SWP SS 24

Projektvorschläge Steen

Exploring Content Selection in Zero Shot Text Summarization

- (L)LMs erstellen in vielen Fällen sehr gute Textzusammenfassung
- Wir haben aber relativ wenig Einsicht darin, wie sie die Inhalte selektieren, die im Output auftauchen
- Können wir das ändern?

Projektidee

1. Unterteile Eingabedokument in atomare Aussagen mittels Prompting
2. Matche atomare Aussagen aus der Eingabe auf die generierte Summary
3. Analysiere Faktoren, die die Inklusion/Exklusion einer Aussage aus der Eingabe in der Summary vorhersagen:
    1. Position im Dokument
    2. Vorkommen bestimmter Worte
    3. Vorkommen bestimmter Individuen
    4. Kontextfaktoren (z.B. Wiederholung von Entitäten)

Literatur

- Yixin Liu, Alex Fabbri, Pengfei Liu, Yilun Zhao, Linyong Nan, Ruilin Han, Simeng Han, Shafiq Joty, Chien-Sheng Wu, Caiming Xiong, and Dragomir Radev. 2023. Revisiting the Gold Standard: Grounding Summarization Evaluation with Robust Human Evaluation. In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 4140–4170, Toronto, Canada. Association for Computational Linguistics.

- Huang, Kung-Hsiang, et al. "Embrace divergence for richer insights: A multi-document summarization benchmark and a case study on summarizing diverse information from news articles." arXiv preprint arXiv:2309.09369 (2023).

Projektart

- Dieses Projekt ist gut geeignet für Studierende, die…
  - gerne mit Sprachmodellen arbeiten möchten
  - besser verstehen möchten, wie LLM output zustande kommt.
  - gerne Experimente mit offenem Ausgang durchführen

Heideltime Reimplementierung

- Heideltime (Strötgen und Gertz, 2012) ist ein Time Expression Tagger
- Findet beispielsweise für TLS Verwendung
- Verwendung mit Python ist aber umständlich und integriert nicht gut mit Python Workflow
- Projektziel: Reimplementierung von Heideltime in Python/Python-Extension

Heideltime

- Heideltime ist ein regelbasiertes System
- Muster identifizieren Ausdrücke, zweites Muster regularisiert
- Bestimmte Muster haben zusätzlich Part-of-Speech Restriktionen
- Beispiel für February 25, 2009
- `EXTRACTION=„(%reMonthLong|%reMonthShort)(%reDayNumberTh|%reDayNumber)[\s]?,? %reYear4Digit(, %reWeekday)?“`
- `NORM_VALUE="group(7)-%normMonth(group(1))-%normDay(group(4))"`

Projektziele

- Reimplementierung von Heideltime mit Originalresourcen (s. https://github.com/HeidelTime/heideltime) in Python Modul
- Implementierung entweder in reinem Python, oder als Python Extension

Projektart

- Dieses Projekt ist gut geeignet für Studierende, die…
  - nicht mit LLMs arbeiten wollen
  - gerne über das Design von Software nachdenken
  - ein praktisch verwendbares Werkzeug schreiben wollen

Literatur

- Jannik Strötgen and Michael Gertz. 2010. HeidelTime: High Quality Rule-Based Extraction and Normalization of Temporal Expressions. In Proceedings of the 5th International Workshop on Semantic Evaluation, pages 321–324, Uppsala, Sweden. Association for Computational Linguistics.