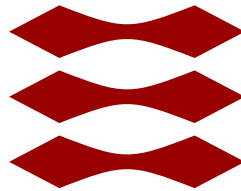


Are GLLMs Danoliterate? Benchmarking Generative NLP in Danish

DTU



TECHNICAL UNIVERSITY OF DENMARK

MASTER'S THESIS IN HUMAN-CENTERED ARTIFICIAL INTELLIGENCE

Søren Vejlgaard Holm, s183911@dtu.dk

Supervised by:

Lars Kai Hansen, DTU Compute
Martin Carsten Nielsen, Alvenir ApS

Project page:

danoliterate.vholm.dk

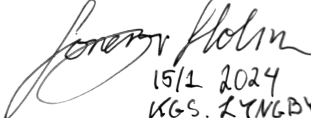
January 15, 2024

Abstract

Generative Large Language Models (GLLMs) like GPT-4 have shown immense potential for business and societal disruptions. However, if practitioners only have tools for measuring model capabilities in English, small language domains like Danish will be left behind by technocapitalistic developments. In this thesis, I present a new benchmark, measuring "Danoliteracy": How capable, calibrated, efficient, toxic and fair GLLMs are across eight diverse Danish Natural Language Processing (NLP) scenarios such as solving citizenship tests or writing helpful social media post replies. Benchmarking 28 models, both openly available and closed, the OpenAI models are found to be uniquely Danoliterate, although open source developments are improving rapidly, especially highlighting the strength of instruct-tuning for zero-shot application. It is concluded that the benchmark, across scenarios, describes one major, underlying dimension of Danish capability, though interesting differences can be seen between model capabilities. Experimental model trainings are presented, shedding light on data considerations for producing a Danish GLLM. The benchmark exemplifies that targeted evaluation in specific domains such as low-resource languages can reveal important, underlying foundation model features that govern how they can be applied.

Preface

This thesis was completed to fulfil the requirements to obtain the Master's degree in Human-Centered Artificial Intelligence at the Technical University of Denmark. The project was produced from August 15, 2023 to January 15, 2024, and was assigned a workload of 30 ECTS points equivalent to around 825 work hours. The project was a collaboration with the AI company Alvenir ApS, alvenir.ai. The resulting code is available at github.com/sorenmulli/danoliterate.



15/1 2024
KGS. ZYNGBY

Acknowledgements I appreciate the helpfulness in the NLP community and give special thanks to Kenneth Enevoldsen, Dan Saattrup Nielsen, Mads Henrichsen and Hailey Schoelkopf for answering all sorts of confused questions. Without the computational resources of the Pioneer Centre for AI, enabled by Nicki Skafte Detlefsen and Henning Christiansen, much less would have been possible. Additionally, I want to thank the colleagues at Alvenir for keeping me on track, and especially the ever-optimistic Martin Carsten Nielsen whose motivational hype, endless stream of literature leads and key insights into AI application were crucial. An exceptional thank you is due Lars Kai Hansen for great project ambition, sharp theoretical inputs and for his central role in the education that has led me here.

Finally, my greatest gratitude is to my wife Simone who, is my reason to always keep going.

Notes on Terminology Before starting, I highlight what kinds of models are discussed in this project. The term GLLM is used in the text, referring to large models capable of solving general textual language tasks without specific adaption. The terminology will be unfolded in due time, but be prepared that this class of investigated models, by some referred to as Generative Language Models, Large Language Models, General Pre-trained Transformers or simply decoders, do not include model families like BERT or T5 among their ranks.

Furthermore, note that I will use the neologism "Danoliteracy" in this project to refer to literacy in Danish, specifically how capable GLLMs are of solving tasks on text in Danish. In Section 2.2.2, this concept will be explored.

Abbreviations

ACL	Association for Computational Linguistics
AI	Artificial intelligence
ASR	Automatic Speech Recognition
BERT	Bidirectional Encoder Representations from Transformer
BPE	Byte-pair encoding
CWR	Contextualized word representation
DDT	Danish Dependency Treebank
DFM	Danish Foundation Models
DL	Deep learning
DNN	Deep neural network
ECE	Expected calibration error
GLLM	Generative large language model
GPT	Generative Pre-trained Transformer
GPU	Graphical Processing Unit
GQA	Grouped Query Attention
HELM	Holistic Evaluation of Language Models
LM	Language modelling
MMLU	Measuring Massive Multitask Language Understanding
ML	Machine Learning
MoE	Mixture of Experts
NER	Named entity recognition
NLG	Natural language generation
NLP	Natural language processing
NLU	Natural language understanding
PCA	Principal component analysis
ROUGE	Recall-Oriented Understudy for Gisting Evaluation
RoPE	Rotary Positional Embeddings
SOTA	State of the art
SWA	Sliding Window Attention
SwiGLU	Swish Gated Linear Unit

