

Projektpräsentation – automatisierte Analyse von Mundart-Chatnachrichten

12.12.2025 | Raphael Weiss

1. Ausgangslage – Projektstand bei Zwischenpräsentation

Idee	Herausforderungen
Automatisierte Sentiment-Analyse für Schweizerdeutsche Chatnachrichten.	<ul style="list-style-type: none">Chatnachrichten sind wenig und extrem kurz (oft 2–5 Wörter)Kein Kontext → Sentiment sehr schwer zu bestimmen (vor allem für Embedding-Modelle)Hohe Accuracy, aber künstlich verzerrt → Modelle erkennen triviale Muster (z.B. einzelne Token, Emojis), dadurch schlechte Generalisierbarkeit (Overfitting)Fehlende Vielfalt im Training (wenig Variationen)
Verwendete Modelle <ul style="list-style-type: none">BoW + Logistic Regression → mit PreprocessingTF-IDF + Logistic Regression → Uni- und Bigramme, PreprocessingSBERT + Logistic Regression → paraphrase-multilingual-MiniLM-L12-v2	

2. Erweiterung & Optimierung des Projekts

2.1 Datenerweiterung und Analyse

- **Generierung zusätzlicher Mundartnachrichten mit mehr Kontext**
→ komplexere Sätze → bessere Modellrobustheit
- **Kein Einsatz von Emojis**
Begründung: Emojis sind extrem starke Sentimentmarker → Modelle lernen „Emoji = Sentiment“ statt semantische Inhalte → schlechte Generalisierung.
- **Mehr Daten- und Modell-Analysen:** z.B. Confusion Matrix und Zipf-Analyse

2.2 Einführung einer mehrstufigen Klassifikation

Sentiment mit Unterklassen (Intents)	Vorteile
<ul style="list-style-type: none">• Positiv → Dankbarkeit, Freude & gute Laune ...• Negativ → Stress & Überforderung, Traurigkeit ...• Neutral → Smalltalk, Organisation & Abmachungen ...	<ul style="list-style-type: none">• Erhöhte semantische Differenzierung• bessere Auswertbarkeit und kontextsensitiver Chat-Antwortoptionen

2.3 Neue Modelle / Methoden

a) Next-Word-Prediction mittels N-Gramm Language Model

- Unterstützung beim Generieren von Text
- Nutzung eines 1-, 2-, 3-Gramm-Modells mit Backoff-Strategie
- "Wie könnte der Satz am wahrscheinlichsten weitergehen?"

b) Predict Answer (SBERT-Embedding Retrieval)

- semantisch ähnliche Nutzeranfragen finden
- passende Antwort aus vordefiniertem Antwortkorpus auswählen

inkl. Debug-Ansicht

- erklärt, welche Nachbarn ähnlich sind und welche Ähnlichkeitswerte vergeben wurden → wichtig für Interpretierbarkeit

Warum Retrieval und kein generatives Modell?

- Zu kleine Datenmenge: für generative Modelle viel mehr Daten nötig
- Funktionsfähigkeit: schnell und stabil bei klar definierten Intents und Standardantworten
- Erklärbarkeit: Mit SBERT-Nachbarn sieht man transparent, warum eine Antwort gewählt wurde.

2.4 Integration in Streamlit-App (Frontend)

<https://mundartchat.streamlit.app>

Modelle & Datengrundlage

- Anzahl Chatnachrichten: 900
- > 😊 Sentiment (3) & Intents (18)
- > 💬 Standardantworten (Defaults)
- > 🖥️ Verwendete Modelle
- > 🔎 Modell-Performance (Testset)
- > 📈 Label-Verteilung
- > 📏 Textlängen (Tokens)
- > 📈 Token-Statistik
- > 📈 Zipf-Analyse (Token-Verteilung)
- > 🌱 N-Gramm-Statistik (LM)
- > 📄 Projektpräsentation (PDF)

Mundart-Chat Demo

Sentiment, Next-Word, Antwort-Retrieval für Schweizerdeutsch-Chat

Mundart-Nachricht eingeben

z.B. «ich ha kei bock meh uf dä stress»

Sentiment-Klassifikation **Next-Word Vorschlag** Antwortvorschlag Debug Nachbarn

Next-Word Vorschläge berechnen

3. Pipeline – Gesamtprozess

1 Datengrundlage <ul style="list-style-type: none">• 900 Sätze (manuell + generiert)• Chatpairs (Usertext–Antwort-Paare) als Grundlage für das SBERT-basierte Antwort-Retrieval	2 Preprocessing <ul style="list-style-type: none">• Dialektstandardisierung• Lowercasing• Tokenisierung auf Wortebene• Entfernung von Stopzeichen / Noise-Token
3 Feature Engineering & Modelltraining BoW (Bag-of-Words): CountVectorizer und Klassifikation via multinomiale Logistic Regression TF-IDF: TfidfVectorizer und Klassifikation via multinomiale Logistic Regression SBERT: SentenceTransformer und multinomiale Logistic Regression auf Embedding-Raum	
3 Neu: N-Gramm Language Model <ul style="list-style-type: none">• Training von 1-, 2-, 3-Grammen• Backoff-Strategie für Next-Word-Vorhersagen• Nutzung in der App zur Unterstützung der Texteingabe	3 Neu: Antwort-Retrieval (SBERT-basiert) <ul style="list-style-type: none">• Suche nach semantisch ähnlicher Chatnachricht und Auswahl der passenden Antwort• Debug-Ansicht für „Nearest Neighbors“

4

Evaluationen & Visualisierungen in der App

- Accuracy, Precision, Recall, F1
- Konfusionsmatrix (Heatmap) für SBERT
- Label-Verteilung
- Token-Statistik, N-Gramm-Statistik
- Zipf-Analyse
- Kosinus-Ähnlichkeit

4. Fazit – Optimierte Lösung und Verbesserungspotential

Erreichte Optimierungen

- etwas bessere Generalisierbarkeit und robustere Modelle für kurze Dialekttexte
- mehr semantische Tiefe dank Intent-Klassen
- Nachvollziehbarkeit durch Debugging
- Demo-App für Sentimentanalyse, Next-Word, Antwortvorschläge

Verbesserungspotenzial

- grössere & vielfältigere Datengrundlage (mehr Dialektvarianten, echtes Chatmaterial)
- mehr Kontextverarbeitung (z.B. Dialogmodelle)
- Erweiterung der Intents und Multi-Label-Klassifikation
- Kombination aus Retrieval und leichter Generierung (RAG)