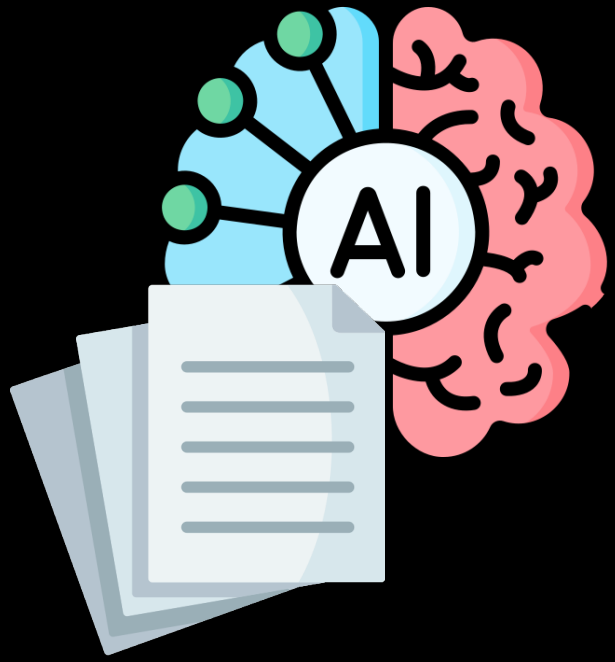
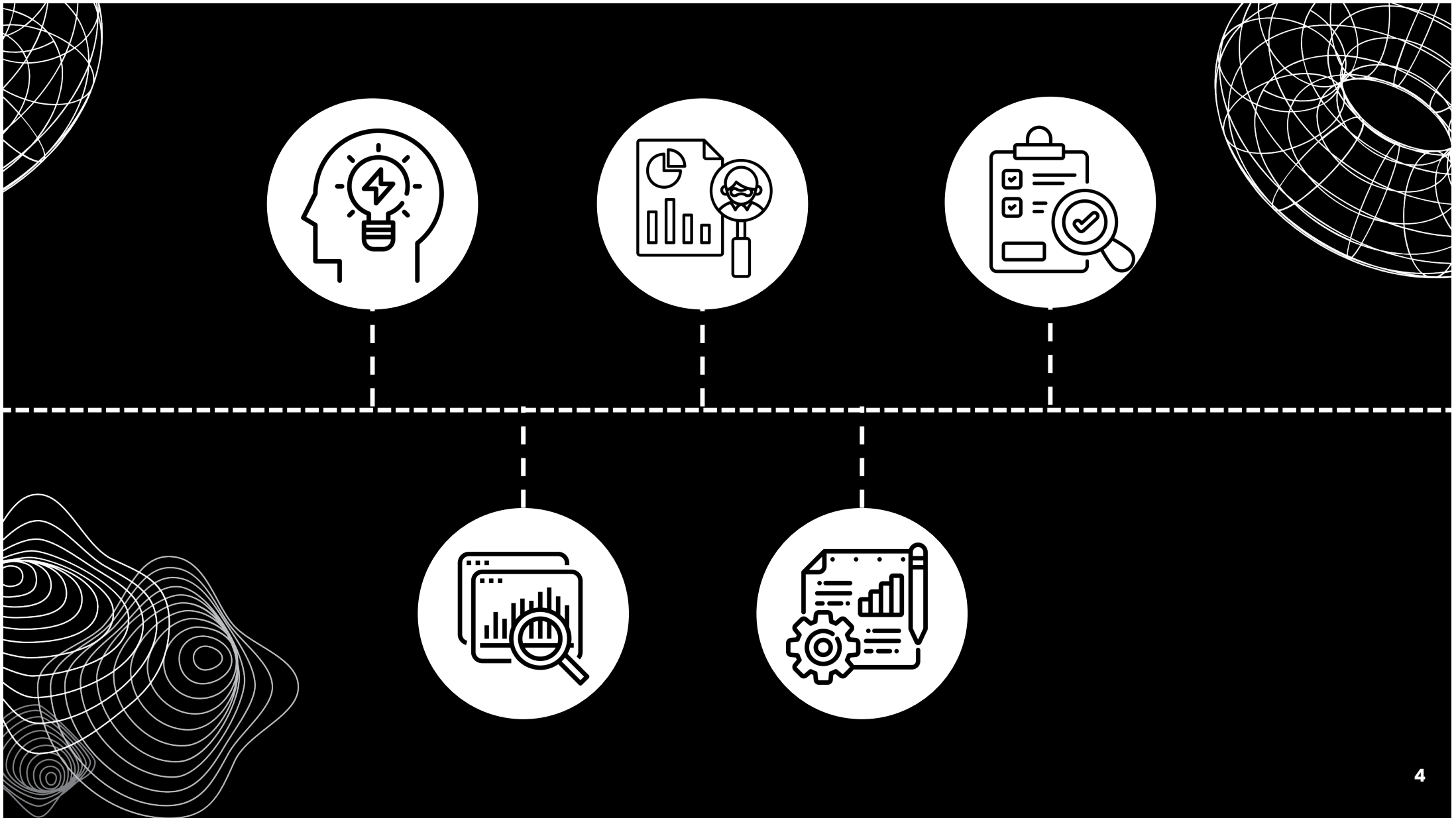
Personenbezogene Daten in LLms

Maximilian Werzinger


AI

- Namen
- E-Mail-Adressen
- Telefonnummern
- Kundennummern



Was sind personenbezogene Daten?

