Vergleich von verschiedenen Sprachmodellen zur Extraktion von Keywords in Eventbeschreibungen

Working paper

Louis Huber*, Student MBI, Universität St.Gallen

St.Gallen, April 2024

Abstract

In dieser Arbeit werden vier beliebte *Specialized Language Models* daran getestet, wie gut sie im Vergleich zu GPT 4.0 Keywords aus Beschreibungstext von Events extrahieren können. Es konnten grosse Unterschiede zwischen den Modellen unter sich festgestellt werden und GPT 4.0 liefert immer noch die besseren Ergebnisse.

1 Einleitung

Für das Modul *IC: From Data2Dollar - Dein Technologiekoffer von der Datenbeschaffung bis zur Visualisierung*[1] führen wir eine Analyse von Eventangeboten verschiedener Internetplattformen durch. Unter anderem führen wir auch eine Analyse der Beschreibungen[2] der jeweiligen Angebote der Webseite durch.

Da wir die wichtigsten Wörter/Termini aus den einzelnen Beschreibungen auslesen wollen, müssen diese zuerst aus den teil längeren Texten extrahiert werden. Aus diesem Grund suchen wir ein geeignetes Language Model, in diesem Kontext kommen grundsätzlich *Large Language Models(LLMs)* und *Specialized Language Models* in betracht. Letztere wurden auf einen spezifischen Use Case trainiert.

In diesem Paper möchte ich als Teilschritt der Arbeit eine Evaluation verschiedener Language Modelle durchführen. Primär ist das Ziel, die

*louisphilippe.huber@student.unisg.ch, 19-742-949

[1] Kurswebseite: https://courses.unisg.ch/event/events/by-term/cdb7331b-2557-46b9-b5dd-4151d8bf0962/14492427

[2] Ein Beispiel wäre: https://www.getyourguide.com/new-york-city-l59/liberty-harbor-helicopter-adventure-t15252/, nach dem Zwischentitel "Full description"

Beantwortung der Frage, welches Language Modell am vollständigsten und präzisesten die semantisch relevanten Keywords aus den en auslesen kann.
Gleichzeitig ist dies ebenfalls eine allgemein intressante Arbeit für die Forschung an Language Models, da sie die Unterschiede von diesen Anhang eines sehr spezifischen Beispiels betrachtet.

2 Literatur

Eine Übersicht über die Entwicklung der LLMs finden wir bei (Zhao et al., 2023). LLMs werden definiert als Tranformer Language Models welche mit einer gewaltigen Menge an Daten trainiert wurden. Beispiele hierfür wären Meta's Llama 3 oder das bekannte GPT 4.0 von OpenAI. Sie sind in der Lage, komplexe Aufgabenstellungen zu lösen, in unseren Fall müssen sie Keywords aus Texten auslesen.
Das Paper von (Grangier et al., 2024) setzt sich mit den unterschieden von kleinen und grossen Languange Models auseinander und fokussiert sich dabei insbesondere auf die Trainingskosten und dessen Intensität. Bei einem kleinem Budget wird von den Autoren empfohlen, sich auf generisches Pretraining zu fokussieren.
(Ding et al., 2024) haben wiederum aufgezeigt durch ihr erstelltes Framework *ULTRAFUSER* mithilfe eines qualitativ hochwertigem und spezialisiertem Datasest ein Modell zu kreieren, welches in den Bereichen Natural Language, Programmiercode und mathematischen Symbolen besonders gute Ergebnisse aufzeigten. Daraus kann man schliessen, dass die Spezialisierung eines Modells dieses deutlich verbessern kann und entsprechend den generischen Modellen auch gegebenenfalls vorzuziehen ist.

3 Methodik

Für die Suche nach geeigneten Modellen wurde die Plattform hugginface[3] verwendet und nach den Begriffen "keyword extractor" und "keyphrase extractor" gesucht und die am meisten verwendeten ausgewählt, welche auch für englische Texte geeignet sind. Hierbei handelt es sich somit um *Specialized Language Models*, die für diesen speziellen Gebrauch erstellt wurden, aber deutlich kleiner sind als LLMs. Als Beispiel dafür wurde das beliebte OpenAI's ChatGPT 4.0 verwendet.
Am Ende wurden vier *Specialized Language Models* ausgewählt: **yanekyuk/ bert-keyword-extractor(BKW)**[4], welches auf googles BERT[5] basiert, welches an-

[3] huggingface.co
[4] https://huggingface.co/yanekyuk/bert-keyword-extractor
[5] https://huggingface.co/google-bert/bert-base-cased

schliessend noch entsprechend "fine tuned" wurde.

