Projektdokumentation: Automatische Klassifizierung von Support-Tickets

Ein kleines KI-Modell vor einer großen Herausforderung

Von: Robin Dar
Datum: September 2025

---

1. Die Mission: Kann eine kleine KI den Support-Alltag meistern?

Ein bekanntes Szenario: Der Posteingang für Support-Anfragen quillt über. Von "Passwort vergessen" über "Meine Rechnung stimmt nicht" bis hin zu emotionalen Kunden-Feedbacks ist alles dabei. Die manuelle Sortierung dieser Tickets ist nicht nur repetitiv, sondern auch ein Zeitfresser und eine potenzielle Fehlerquelle.

Die Kernidee dieses Projekts war es, dieser Herausforderung mit den Mitteln kleiner, ressourcenschonender KI-Modelle zu begegnen. Das Ziel war klar definiert: Ein lauffähiges Python-Programm zu entwickeln, das eine Support-Anfrage analysiert und ihr automatisch eine passende Kategorie zuordnet – und das alles auf einem Standard-Rechner, ohne die Notwendigkeit teurer Server-Infrastruktur.

---

2. Die Werkzeuge: Ein Team aus Spezialisten im Test

Im Laufe des Projekts wurden verschiedene kleine KI-Modelle evaluiert, um eine zuverlässige Lösung zu finden. Der finale Prototyp nutzt eine **hybride Architektur**, die eine mehrstufige KI-Pipeline mit regelbasierter Logik kombiniert, um die Präzision zu maximieren. Die zentralen KI-Komponenten sind:

1. **Der Emotion-Scanner (oliverguhr/german-sentiment-bert):** Ein spezialisiertes BERT-Modell, das die emotionale Tonalität eines Textes (positiv, negativ, neutral) bestimmen soll.
2. **Der Themen-Experte (MoritzLaurer/mDeBERTa-v3-base-mnli-xnli):** Ein flexibles NLI-Modell, das für die "Zero-Shot"-Klassifizierung der Absicht und des finalen Themas eingesetzt wird.

Technisch stützt sich das Projekt auf die transformers-Bibliothek von Hugging Face als universelle Schnittstelle zu den Modellen und torch als zugrundeliegendes Berechnungs-Framework.

---

3. Die Reise zur Lösung: Ein iterativer Prozess voller Erkenntnisse

Der Weg zum finalen Code war ein klassischer Entwicklungsprozess aus Hypothesen, Tests und der Analyse von Erfolgen und Fehlschlägen.

- **Phase 1: Einfache Ansätze scheitern an Komplexität**
  Erste Versuche mit einem einzelnen Zero-Shot-Modell und unterschiedlich detaillierten Prompts (von extrem lang bis minimalistisch) zeigten schnell die Grenzen auf. Das Modell war entweder überfordert oder hatte zu wenig Kontext, was zu unzuverlässigen und oft falschen Klassifizierungen führte.
- **Phase 2: Die Pipeline – Eine gute Idee mit schlechten Ergebnissen**
  Um die Schwächen auszugleichen, wurde eine mehrstufige Pipeline entworfen, die Emotion, Absicht und Thema nacheinander analysiert.
    - **Realität:** Dieser Ansatz scheiterte in der Praxis. Die Analyse zeigte, dass die allgemein trainierten Modelle den spezifischen Kontext von Support-Anfragen oft fundamental missverstehen. Das Sentiment-Modell stufte neutrale Anfragen wie "Passwort vergessen" als NEGATIVE ein. Da die erste Stufe der Pipeline bereits fehlerhafte Daten lieferte, war das Endergebnis oft noch unbrauchbarer als zuvor.
- **Phase 3: Der finale hybride Ansatz**
  Als letzte Iteration wurde die KI-Pipeline mit einem regelbasierten Keyword-System kombiniert. Die Hoffnung war, dass die klaren Regeln der KI als "Leitplanken" dienen könnten. Wie die Tests jedoch zeigten, war auch dieser Ansatz nicht in der Lage, die grundlegenden Schwächen der Modelle zuverlässig auszugleichen.

---

4. Die finale Implementierung: Ein fehlgeschlagenes Experiment mit klaren Ergebnissen

Der finale Code repräsentiert den am weitesten entwickelten Prototypen dieses Projekts. Er kombiniert Keyword-Matching mit einer Zwei-Stufen-KI-Analyse, um eine Klassifizierung vorzunehmen.

Trotz seiner konzeptionellen Raffinesse hat die praktische Erprobung gezeigt, dass das Programm **keine zuverlässige Klassifizierung** leistet. Die Ergebnisse sind oft unpräzise oder schlichtweg falsch, was die grundlegenden Limitationen der verwendeten Modelle unterstreicht.

Der vollständige Quellcode, der den finalen hybriden Ansatz demonstriert, ist im folgenden GitHub-Repository einsehbar:

github.com/Rattatatt-git/slm/tree/main/SLM-Projekt

---

5. Mein Fazit: Ein ehrliches Urteil über kleine Modelle in der Praxis

Dieses Projekt war ein wertvoller Praxistest, der vor allem die realen Grenzen kleiner KI-Modelle aufzeigte.

- **Kleine Modelle sind hochspezialisierte Werkzeuge, keine Mini-Alleskönner.** Im Gegensatz zu großen, generativen Modellen wie GPT-4 fehlt ihnen das tiefe, intuitive Sprachverständnis. Man muss lernen, die Aufgabe an die Fähigkeiten des Modells anzupassen, nicht umgekehrt.
- **Der Kontext ist der entscheidende Faktor.** Die größte Hürde war, dass die allgemein trainierten Modelle den spezifischen Kontext von Support-Anfragen oft falsch interpretierten. Diese semantische Lücke war die Hauptursache für die meisten Fehlklassifizierungen.
- **Ein "begrenzter Erfolg" ist ein wertvolles Ergebnis.** Das finale Programm ist kein 100%ig treffsicherer Klassifikator. Und genau das ist die wichtigste Erkenntnis dieses Projekts. Der iterative Prozess hat unmissverständlich bewiesen, dass die hier eingesetzten, allgemein vortrainierten Modelle für diese nuancierte Aufgabe **ungeeignet sind**.

Zusammenfassend lässt sich sagen: Das Projekt hat erfolgreich einen funktionierenden Prototypen hervorgebracht und dabei die klaren Grenzen kleiner, allgemeiner KI-Modelle für spezialisierte Aufgaben demonstriert. Das finale Programm funktioniert technisch, aber die Qualität seiner Ergebnisse ist unzureichend für einen Praxiseinsatz.

Für eine wirklich produktiv einsetzbare Lösung gibt es nur einen logischen nächsten Schritt: das **Fine-Tuning**. Man müsste ein Basis-Modell (wie das empfohlene DistilBERT) nehmen und es auf einem Datensatz von echten, anonymisierten Support-Tickets nachtrainieren. Nur so kann die KI den spezifischen Kontext und die Sprache der Kunden lernen, die für eine hochpräzise Klassifizierung unerlässlich sind.

Das Projekt endet somit nicht mit einer perfekten Lösung, sondern mit einer professionellen und fundierten Handlungsempfehlung, die auf praktischen Experimenten und einer ehrlichen Analyse der Ergebnisse basiert.