„Personenbezogene Daten sind alle Infos, die sich auf eine identifizierte oder identifizierbare Person beziehen.“ [Q1]

Person ist direkt erkennbar
→ *Max Mustermann, Musterstraße 1, 12345 Musterstadt.*

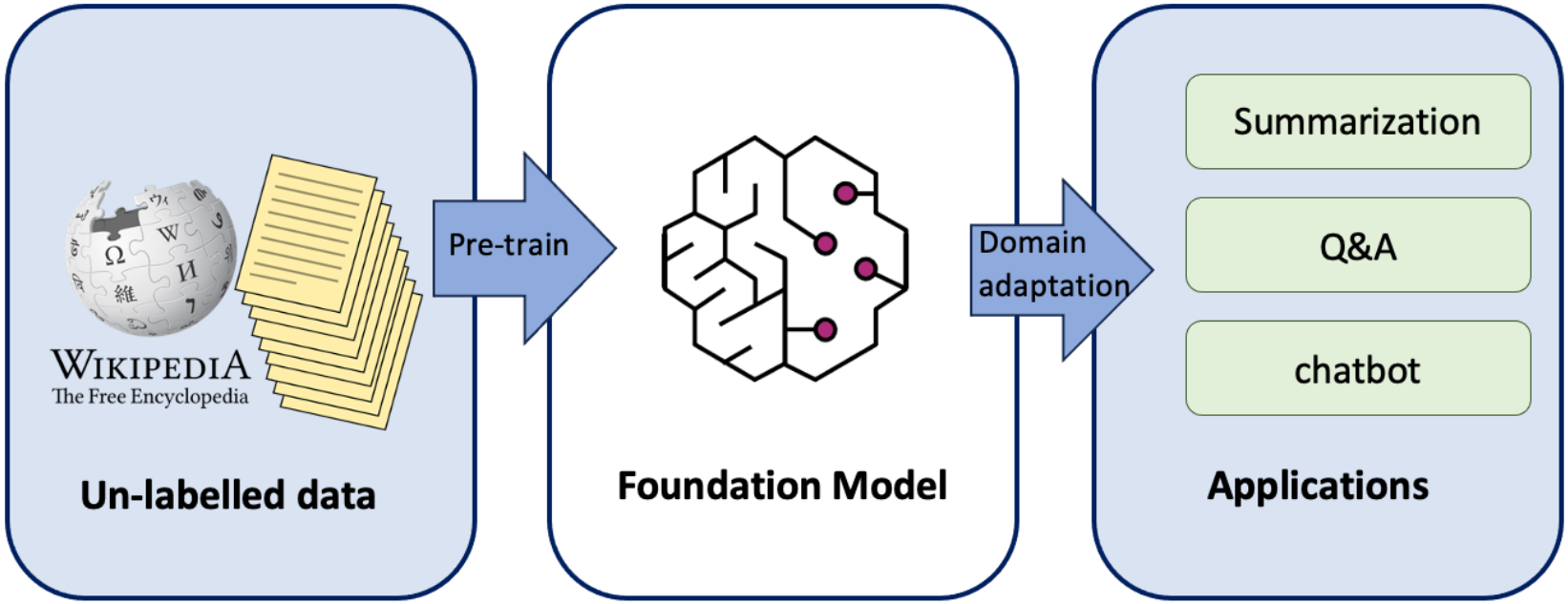
Person wird erst durch Kombination von Infos erkennbar
→ z.B. „Data Scientist in Firma X in München, Jahrgang 1999“.

Was sind personenbezogene Daten?

- Name, Anschrift
- E-Mail-Adresse, Telefonnummer
- Ausweis-, Kundenummer, Kontonummer

- Gesundheitsdaten
- Biometrische/genetische Daten
- Politische Meinung, Religion/Weltanschauung
- Gewerkschaftszugehörigkeit, sexuelle Orientierung

- IP-Adresse, Cookie-ID, Geräte-ID, Browser-Fingerprint
- Standortdaten (GPS, Wohnort + Postleitzahl)
- Kombinationen wie *Jobtitel + Firma + Stadt*


WIKIPEDIA
The Free Encyclopedia
Un-labelled data
Pre-train
Foundation Model
Domain adaptation
Summarization
Q&A
chatbot
Applications

[Q1]

Wie gelangen PII in LLM-Trainingsdaten?

LLMs werden auf riesigen Textmengen aus dem Web, Büchern, Foren, Code usw. trainiert.