Als nächstes das Modell **ml6team/ keyphrase-extraction-kbir-inspec(KEK)**[6] und basiert auf dem Keyphrase Boundary Infilling with Replacement (KBIR) Model[7], welches dann auf das Trainieren von Abstracts trainiert wurde. Die Ersteller empfehlen es auch ausschliesslich dafür zu verwenden, jedoch ist es gerade interessant, es für andere Texte zu verwenden.

Das dritte Modell ist **sentence-transformers/ all-MiniLM-L6-v2(AML)**[8] und basiert auf dem MiniLM Model[9] und wurde für spezifisch für das Clustering und semantischer Suche trainiert.

Das vierte und letzte Modell nennt sich **ml6team/keyphrase-generation-t5-small-inspec(KGSI)**[10] und wurde ebenfalls mit Abstracts, analog zu KBIR, trainiert, basiert aber auf google's T5 small Model[11].

Mithilfe des Github Copilot wurde ein Python Skript(siehe Anhag) erstellt, welches neun zufällig ausgewählte Eventbeschreibungen mit unterschiedlicher Länge mit dem jeweiligen Modells ausführte.

Da ChatGPT 4.0 die besten Ergebnisse produziert hat[12], werden dessen Ergebnisse als Benchmark verwendet. Ein eigens erstelltes GPT[13]vergleicht nun die Liste der Ergebnisse der einzelnen Models mit den von ChatGPT anhand der semantischen Ähnlichkeit auf einer Skala von 0 bis 1.

4 Ergebnisse

Die Ergebnisse der einzelnen Modelle fielen sehr unterschiedlich aus. Ein Beispiel wird im Anhang (Keyword Extraktion: Beispiel) aufgezeigt. Die gesamten Ergebnisse sind als Tabelle im Anhang (Ergebnisse) zu finden, sowie einen Plot der Ergebnisse.

Insgesamt hat sich gezeigt, dass AML und KGSI die besseren Ergebnisse geliefert haben. Sie haben jeweils deutlich mehr Keywords extrahieren können. Ein besonderer Beschreibungstext war die Nummer fünf, wo nur ein sehr kurzer Text untersucht wurde. Dort wurden von KEK gar keine Keywords extrahiert. Mit einem Durchschnitt von 0.89 kommt KGSI dem Ergebnis von GPT 4.0 gemäss dieser Überprüfung schon recht nahe. AML ist mit diesem Ergebnis fast

[6] https://huggingface.co/ml6team/keyphrase-extraction-kbir-inspec
[7] https://huggingface.co/bloomberg/KBIR
[8] https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2
[9] https://huggingface.co/nreimers/MiniLM-L6-H384-uncased
[10] https://huggingface.co/ml6team/keyphrase-generation-t5-small-inspec
[11] https://huggingface.co/google-t5/t5-small
[12] Dies wird im Diskussionteil noch erläutert
[13] https://chat.openai.com/g/g-Lu88eIIxw-keyword-rater

gleichauf. Auffallend ist dabei, dass sowohl bei AML und bei KGSI deutlich mehr Keywords extrahiert wurden, was je nach Use Case nicht wünschenswert ist. Die Modelle generieren auch immer völlig unsinnige Keywords: Beispielsweise ein "St" ohne das folgende, entscheidende Wort auch mitzuextrahieren. Teilweise werden auch einfach Wörter getrennt: Statt "Roasted" (von Roasted Duck) extrahierte KEK "Ro" und "asted" getrennt.

5 Diskussion

In dieser Arbeit wurden zuerst vier verschiedene Language Modelle ausgewählt und ihnen zufällige Eventbeschreibungen gegeben, welche sie analysieren sollten und die Keywords extrahieren sollten. Das Ergebnis hat gezeigt, dass die Specialized Language Models noch nicht auf den Stand der LLMs, in diesem Fall GPT 4.0, sind: Sie lassen Vieles weg oder extrahieren deutlich mehr Daten als nötig. Sie sind auch nicht zuverlässig, was die Qualität des Outputs betrifft: Teilweise extrahieren sie nur einzelne Buchstaben oder trennen Wörter.