Contents

1	Introduction	7
1.1	The Project	8
2	Theory	10
2.1	Modelling Language: From Rules to Generation	10
2.1.1	The World of Natural Language Processing	10
2.1.2	NLP Meets Deep Learning: The Transformer Revolution	15
2.1.3	Performant NLG: GPT Can Write, Now?	18
2.2	Modelling Danish: Encoding and Decoding the Modersmål	21
2.2.1	The State of Danish NLP	21
2.2.2	The Meaning of Danoliteracy	23
2.3	GLLM Evaluation	26
2.3.1	The Problem of Freedom: Evaluation of Generative AI	27
2.3.2	Foundation Models Cheat: Contamination	36
2.3.3	Holistic Evaluation: Accuracy is Not Enough	36
2.3.4	Previous Work on Danish Evaluation	38
3	Review	40
3.1	Danish NLP Scenarios	40
3.1.1	Scenario Taxonomy	40
3.1.2	Existing Scenarios	41
3.1.3	Missing Scenarios	43
3.2	Candidate Models	43
3.2.1	Taxonomy of Models	44
3.2.2	Search Criteria	44
3.2.3	Existing Models	44
3.2.4	Missing Models	51
4	Data	52
4.1	Model Pre-training Data	52
4.2	Evaluation Data	53
4.2.1	Citizenship Test	53
4.2.2	HyggeSwag	55
4.2.3	#twitterhjerne	58
4.2.4	Reading Tests: Cloze Self Test and Gym 2000	59

4.2.5	Existing NLP Tasks: DaNE, AngryTweets and Nordjylland News	61
5	Methods	62
5.1	Baseline Model Training	62
5.1.1	Model Architecture and Transfer Learning	62
5.1.2	Training Approach	63
5.2	Evaluation Models: Who Were Invited?	65
5.2.1	Openly Available Models	65
5.2.2	Closed Models: Black Box Text Generators	66
5.3	Scenarios: What Did I Use?	66
5.3.1	Criteria	67
5.3.2	Prompting	68
5.3.3	Resulting Scenarios	69
5.4	Metrics: How Did I Measure?	71
5.4.1	Inference Modes	72
5.4.2	Similarity Algorithms	75
5.4.3	Metrics of Capability	75
5.4.4	Further Metric Dimensions	77
5.4.5	Uncertainty estimation	79
5.4.6	Average Index	79
5.5	The Open Benchmarking Framework: danoliterate	79
6	Central Results	82
6.1	Model Trainings	82
6.2	The Benchmark	84
6.3	Leaderboards	85
7	Discussion and Analysis	92
7.1	Impact of Further Pre-training	92
7.2	Reactions: How Did the Tables End Up?	93
7.3	Result Correlations and Trends	95
7.3.1	Between Models	95
7.3.2	Between Scenarios	101
7.4	Scenario Experiments: Variations, Ablation	105
7.4.1	Options Presented or Free Generation, NLG or LM	105
7.4.2	Alternative Metrics	108
7.4.3	The Art of Prompting	110
7.5	Predictions: What Do the Models Generate?	113
7.6	What's Next for Benchmarking Danoliteracy?	116
8	Conclusions	119
	Bibliography	120

A Appendix	131
A.1 Scenario Details	131
A.1.1 Scenario Prompt Catalogue	131
A.1.2 Alternative Prompts	133
A.1.3 Augmentation Examples	134
A.2 Training Results	136
A.2.1 Mistral Experiments	136
A.3 Result Details	138
A.3.1 Further Analysis Plots	138
A.3.2 Further Experimentation	139
A.3.3 Further Prediction Inspection	140
A.3.4 Model Generation Examples	142

1. Introduction

Benchmarks shape technologies. Benchmarks act as normative guidelines for technology applications, while also suggesting directions for new developments. Few fields have had as intense a discussion about how to approach practical application and which future developments to pursue as natural language processing (NLP): Artificial intelligence (AI) which can process, understand and generate natural language has the potential to be among the most impactful technologies of our time. As a consequence, this has resulted in a global effort to produce benchmarks that evaluate capabilities, find limits and judge safety in such a rapidly developing field [Lia+22].