Web-Scraping erfasst dabei auch personenbezogene Daten:

- öffentliche Profile, Blogs, Foren, Social Media
- GitHub-Commits, Issue-Tracker
- Datenleaks, Pastebin & Co.
- Lebensläufe, Berichte, Gerichtsurteile

Im Unternehmen:

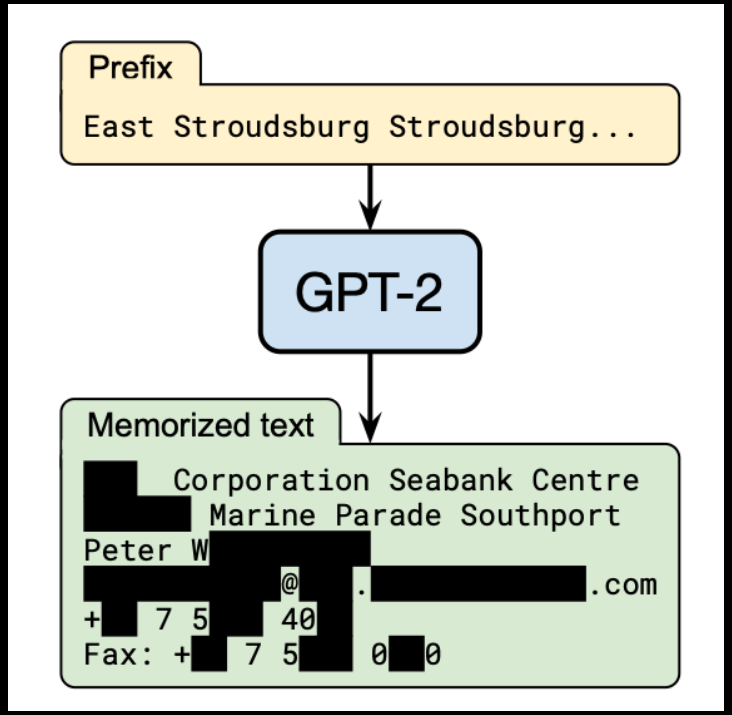
- Logs
- Tickets
- Internet Dokumente


usenix
THE ADVANCED
COMPUTING SYSTEMS
ASSOCIATION
Extracting Training Data from Large Language Models
Nicholas Carlini, Google; Florian Tramèr, Stanford University; Eric Wallace,
UC Berkeley; Matthew Jagielski, Northeastern University; Ariel Herbert-Voss,
OpenAI and Harvard University; Katherine Lee and Adam Roberts, Google;
Tom Brown, OpenAI; Dawn Song, UC Berkeley; Úlfar Erlingsson, Apple;
Alina Oprea, Northeastern University; Colin Raffel, Google

[Q2]

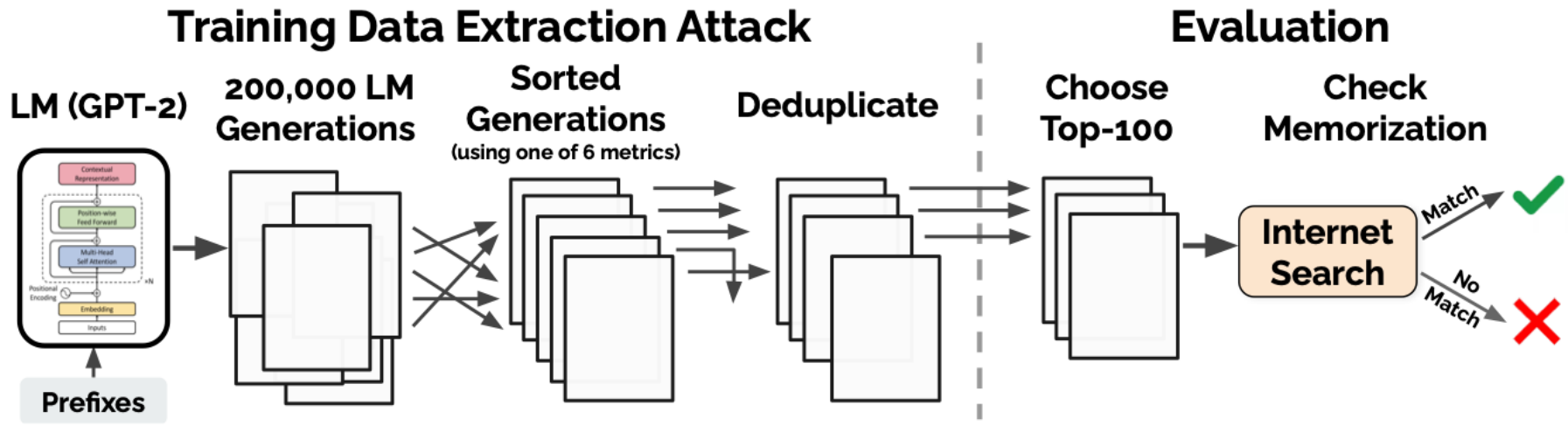
Fallstudie: Extracting Trainign Data From Large Language Models

- GPT-2 (1,5Milli. Parameter) auf Web-Scrapes trainiert
- Forschende extrahieren hunderte wortwörtliche Trainingssätze
- Darunter reale PII: Namen, Telefonnummern, E-Mails, Adressen
- Teilweise stammen die Sequenzen aus nur einem einzigen Dokument im Training



Beispiel einer PII-Extraktion aus GPT-2 [Q2]

Fallstudie: Extracting Trainign Data From Large Language Models



[Q2]

Angriffsarten auf LLMs – Angriffe auf PII

[Q3]

