Goethe Universität
Institut für Informatik
Texttechnologie

SoSe 2025

NLP-gestützte Data Science

Übung 2

Manuel Schaaf
Prof. Dr. Alexander Mehler

24.06.2025

Übung 2: Transformers

Die Transformer-Architektur ist vielseitig und in NLP aktuell allgegenwärtig. Entsprechend werden Sie unzählige Blogs, Videos und detaillierte Erklärungen zu jedem erdenklichen Anwendungsszenario nach kurzer Suche im Internet finden. Die folgenden Ressourcen sollen daher keine abschließende Sammlung darstellen, sondern dienen nur zur Orientierung.

Allgemein

- Richten Sie Ihre Python 3.13 Entwicklungsumgebung ein.
- Installieren Sie 🤗 `transformers` und `openapi`.
- Machen Sie sich mit 🤗 `transformers` und dem `openai` Client vertraut.

Forschung

- „Attention is All You Need“, Vaswani et al. (2017)
- „BERT“, Devlin et al. (2018)
- „GPT-2“, Radford et al. (2018)
- „RoBERTa“, Liu et al. (2019)

Blogs

- Jay Alammar's Blog: `https://jalammar.github.io/`, insb. Seq2Seq with Attention & The Illustrated Transformer
- Christopher Olah' Blog: `https://colah.github.io/about.html`, insb. Attention (on Distill)
- Stephen Wolfram's Blog: What Is ChatGPT Doing … and Why Does It Work?

Notebooks & Tutorials:

- 🤗 `transformers` Notebooks: `https://huggingface.co/docs/transformers/notebooks`
- „Natural Language Processing with Transformers“, Tunstall (2021): Buch: Buch, Notebooks

Evaluating Zero Shot Performance with oLMpics **50 + 10 P**

Neben dem in der Vorlesung vorgestellten Benchmark für LMs, *GLUE* (Wang et al., 2019), welcher sich auf die Performanz von Modellen nach dem *fine tuning* auf bestimmten *downstream tasks* fokussiert, existieren auch einige Evaluationsdatensätze, die einen eher intrinsischen Ansatz verfolgen. Dazu gehört der *oLMpics* Benchmark (Talmor et al., 2020), der acht Experimente umfasst, welche jeweils das „logische Denkenvermögen" von Sprachmodellen testen sollen.

- Machen Sie sich mit der 🤗 `transformers` Bibliothek[1] vertraut.
  - Hier insbesondere mit *Masked Language Models* und wie Sie die wahrscheinlichste Vorhersage aus einer Auswahl von Tokens für ein maskiertes Token in einem Eingabesatz mit genau einem `[MASK]` Token erhalten können.
- Lesen Sie das Paper: „oLMpics–On What Language Model Pre-training Captures", Talmor et al. (2020)
- Laden Sie sich die `test` Daten für die Experimente vom 🤗 `datasets` Hub herunter: `KevinZ/oLMpics`
- Für alle Aufgaben gilt:
  - **Dokumentieren** und **interpretieren** Sie Ihre Ergebnisse.
  - **Erläutern** Sie den Experimentaufbau und etwaige Varianten je *kurz*.
  - **Stellen** Sie die Ergebnisse tabellarisch und/oder graphisch **dar**.
  - **Analysieren** Sie Ihre Ergebnisse und **vergleichen** Sie diese mit denen aus Talmor et al. (2020).

1 Masked Language Models **35 + 5 P**

1.1 Multiple-Choice Masked Language Modeling **35 P**

- Implementieren Sie den `MC-MLM` Versuchsaufbau mit der 🤗 `transformers` Bibliothek.
- Führen Sie das `MC-MLM`-Experiment im *zero shot* Aufbau (Talmor et al., 2020, §3.1) mit BERT (`bert-base-cased`) und RoBERTa (`roberta-base`) für die sechs Varianten durch:
  - `Age Comparison`
  - `Objects Comparison`
  - `Taxonomy Conjunction`
  - `Always-Never`
  - `Antonym Negation`
  - `Multi-Hop Composition`
- Analysieren und diskutieren Sie die Ergebnisse *ausführlich*!