The NLP community has adopted an ideal that lower-resource languages should not be left behind by new developments [21]. In Denmark in particular, multiple, ongoing endeavours seek to build up evaluation of which models hold Danish language capabilities; which of the many AIs are Danoliterate. Existing work evaluates models on an array of natural language understanding (NLU) tasks, such as document classification [Bro+21; Nie23a]. So far, the focus has predominantly been on the discriminative language understanding capability of task-specific, fine-tuned models [Nie23a] and not on the inherent Danoliteracy of generative, large language models (GLLMs). The unprecedented success of the public demonstration of ChatGPT has, however, demonstrated that using GLLMs for Danish is feasible [Ols23]. The degree of ChatGPT Danoliteracy and especially how the many, openly available models compare remains largely unexplored.

This comparison is not made any easier by the speed of model development. Since the 2020 release of GPT-3 [Bro+20], and accelerating in 2023, there have been frequent releases of GLLMs. Such natural language generation (NLG) models seek to generate coherent, human-like and useful textual language for tasks such as virtual assistance, automatic content creation and summarisation. The Klondike environment in which models are trained by actors such as tech giants [Tou+23b], governments [Sah22], open-source collaborations [Wor+23], specialized AI firms [Dey+23], and even individuals [Har23] results in a diverse plethora of models. They vary in

many ways: From lighter, compute-efficient versions to behemoths with hundreds of billions of parameters; from models with heavy censorship of potentially toxic output to community versions intentionally removing all guardrails; and from strictly English efforts to massively multilingual projects that parse all languages on the internet.

Benchmarking these NLG models is an open field of research. The immensely transformative commercial opportunities of performant models [Chu+23; Bri23] induce pressure on NLP practitioners to answer crucial questions about GLLM use: What performance can be expected from a given model in a specific application scenario? What are the costs of applying a model of a given size? How risky is it to use the artificially generated text?

The Center for Research on Foundation Models at the Stanford Institute for Human-Centered Artificial Intelligence approaches this benchmarking undertaking with the goal of holistic evaluation. By taxonomising potential application scenarios of NLP, Percy Liang, Rishi Bomassani, Tony Lee, et al. identify seven metrics and 16 core scenarios for Angloliterate NLP models in "Holistic Evaluation of Language Models" (HELM) [Lia+22]. A taxonomy of benchmarks can help find the under-represented scenarios and metrics that gives a complete view of model differences.

The work of HELM focuses on English. I will, however, argue that this structured benchmarking approach is even more important when working in lower-resource languages such as Danish. Here, there is a lower number of available corpora and a more confused situation concerning which models possess Danish competencies. These factors require the evaluator to make harder resource prioritisation between scenarios and metrics. Thus, the task of inferring the state of GLLMs in Danish requires the production of a holistic suite of scenarios that test model generative metrics across dimensions such as capability, efficiency and robustness.

1.1 The Project

This project seeks to produce a benchmark for GLLMs in Danish. It is my goal that the benchmark can guide the expectations of NLP practitioners about what to expect from this class of large, general models.

Due to the limited existing work on Danish NLG, the goal of benchmarking GLLMs is not just one of standardisation of existing corpora and models: To improve benchmark coverage, datasets must be mutated or produced anew to include

missing scenarios and cover metrics other than accuracy. Furthermore, note that, in addition to evaluation of existing models, novel model trainings are carried out to expand modelling insights.

To produce a structured benchmark, the work will have the following component goals:

1. To review existing, Danish benchmark scenarios and to describe them according to a taxonomy of scenarios.
2. To review and also taxonomise existing GLLMs that may be Danoliterate.
3. To use the scenario review to identify missing scenarios and to contribute new corpora to remedy the scarcity.
4. To use the model review to identify gaps in NLG models and to carry out new trainings to get insight into how to fill them in.
5. To combine the different models and scenarios into a standardised, open-source benchmark that answers important questions for practitioners.
6. To build a practical and usable open-source framework that allows for further evaluation and displays a live leaderboard.

Contents Initially, Chapter 2 reviews theoretical concepts used in the project in the broader field of NLP and then specifically GLLMs, followed by the relation to Danish, finally considering the literature on GLLM evaluation. Continuing in the reviewing mood, Chapter 3 presents results of reviewing both the existing Danish NLP evaluation scenarios and GLLMs that are candidates for Danoliteracy.