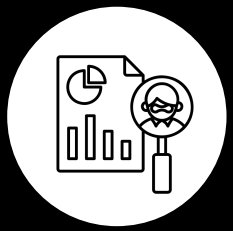
1. Training-Data-Extraction
   - Gezielte Rekonstruktion von Trainingsdaten

2. Prompt Hacking / Jailbreaks
   - Sicherheitsfilter umgehen, um sensible Antworten zu erzwingen

3. Membership Inference & Profiling
   - Herausfinden, ob ein bestimmtes Beispiel im Training war

- PII sind in Trainings- und Eingabedaten weit verbreitet
- LLms können einzelne Beispiele memorisieren
- Angriffe können diese Informationen wieder sichtbar machen

PII sollten vor der Verarbeitung erkannt und geschützt werden



Organisatorischen Schutzmaßnahmen

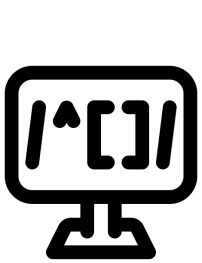
Technischen Schutzmaßnahmen

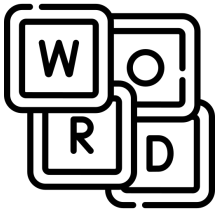
Rechtlichen Schutmaßnahmen

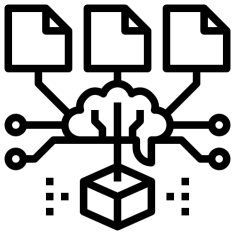
Microsoft Presidio

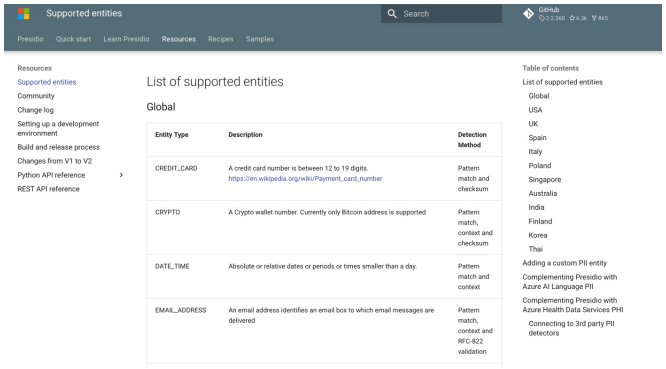
- Open-Source-SDK von Microsoft zur Erkennung & Anonymisierung sensibler Daten.
- Unterstützt Text, Bilder und strukturierte Daten.
- Zwei Kernkomponenten:
  - Analyzer-Engine → findet PII
  - Anonymizer-Engine → maskiert/ersetzt gefundene PII

