Coding Challenge Algonaut: Christian Liedl

Part 1: LLM Coding Projekt

Ziel: Klassifizierung von Tweets über Katastrophen: Fake oder real?

Datengrundlage: Tweets über Katastrophenfälle (labeled)

Part 1: LLM Coding Projekt

Struktur der Lösung

1. EDA: Daten sind balanced, wenige fehlende Werte, Untersuchung der häufigsten Wörter und Wortpaare in beiden Kategorien, …
2. Data cleaning and preprocessing: urls löschen, Abkürzungen ersetzen, Emojis durch Text ersetzen.
3. Feature Engineering: “Meta data”: word_count, url_count, punctuation_count, hasthag_count, mention_count
4. Embedding mit BERT
5. Classifer: Dense Neuron Layer mit Dropout
6. Cross-validation während Training

Ich habe erst kürzlich mit diesem Kaggle-Projekt begonnen. Daher ist NLP relativ neu für mich und ich traue mir nicht zu, große Ratschläge zu geben: Allerdings habe ich wesentlich mehr Erfahrung mit Data Engineering, Machine Learning und Deep Learning in anderen Kontexten (Computer Vision, Forecasting, etc.).
Generell schließe ich aus meinen bisherigen Erfahrungen, dass die Datenqualität oft wesentlich wichtiger ist als das Modell. Oft weisen mehrere fine-getunete Modelle ähnliche Performance auf, die Art der Datenaufbereitung führt allerdings zu wesentlichen Unterschieden.
Mein zweiter Tipp wäre, mit Kollegen über technische Optionen und Probleme zu reden, bevor man sich auf einen Lösungsweg festlegt, wenn man sich nicht sicher ist. Man spart dadurch meiner Erfahrung nach sehr viel Zeit, lernt schneller und verbessert das Arbeitsklima.

Coding Challenge Algonaut

Part 2: Fragen zu Llama-2

Ziel: NLP Lösung zur Beantwortung von Fragen zu aktueller Forschung zu Llama-2

Datengrundlage: Titel und Abstracts von arXiv Papen zum Thema Llama-2

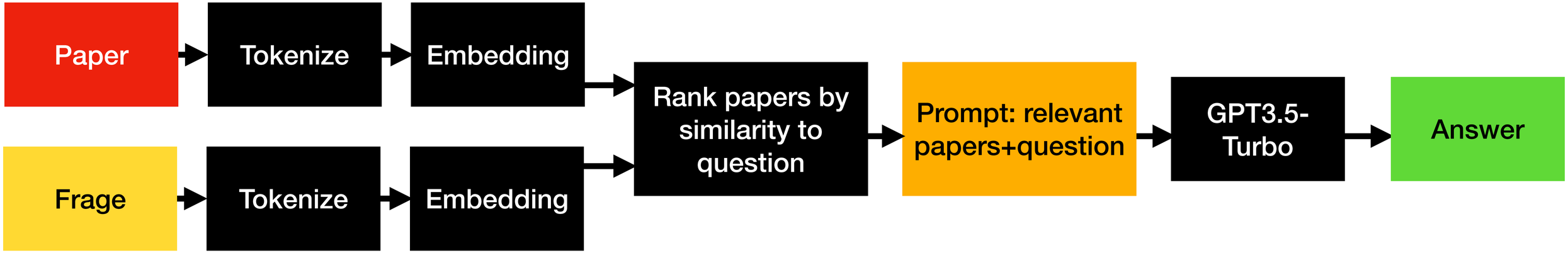
Part 2: Llama-2 chatbot

Struktur der Lösung

Ich habe mich für ein RAG (Retrieval-augmented generation) model entschieden

Die Idee ist, einen bestehenden Chatbot mit zusätzlicher Information (Context) zu füttern, die für die eigentliche Frage relevant ist.

1. Frage und Paper werden tokenized und embedded (Hier habe ich ein Transformer Modell gewählt, das genug Tokens zulässt, um den gesamten Titel und Abstract zu verarbeiten: “allenai/longformer-base-4096”
2. Die Paper werden aufgrund ihrer Ähnlichkeit (Cosine-Similarity) zur Frage gerankt
3. Die 10 ähnlichsten Paper werden als Context dem Prompt hinzugefügt
4. Der Prompt wird an GPT-3.5-Turbo übergeben, welches eine Antwort generiert



Part 2: Llama-2 chatbot

Dokumentation

1. Create virtual environment
2. Install dependencies from requirements.txt
3. Activate virtual environment
4. API key: You can insert API key as “api_key” in main.py (line 8). If not, the program will prompt you for the API key when you run it
5. Run main.py
6. Insert your question

create_dataset.ipynb:
This Jupyter notebook creates the dataset of arXiv papers used in the model:
Importantly, it also creates the Embeddings for the Papers, so whenever the model is changed, one has to run this script again.

Part 2: Llama-2 chatbot

Ein paar Kommentare

1. Embedding: Zunächst habe ich BERT ausprobiert. Ich musste die Größe der Abstracts reduzieren, da BERT nur bis zu 512 Token zulässt. Dazu habe ich die keyword extraction von spacy verwendet. Ich fand jedoch, dass die Leistung besser ist, wenn der Volltext verwendet wird, weshalb ich "allenai/longformer-base-4096" (bis zu 4096 Zeichen) gewählt habe. Natürlich kann man auch einfachere Einbettungen als Transformatoren verwenden, aber ich hatte keine Zeit, deren Performance zu testen.

2. Die Daten schienen bereits sehr sauber zu sein, daher habe ich nur Zeilenumbrüche entfernt. Evtl. kann man noch urls/links/stopwords entfernen.

3. Die Ergebnisse sind sehr abhängig von der Prompt-Struktur. Ich habe mit verschiedenen Möglichkeiten gespielt, den Kontext (Paper) in den Prompt zu integrieren, und die Ergebnisse sind sehr unterschiedlich! Wahrscheinlich kann hier noch viel verbessert werden.

4. Die Antworten sind nicht besonders robust: Bei der gleichen Frage, wird manchmal behauptet, der Kontext (Paper) enthalte keine Information zur Frage, manchmal wird der Kontext richtig erfasst.