Chapter 4 introduces data used in the project, both for GLLM training experiments and, centrally, the data used to create evaluation scenarios, followed by Chapter 5 explaining which training and evaluation methodologies were used. Crucially, Chapter 6 presents the trained models, the contents of the final benchmark, as well as leaderboards showcasing results of the many evaluated models. Chapter 7 then reacts to these results and presents several ablations and experiments, discussing what was learned about models and benchmarking. Finally, Chapter 8 summarises findings.

2. Theory

2.1 Modelling Language: From Rules to Generation

These sections introduce foundational concepts from NLP and specific details of GLLMs used in the report. The explanation is built up chronologically, starting with rule-based NLP, going over deep learning (DL) to ending with modern GLLM developments.

2.1.1 The World of Natural Language Processing

Linguistics, AI and NLP The foundational science of linguistics charts all dimensions of human language. This includes the cognitive processes that underlie language, cultural contexts that steer formulations and lingual grammatical or structural rule sets [23c]. The 20th-century developments of this field have often been characterised as divided between two approaches to language. The rationalist position, often personified by Noam Chomsky, entails that human language ability is predominantly pre-established before being shaped by senses; this knowledge is innate, ingrained or, with a computational metaphor, hard-coded by evolutionary processes. Empiricists, on the other hand, worked from the assumption that language understanding is learned by a mind that starts with more general principles of pattern recognition; not specific lingual knowledge [MS99, Sec. 1].

Discussions with a similar blend of fundamental philosophical views of epistemology can be found across many fields. One of them is the comparatively young academic field of AI. Here, the rationalist linguists can be compared to the symbolic subfield of AI: Creating automatic reasoning and problem-solving by modelling real-world problems according to explicit rules and knowledge representations. However, the current wave of DL and AI success is relatable to the empiricist stance. The idea of using machine learning to achieve AI follows the sub-symbolic or statistical paradigm where machine learners let the intelligence mainly be shaped by empirical inputs: large datasets [RN09].

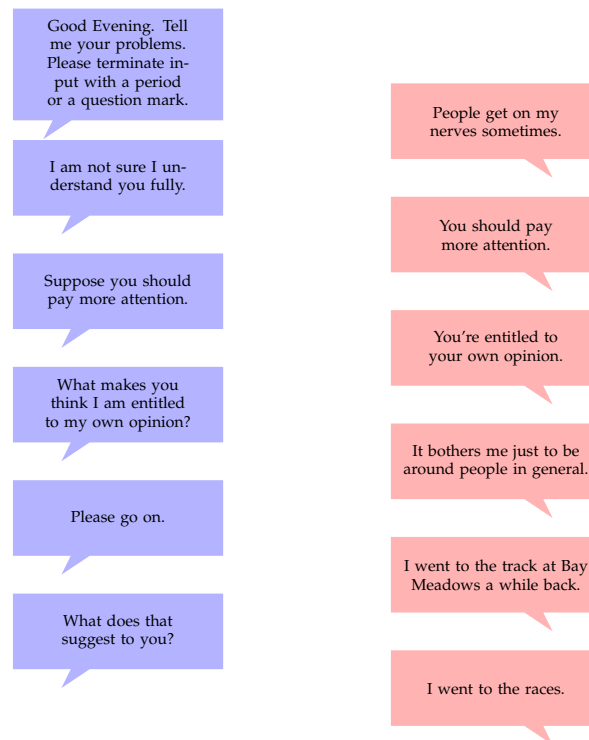


Figure 2.1: ChatGPT's symbolic great-grandparents: The Doctor chat(ter)bot ELIZA (blue) and the paranoid schizophrenic patient chatterbot PARRY (red) conversing over ARPANET in 1972 [73]. ELIZA, created in 1964 by Joseph Weizenbaum, performed NLG by maintaining an ontology of keywords each mapping to a transformation of the input text to a reply, reflecting part of the given words back to the patient [Wei66].

Symbolic NLP It should be of no surprise that, when linguistics and AI meet, the same division is found. NLP is an interdisciplinary field combining linguistics and computer science to algorithmically process natural language by understanding, manipulating or generating textual language ^{2.1}. Ever since the birth of AI as an academic field, it has been linked to linguistics specifically as Turing used language as medium for testing intelligence [Tur50]. Symbolic NLP, following the rationalist, Chomskyan approach, has, with a heyday in the 1970s, 80s and 90s, achieved great success in rule-based parsing of language and structuring real-world information digitally [MS99, Sec. 1]. One of the more amusing successes of this technology is the

NLG capabilities of early chatbots, then called chatterbots [73], shown in Figure 2.1.