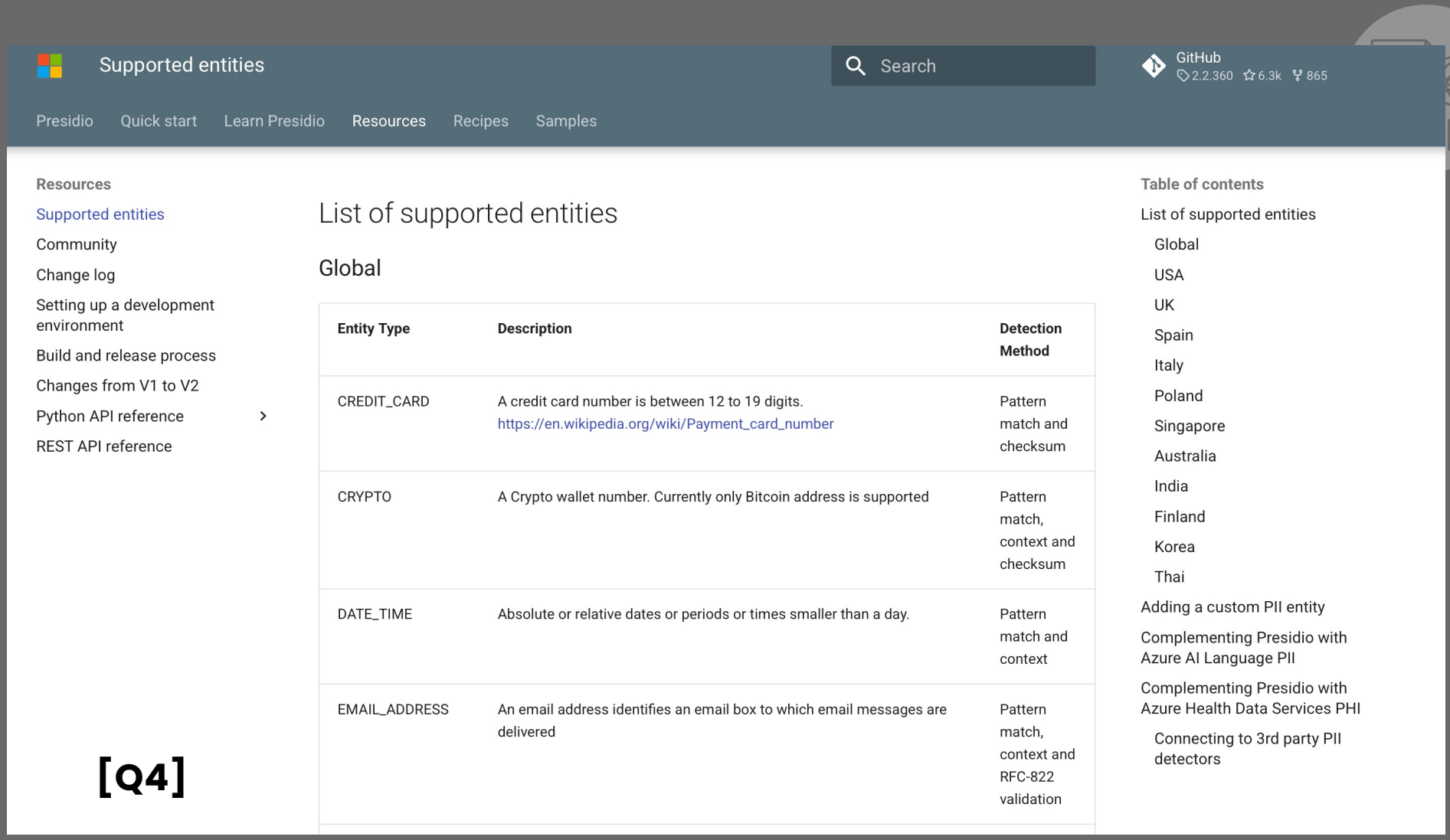
Analyzer




W
R
D

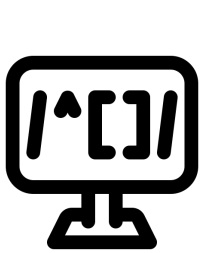


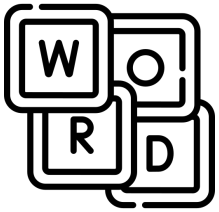

Supported entities
Search
GitHub
Presidio
Quick start
Learn Presidio
Resources
Recipes
Samples
Resources
Supported entities
Community
Change log
Setting up a development environment
Build and release process
Changes from V1 to V2
Python API reference
REST API reference
List of supported entities
Global
Entity Type
Description
Detection Method
CREDIT_CARD
A credit card number is between 12 to 19 digits.
https://en.wikipedia.org/wiki/Payment_card_number
Pattern match and checksum
CRYPTO
A Crypto wallet number. Currently only Bitcoin address is supported
Pattern match, context and checksum
DATE_TIME
Absolute or relative dates or periods or times smaller than a day.
Pattern match and context
EMAIL_ADDRESS
An email address identifies an email box to which email messages are delivered
Pattern match, context and RFC-822 validation
Table of contents
List of supported entities
Global
USA
UK
Spain
Italy
Poland
Singapore
Australia
India
Finland
Korea
Thai
Adding a custom PII entity
Complementing Presidio with Azure AI Language PII
Complementing Presidio with Azure Health Data Services PHI
Connecting to 3rd party PII detectors

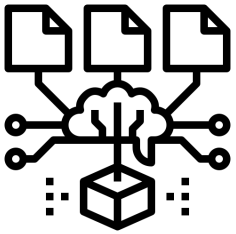

Supported entities
Search
GitHub
2.2.360
6.3k
865
Presidio
Quick start
Learn Presidio
Resources
Recipes
Samples
Resources
Supported entities
Community
Change log
Setting up a development environment
Build and release process
Changes from V1 to V2
Python API reference
REST API reference
List of supported entities
Global
Entity Type
Description
Detection Method
CREDIT_CARD
A credit card number is between 12 to 19 digits. https://en.wikipedia.org/wiki/Payment_card_number
Pattern match and checksum
CRYPTO
A Crypto wallet number. Currently only Bitcoin address is supported
Pattern match, context and checksum
DATE_TIME
Absolute or relative dates or periods or times smaller than a day.
Pattern match and context
EMAIL_ADDRESS
An email address identifies an email box to which email messages are delivered
Pattern match, context and RFC-822 validation
Table of contents
List of supported entities
Global
USA
UK
Spain
Italy
Poland
Singapore
Australia
India
Finland
Korea
Thai
Adding a custom PII entity
Complementing Presidio with Azure AI Language PII
Complementing Presidio with Azure Health Data Services PHI
Connecting to 3rd party PII detectors

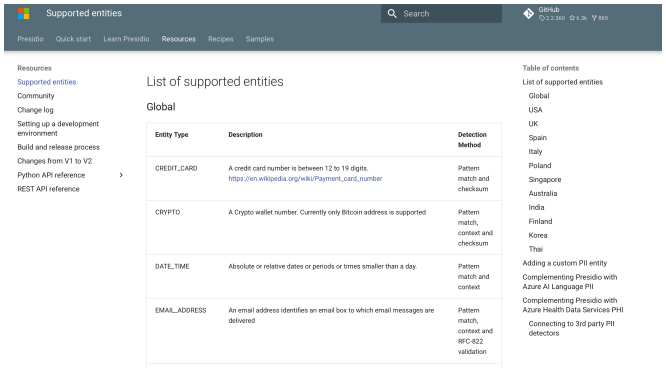
[Q4]