Auffallend ist, das obschon KEK und KGSI auf die Extraktion von Abstracts trainiert wurden, die Ergebnisse sich so deutlich unterscheiden. Naheliegend wäre es, dass das zugrundeliegende Modell in diesem Fall für diesen Case passender war.

Für unsere weitere Arbeit an unserem Projekt würde ich deshalb KGSI oder AML nutzen, da dort die Ergebnisse am besten waren. Gleichzeitig sind nun auch die Schwächen und Limitationen bekannt.

Meine These ist, dass GPT 4.0 durch das grössere Language Modell sowie die erhöhte Rechenkapazität qualitativ nicht durch solche deutlich kleineren und lokal ausgeführten Modelle zu schlagen ist. Eine spannende Idee wäre die Umsetzung von *Specialized Large Language Models*: Diese würden auf Basis von grosser Datenmenge für spezifische Anforderungen trainiert werden und hätten das Potential, die nicht-spezialisierten Modelle zu schlagen. Dies wäre andererseits entsprechend aufwändiger und kostenintensiver.

Weitere Forschung könnte auch in Zukunft untersuchen, wie sich verschiedene Modelle in bestimmten Use Cases verhalten und ob der Gap zu den LLMs grösser oder kleiner wird. Als Limitation in der Arbeit ist aufzuführen, dass Anzahl von Beschreibungstexten und Modellen noch erhöht werden könnte, was die Präzision erhöhen könnte. Bezüglich der Methodik ist weiterhin zu bedenken, dass Chat GPT eher ein pragmatischer Ansatz ist, um ein Testscoring durchzuführen. Auch dies könnte

in späterer Forschung noch überdacht werden.

6 Anmerkungen

Zur Korrektur des Textes und zur Erstellung der Plots wurde (OpenAI, 2024) verwendet.

Appendix

Python Skript

```
from transformers import AutoTokenizer,
AutoModelForTokenClassification,
T5ForConditionalGeneration, T5Tokenizer
import torch

# Load the model and tokenizer
tokenizer = AutoTokenizer.from_pretrained
('ml6team/keyphrase-generation-t5-small-inspec')
model = AutoModelForTokenClassification.
from_pretrained('ml6team/keyphrase-generation-t5-small-inspec')

def extract_keywords(texts):
    for text in texts:
        keywords = []
        # Tokenize the text
        inputs = tokenizer(text, return_tensors="pt",
        truncation=True, max_length=512)
        # Get the model's output
        outputs = model(**inputs)
        # Get the predictions
        predictions = torch.argmax(outputs.logits, dim=-1)
        # Extract the keywords
        for i, token in enumerate(inputs["input_ids"][0]):
            if predictions[0, i].item() == 1:
                keywords.append(tokenizer.decode([token]))
        print(', '.join(keywords))

# Test the function
texts=["Hier sind die Beschreibungen der Events."]
extract_keywords(texts)
```

Keyword Extraktion: Beispiel

Die originale Beschreibung finden Sie hier: https://www.getyourguide.com/new-york-city-l59/washington-dc-day-trip-by-bus-from-new-york-t69848/. Die folgenden Ergebnisse haben die Modelle produziert:

- **GPT 4.0**: Washington DC, day trip, New York City, American Revolution, Civil War, Lincoln Memorial, Vietnam Veterans Memorial, Arlington Cemetery, Capitol Hill, White House, photo opportunity, Jefferson, Iwo Jima Memorial, Union Station
- **yanekyuk/bert-keyword-extractor(BKW)**(Score:0.65): Washington, New, American, Lincoln, Vietnam, Arlington, New, Washington, Capital, White, I, Vietnam, Arlington, time, Union
- **ml6team/keyphrase-extraction-kbir-inspec(KEK)**(Score:0.80): DC, Revolution, War, Memorial, Veterans, Memorial, Cemetery, DC, Hill, House, of, Independence, Cemetery, War, ed, ys, Station
- **sentence-transformers/all-MiniLM-L6-v2(ALM)**(Score:0.90): discover, monuments, of, dc, on, day, trip, from, new, learn, about, dive, into, key, like, american, revolution, war, see, sights, like, lincoln, memorial, vietnam, memorial, arlington, cemetery, depart, new, travel, air, conditioned, bus, to, dc, relax, on, drive, head, to, hill, once, get, dc, where, meets, stop, hear, more, from, guide, about, workings, of, admire, memorials, to, jefferson, who, of, independence, lincoln, who, visit, i, ##wo, jim, ##a, and, vietnam, memorials, understand, these, events, country, visit, arlington, cemetery, over, 200, 000, soldiers, civil, war, kennedy, were, buried, finally, some, time, at, station
- **ml6team/keyphrase-generation-t5-small-inspec(KGSI)**(Score:0.88): from, New, York, City, Learn, about, the, government, and, dive, into, key, historic, moments, like, the, American, Revolution, and, War, See, like, Memorial, ,, part, New, York, City, and, travel, spacious, air, conditioned, to, Relax, on, the, drive, and, head, to, once, you, get, to, to, find, out, where, congress, meets, Stop, outside, White, House, for, photo, opportunity, and, hear, more, from, guide, about, inner, working, of, the, building, e, to, ,, who, wrote, the, of, and, who, ended, Visit, w, o, Jim, to, understand, events, country, Visit, Cemetery, where, over, 200,000, since, the, War, s, were, buried, Finally, ,, time, at, Station, ¡/s¿