It is not straightforward to directly apply the rule-based algorithms of symbolic NLP on the immediate data format of text. Digital texts represented as strings of binary encoded characters in formats such as ASCII or Unicode are designed with cross-platform data transfer and storage in mind; not for encoding linguistic meaning. In NLP, the general approach to converting input strings into useful representations is to perform tokenisation: The process of subdividing the input text into a list of units. In symbolic NLP, each unit generally corresponds to a lexical token, that is, a full word [MS99, Sec. 4.2.2]. Early work used the simple concept of graphic words: Split the text at whitespace characters [KF67]. Further work on lexical natural language tokenisation has used hand-crafted regular expressions such as for the Penn Treebank [MSM93].

Simple string division methods do not normalise the representation of a word which can occur in multiple inflected forms, yielding a significant challenge for the rulesets. To overcome this, the produced tokens can be further reduced by rules that transform words to their base stem such as the 1980 Porter stemmer [Por80]. The rule sets can be complex, using context to stem a word by performing morphological analysis, transforming words to their base depending on a large vocabulary through part of speech analysis. This process is referred to as lemmatisation [MS99, Sec. 4.2.3]. See Figure 2.2 for exemplification of different methods and some of their issues.

^{2.1}In many definitions of NLP, there is no constraint excluding spoken language. Automatic speech recognition and other multimodal language AI might thus be included under the umbrella of NLP, but in this thesis, I consider textual language exclusively.

Whitespace Tokenisation in English:

It had a perfectly round door like a porthole, painted green.

Trebank Tokenisation:

It had a perfectly round door like a porthole, painted green .

Porter Stemmed:

it had a perfectli round door like a porthol , paint green .

SpaCy Lemmatised:

it have a perfectly round door like a porthole , paint green .

Whitespace Tokenisation in Danish:

Det havde en cirkelrund dør ligesom et køje, der var malet grøn.

Trebank Tokenisation:

Det havde en cirkelrund dør ligesom et køje, der var malet grøn .

Porter Stemmed:

det havd en cirkelrund dør ligesom et køj , der var mal grøn .

SpaCy Lemmatised:

det have en cirkelrund dør ligesom en køje, der være male grøn .

Figure 2.2: Lexical tokenisation, stemming and lemmatisation of words in both English and Danish. Danish Porter stemming is done using the Snowball implementation [Por01]. Lemmatisation is performed using the rule-based lemmatisers implemented in the SpaCy framework [Hon+20]. Issues can arise when applying the principles developed for English text to Danish text, for example around the word "cirkelrund". This word is not tokenized into its compound parts, a process some symbolic NLP pipelines might depend on. Generalizing rule-based NLP from English to languages with very different syntactic structures such as Chinese, which does not delimit with whitespace, is even more problematic.

Statistical NLP As digital data availability and machine compute capability rose in the 1990s and 2000s, statistical NLP largely began to dominate the applied field. Here, NLP systems process language according to statistical models inferred on large corpora [MS99, Sec. 13]. Amongst successes in machine translation by e.g. IBM alignment models, this statistical approach had multiple lasting technical contributions, one such being statistical language modelling. A language model (LM) \mathcal{M} assigns probabilities to single tokens τ (such as "perfectly") or to sequences of tokens;

$$\sigma = [\tau_1, \tau_2, \dots, \tau_N]. \quad (2.1)$$

The type of language modelling that I will mainly consider is the computation of the conditional probability of a token being the next token in the sequence:

$$p_{\mathcal{M}}(\tau|\sigma) = P(\tau_{N+1} = \tau|\tau_1, \tau_2, \dots, \tau_N, \mathcal{M}) = P(\tau|\tau_1, \tau_2, \dots, \tau_N, \mathcal{M}). \quad (2.2)$$

In statistical NLP, this problem is approached using n -gram LMs \mathcal{M}_n . Here, it is not attempted to condition over the entire sequence but only the last $n - 1$ tokens such that

$$\begin{aligned} p_{\mathcal{M}_n}(\tau|\sigma) &= P(\tau|\tau_{N-n+2}, \dots, \tau_N, \mathcal{M}_n) && \text{e.g.} \\ n = 1: \quad p_{\mathcal{M}_1}(\tau|\sigma) &= P(\tau|\mathcal{M}_1) && \text{(unigram)} \\ n = 2: \quad p_{\mathcal{M}_2}(\tau|\sigma) &= P(\tau|\tau_N, \mathcal{M}_2) && \text{(bigram)} \\ n = 3: \quad p_{\mathcal{M}_3}(\tau|\sigma) &= P(\tau|\tau_{N-1}, \tau_N, \mathcal{M}_3) && \text{(trigram).} \end{aligned}$$