Analyzer









PII-Typ

Start/End-Index

Score

Anonymizer

- **Redact**
  - Max Mustermann → ▉ oder gar nichts
- **Replace**
  - Max Mustermann → [PERSON]
- **Mask**
  - max.mustermann@example.com→ m*********@example.com
- **Hash/Encrypt**
  - ID 12345678 → bca896f7a640039367c209c2441a18b0

Presidio Detection Flow

[Q5]

Regex
pattern recognition

NER(ML)*
leveraging natural language to detect entities

Checksum
validate patterns (if applicable)

Context Words
increase the detection confidence

Anonymization
multiple anonymization techniques

*NER – Named Entity Recognition

Projekt – Ziele, Artefakt & Praxis-Levels

- Redaktion-Script für personenbezogene Daten vor LLM-Verarbeitung
- Vor/Nach-Vergleich der LLM-Antwort
- Messung von:
  - Antwortqualität
  - Datenschutz

Regel-/NER-basierte Erkennung & Maskierung

Analyse von Falsch-Positiven/-Negativen auf 50 Beispielen

Selektive Demaskierung (post-hoc) mit Audit-Log

Redaktion-Script

```
EMAIL = re.compile(r"[a-zA-Z0-9_.+-]+@[a-zA-Z0-9-]+\.[a-zA-Z0-9-.]+")
ID = re.compile(r"(ID[-:]?\s?\d{3,}|TK-\d{3,}|\d{4}-LLM)")
NAME = re.compile(r"\b[A-Z][a-z]+\s[A-Z][a-z]+\b")
```

```
def mask_pii(text: str) -> Tuple[str, Dict[str, int]]:
    counts = Counter()
    def repl_email(m):
        counts['email'] += 1
        return '[EMAIL]'
    def repl_id(m):
        counts['id'] += 1
        return '[ID]'
    def repl_name(m):
        counts['name'] += 1
        return '[NAME]'
    text = EMAIL.sub(repl_email, text)
    text = ID.sub(repl_id, text)
    text = NAME.sub(repl_name, text)
    return text, counts
```

Redaktion-Script

```
Original : Anna Schmidt (anna.schmidt@example.com) bestellte Laptop ID-99123.
Maskiert : [NAME] ([EMAIL]) bestellte Laptop [ID].
Gefunden : {'email': 1, 'id': 1, 'name': 1}

Original : Kontakt: Max Bauer, max.bauer@firma.de, Ticket ID: TK-4455.
Maskiert : Kontakt: [NAME], [EMAIL], Ticket ID: [ID].
Gefunden : {'email': 1, 'id': 1, 'name': 1}

Original : Supportanfrage von Li Wei: li.wei@uni.edu zum Kurscode ID-2024-LLM.
Maskiert : Supportanfrage von [NAME]: [EMAIL] zum Kurscode [ID]-LLM.
Gefunden : {'email': 1, 'id': 1, 'name': 1}
```

Redaktion-Script - Antwortqualität der LLMs (roh vs. maskiert)

Modell	Frage	Similarity roh vs. Maskiert
Qwen2.5-3b	Bestellung Kundin?	0.085
Qwen2.5-3b	Ticketnummer & Einreicher?	0.113
Llama3.2-3b	Bestellung Kundin?	0.718
Llama3.2-3b	Ticketnummer & Einreicher?	0.139

Redaktion-Script - Antwortqualität der LLMs (roh vs. maskiert)

„Welche Bestellung hat die Kundin aufgegeben?“

[Q6]

„Die Kundin hat Laptop ID-99123 bestellt.“ Ohne Maskierung

„Die Kundin hat Laptop [ID] aufgegeben.“ Mit Maskierung

Bronze – Regel-/NER-Erkennung

```python
@dataclass
class Entity:
    label: str
    start: int
    end: int
    value: str

EMAIL = re.compile(r"[a-zA-Z0-9_.+-]+@[a-zA-Z0-9-]+\.[a-zA-Z0-9-.]+")
ID = re.compile(r"(ID[-:]?\s?\d{3,}|TK-\d{3,}|ORD-\d{3,}|\d{4}-LLM)")
NAME = re.compile(r"\b[A-Z][a-z]+\s[A-Z][a-z]+\b")
DATE = re.compile(r"(\d{2}\.\d{2}\.\d{4}|\d{4}-\d{2}-\d{2})")
```

Bronze – Regel-/NER-Erkennung

```
--- Beispiel 3 ---
Original : Kursanmeldung: Li Wei schreibt an li.wei@uni.edu bezueglich Kurscode ID-2024-LLM am 01.04.2024.
Maskiert : Kursanmeldung: [PERSON] schreibt an [EMAIL] bezueglich Kurscode [ID]-LLM am [DATE].
Gefunden : [('PERSON', 'Li Wei'), ('EMAIL', 'li.wei@uni.edu'), ('ID', 'ID-2024'), ('DATE', '01.04.2024')]
```

```python
PATTERN_REGEX = [
    (re.compile(r"\bACME\sGmbH\b", re.IGNORECASE), "ORG"),
    (re.compile(r"\buni\b", re.IGNORECASE), "ORG"),
    (re.compile(r"\bAnna\sSchmidt\b", re.IGNORECASE), "PERSON"),
    (re.compile(r"\bMax\sBauer\b", re.IGNORECASE), "PERSON"),
    (re.compile(r"\bLi\sWei\b", re.IGNORECASE), "PERSON"),
    (re.compile(r"\bMaria\sLopez\b", re.IGNORECASE), "PERSON"),
    (re.compile(r"(id|tk|ord)[-]\d+", re.IGNORECASE), "ID"),
]
```