Table 1: Tabelle der Ergebnisse.

Model:	yanekyuk/ bert-keyword-extractor(BKW)	ml6team/ keyphrase-extraction-kbir-inspec(KEK)	sentence-transformers /all-MiniLM-L6-v2(AML)	ml6team/ keyphrase-generation-t5-small-inspec (KGSI)
Score 1:	0.65	0.85	0.8	0.72
Score 2:	0.7	0.75	0.85	0.88
Score 3:	0.6	0.85	0.88	0.9
Score 4:	0.65	0.8	0.9	0.88
Score 5:	0.65	0.8	0.9	0.88
Score 6:	0.25	0.35	0.9	0.95
Score 7:	0.45	0.8	0.86	0.95
Score 8:	0.35	0.6	0.9	0.95
Score 9:	0.45	0.8	0.95	0.9
median:	0.6	0.8	0.9	0.9
avg:	0.528	0.733	0.88	0.89

Ergebnisse

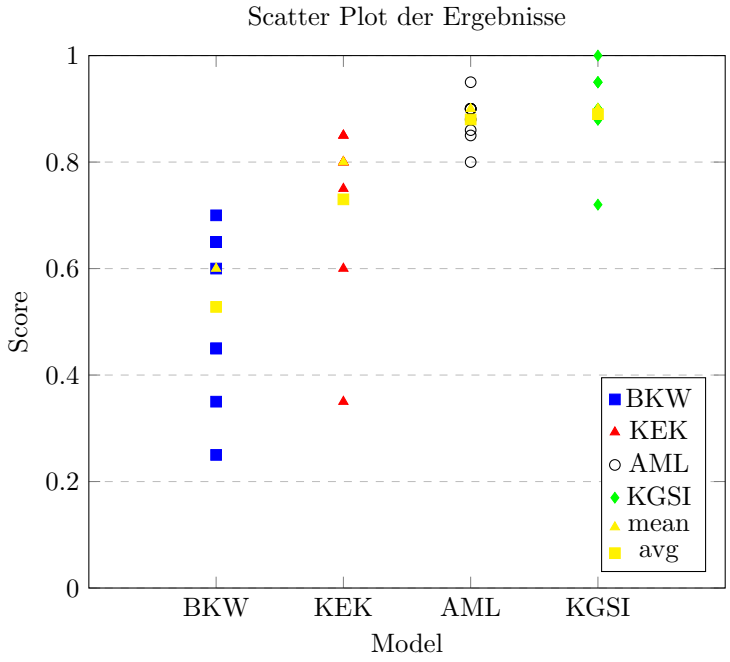

Figure 1: Scatter Plot der Ergebnisse

Plot der Ergebnisse

References

Ding, N., Chen, Y., Cui, G., Lv, X., Zhao, W., Xie, R., Zhou, B., Liu, Z., & Sun, M. (2024). Mastering text, code and math simultaneously via fusing highly specialized language models.

Grangier, D., Katharopoulos, A., Ablin, P., & Hannun, A. (2024). Specialized language models with cheap inference from limited domain data.

OpenAI. (2024). ChatGPT (Version 28. April 2024) [Large language model]. https://chat.openai.com/share/3fd1984d-0b45-4b2b-9622-08057afed1ee

Zhao, W. X., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., Min, Y., Zhang, B., Zhang, J., Dong, Z., Du, Y., Yang, C., Chen, Y., Chen, Z., Jiang, J., Ren, R., Li, Y., Tang, X., Liu, Z., ... Wen, J.-R. (2023). A survey of large language models.