1.2 `Age Comparison`: Perturbed Language **5 B**

- Ersetzen Sie die Wörter `age` und `than` in den Prompts des `Age Comparison` Tests durch Nonsenswörter wie `'blah'` und `'da'`.
- Wiederholen Sie das Experiment und dokumentieren Sie etwaige Veränderungen in den Ergebnissen.

Hinweise

- Falls Ihnen kein persönlicher Rechner mit ausreichender Leistung zur Verfügung steht, können Sie Ihre Experimente auf den Rechnern der RBI durchführen.
- Bei beschränkter Rechenleistung können Sie eine „destillierte" Variante von BERT oder RoBERTa verwenden, z.B. `prajjwal1/bert-mini` oder ein `MiniLM` verwenden.

[1] `https://huggingface.co/docs/transformers/index`

2 Generative Language Models 15 + 5 P

2.1 Multiple-Choice Question Answering 15 P

- Führen Sie die beiden `MC-QA` Experimente mit einem generativen LLM durch:
  - `Property Conjunction`
  - `Taxonomy Conjunction`
- Entwerfen Sie dafür zunächst einen passenden *System Prompt*.
- Formatieren Sie die Samples in sinnvolle Prompts und extrahieren Sie den `answerKey` (etwa: A, B, C) aus der Antwort des Modells.
- Evaluieren Sie die Ergebnisse wie in der vorangegangenen Aufgabe.
  - Erhalten Sie immer die Ergebnisse, die Sie erwartet hätten, in dem Format, in denen Sie sie erwartet hätten? Wenn nicht, woran könnte das liegen?

2.2 True Multiple-Choice 5 B

- Konfigurieren Sie Ihre `ChatCompletion` Requests so, dass die Ausgaben des Modells auf die `answerKeys` beschränkt sind.
- Verbessern sich so die Ergebnisse? Beschreiben Sie Ihren Ansatz und evaluieren Sie kurz die Unterschiede zwischen beschränktem und unbeschränktem Prompting.

Hinweise

- Für die Laufzeit der Übung stellen wir Ihnen `llama3.1:8b` über eine REST-API zu Verfügung.
  - Der LLM-Server ist aus dem Universitätsnetzwerk unter `http://llm.lehre.texttechnologylab.org/` zu erreichen.
  - Der Server läuft mit `llama.cpp` und unterstützt die OpenAI Chat-API. Sie können entsprechend die `openapi` Python Library nutzen. Nähreres finden Sie hierzu auch in der `llama.cpp` Dokumentation.
  - Die OpenAI Chat-API ist unter `http://llm.lehre.texttechnologylab.org/v1/` eingebunden und stellt genau ein Modell (`llama3.1:8b`) zur Verfügung. Ein API-Key ist nicht notwendig; zur Verwendung des Clients aus der `openai` Library können sie einen beliebigen String als API-Key verwenden.
- Sie dürfen aber auch jedes andere frei verfügbare Open-Source LLM nutzen, wenn Sie dies entsprechend kennzeichnen.

Literatur

Devlin, Jacob et al. (2018). „BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding“. In: *CoRR* abs/1810.04805. arXiv: 1810.04805.

Liu, Yinhan et al. (2019). „RoBERTa: A Robustly Optimized BERT Pretraining Approach“. In: *CoRR* abs/1907.11692. arXiv: 1907.11692.

Radford, Alec et al. (2018). „Improving Language Understanding by Generative Pre-Training“. In: URL: https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf.

Talmor, Alon et al. (Dez. 2020). „oLMpics-On What Language Model Pre-training Captures“. In: *Transactions of the Association for Computational Linguistics* 8, S. 743–758. ISSN: 2307-387X. DOI: 10.1162/tacl_a_00342.

Tunstall, Lewis (2021). *Natural Language Processing with Transformers*. Hrsg. von Leandro von Werra und Thomas Translators Wolf. O'Reilly Media, Inc. ISBN: 9781098103231. URL: https://ubffm.hds.hebis.de/Record/HEB479538573.

Vaswani, Ashish et al. (2017). „Attention is All you Need“. In: *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*. Hrsg. von Isabelle Guyon et al., S. 5998–6008. URL: https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

Wang, Alex et al. (2019). „GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding“. In: *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=rJ4km2R5t7.