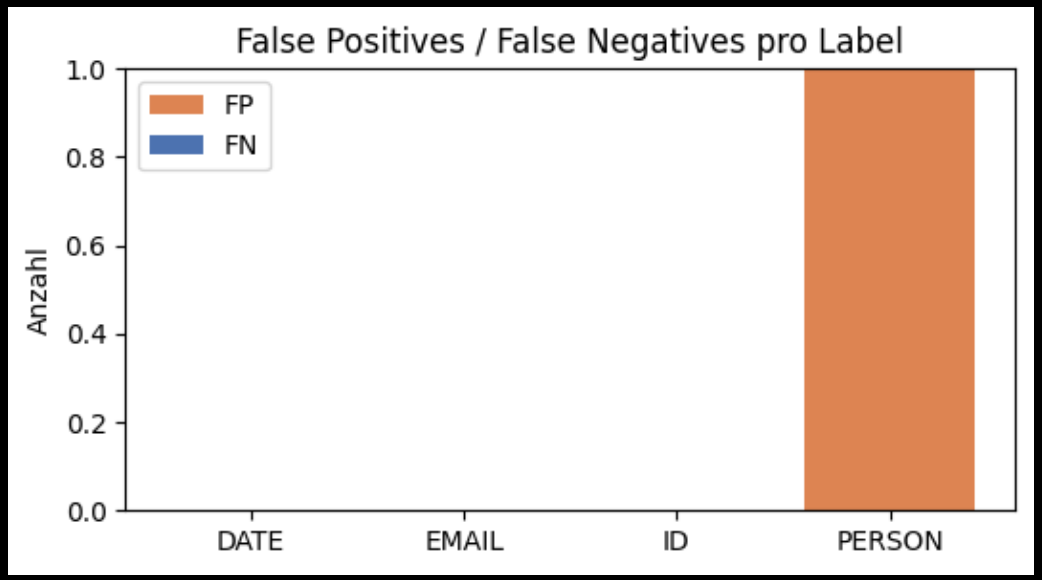
Silber – Falsch-Positiv/-Negativ

„Ticket ORD-7788 gemeldet von Anna Schmidt…“

„Allgemeine Anfrage ohne Kundendaten.“

```
TP: {'PERSON': 33, 'ID': 30, 'EMAIL': 25, 'DATE': 12}
FP: {'PERSON': 1}
FN: {}
Precision: 0.990 | Recall: 1.000 | F1: 0.995
```

Silber – Falsch-Positiv/-Negativ




```
TEXT: Allgemeine Anfrage ohne Kundendaten.
Truth: set()
Pred: {'PERSON'}
FP: ['PERSON']
```

Gold– Selektive Demaskierung & Audit Log

```
Original: Support: Max Bauer, Ticket TK-4455, Kontakt max.bauer@firma.de.
Maskiert: Support: [PERSON:d58a3053a1], Ticket [ID:3abd9d1614], Kontakt [EMAIL:408298183d]
```

```
Maskierter Text: Support: [PERSON:d58a3053a1], Ticket [ID:3abd9d1614], Kontakt [EMAIL:408298183d]
Demaskiert  : Support: Max Bauer, Ticket TK-4455, Kontakt max.bauer@firma.de.
```

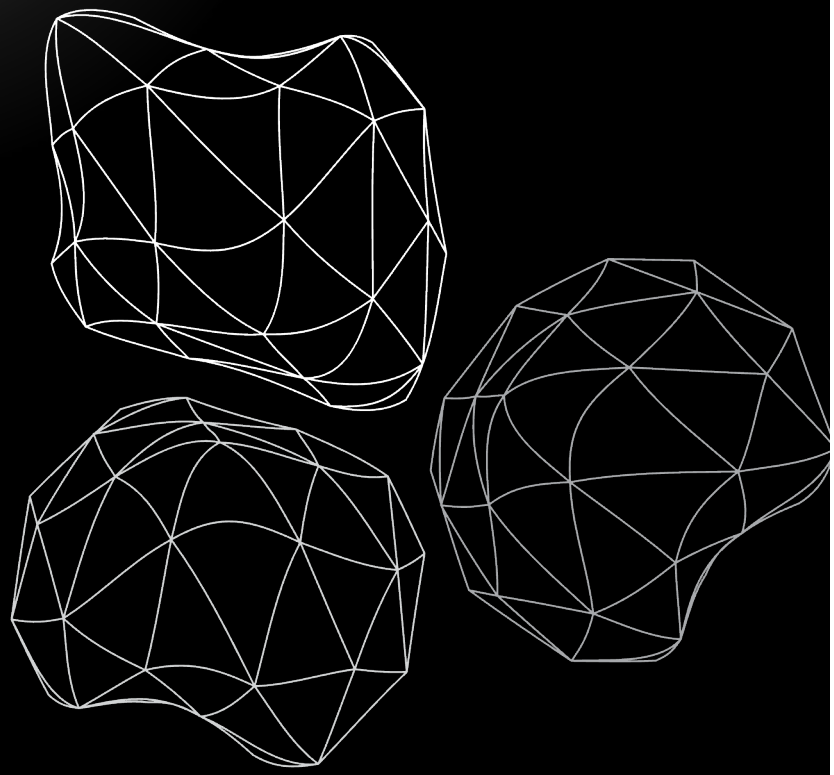
```
Audit-Events:
{'actor': 'analyst', 'reason': 'Support eskalation', 'token_hash': 'd58a3053a1', 'original': 'Max Bauer', 'label': 'PERSON', 'timestamp': '2025-12-11T18:20:58.049982Z'}
{'actor': 'analyst', 'reason': 'Support eskalation', 'token_hash': '3abd9d1614', 'original': 'TK-4455', 'label': 'ID', 'timestamp': '2025-12-11T18:20:58.049999Z'}
{'actor': 'analyst', 'reason': 'Support eskalation', 'token_hash': '408298183d', 'original': 'max.bauer@firma.de.', 'label': 'EMAIL', 'timestamp': '2025-12-11T18:20:58.050005Z'}
{'actor': 'qa', 'reason': 'Qualitaetskontrolle', 'token_hash': 'a21e4283ee', 'original': 'Anna Schmidt', 'label': 'PERSON', 'timestamp': '2025-12-11T18:20:58.080713Z'}
```

Fazit

- **PII ist überall und LLMs vergessen nicht von selbst**
- **Redaktion-Script oder allgemeiner Schutz bevor Daten ins LLM gehen sind ein wirksamer Schutzfilter**
- **Auswirkung auf die Qualität ist meist moderat**

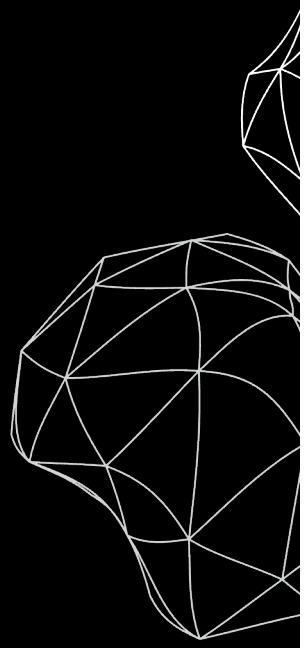
Ausblick

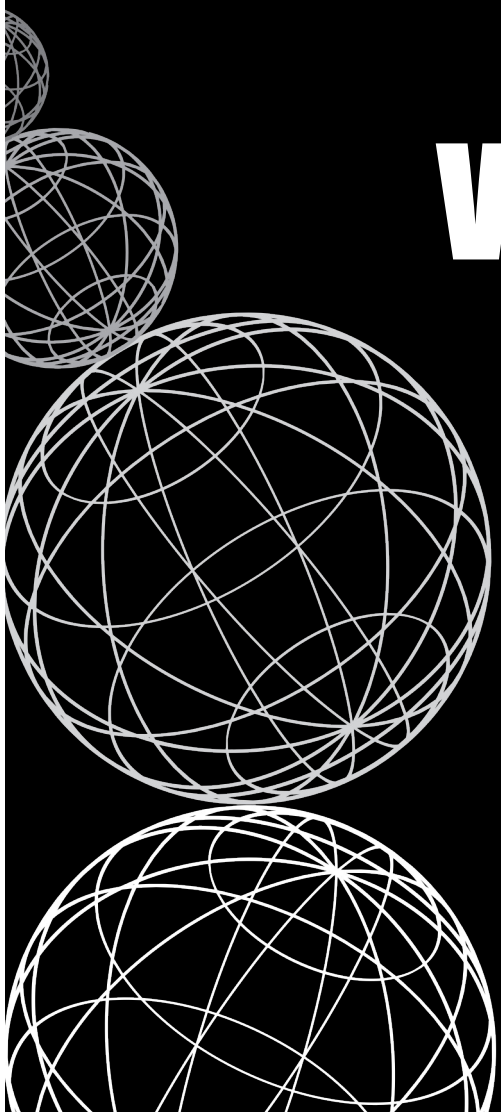
- **Bessere PII-Erkennung**
- **Weitere Qualitätsbewertung**
- **Integration in wirklichen Workflow**



Quellen

- [Q1]: https://ai.gopubby.com/the-training-pipeline-of-large-language-models-afd5fa57df46
- [Q2]: https://arxiv.org/abs/2012.07805
- [Q3]: https://www.unite.ai/de/prompt-hacking-and-misuse-of-llm/
- [Q4]: https://microsoft.github.io/presidio/supported_entities
- [Q5]: https://microsoft.github.io/presidio/
- [Q6]: https://www.youtube.com/watch?v=ITpyjyZDwcs





Vielen Dank für eure Aufmerksamkeit

Fragen ?