These values can be calculated using corpora by observing and storing the frequency of n -grams as

$$P(\tau_{N+1} = \tau|\tau_{N+2-n}, \dots, \tau_N, \mathcal{M}_n) = \frac{C(\tau_{N+2-n}, \dots, \tau_N, \tau)}{C(\tau_{N+2-n}, \dots, \tau_N)}, \quad (2.3)$$

where $C(\sigma)$ is the number of occurrences of the token sequence σ in a training corpus. There are multiple details to this implementation such as how to handle n -grams never seen before or avoiding division by zero, but this approach is defining for the overarching idea of utilizing big datasets to describe language using general statistical models and not explicit rules [JM23, Chap. 3], [MS99, Sec. 6.12]. The chain rule of probability makes it possible to compute a joint model probability of an entire sequence which is often used with n -gram LMs for automatic speech recognition (ASR), spelling correction or machine translation [JM23, Chap. 3]:

$$\begin{aligned} p_{\mathcal{M}_n}(\sigma) &= P(\tau_N|\tau_{N-n+1}, \dots, \tau_{N-1}, \mathcal{M}_n)P(\tau_{N-1}|\tau_{N-n}, \dots, \tau_{N-2}, \mathcal{M}_n) \cdots P(\tau_1|\mathcal{M}_n) \\ &= \prod_{i=1}^N P(\tau_i|\tau_{i+1-n}, \dots, \tau_{i-1}, \mathcal{M}_n). \end{aligned}$$

n -gram LM is exemplified in Table 2.1.

σ	Det havde en cirkelrund dør ligesom et køøje, der var malet			
$p_{\mathcal{M}_3}(\sigma)$	$10^{-42.9}$			
τ	grøn	gul	grøøn	islagkage
$p_{\mathcal{M}_3}(\tau \sigma)$	$10^{-4.6}$	$10^{-4.9}$	$10^{-6.6}$	$10^{-6.3}$
$p_{\mathcal{M}_3}(\sigma, \tau)$	$10^{-47.5}$	$10^{-47.8}$	$10^{-49.5}$	$10^{-49.2}$

Table 2.1: LM in Danish using the Alvenir 3-gram DSL model [NJ19] over four different completions to a text where typos and surprising completions lower the joint probability considerably.

These foundational NLP methods are used in a task-centric manner, which has also shaped the field as it is used today [Lia+22, Sec. 3.1]. While different task solutions can share tokenisation or be connected through NLP pipelines such as SpaCy [Hon+20], specific algorithms are generally created for each downstream task, such as Named Entity Recognition (NER) or Sentiment Analysis. These two tasks are briefly introduced below.

NER requires token-level classification. Given an input text, predict which tokens belong to Named Entities: real-world objects with rigid definitions such as specific persons, locations or organisations [SD03]. A symbolic NLP solution could be to implement rules such as taking capitalised words followed by "Street" as locations or to maintain entity lexicons for matching. Sentiment Analysis also requires predictions based on text, but on document level instead of token level. Here, the algorithm must assign a specific class or a numerical value to the emotional tone of a text for example on a negative to positive valence scale. Symbolic rulesets such as VADER (Valence Aware Dictionary and sEntiment Reasoner) [HG14] or statistical algorithms such as SentiWordNet [BES10] show high performance on the languages that they were produced for.

2.1.2 NLP Meets Deep Learning: The Transformer Revolution

DL and RNNs Veni. Vidi. Vici. Starting around 2010, the statistical AI subfield DL increasingly emerged victorious across many fields, carrying hype, attention and results [Sej18, Chap. II]. DL, the use of Deep Artificial Neural Networks (DNNs) for Machine Learning (ML), had existed since the 60s but the gains in available data as well as computationally fast implementations gave rise to this DL revolution [con23, "Deep Learning Revolution"]: Here, DNNs started dominating ML competitions

in computer vision (CV) and ASR [Sej18, Sec. 9]. Work was underway to create a similar revolution in NLP, a field already shaped by statistical learning and ML.

Long short-term memory (LSTM) was one of the main characters in the DL revolution by dramatically improving the state of the art (SOTA) on ASR [con23]. This DNN architecture, invented in 1997 by Hochreiter and Schmidhuber, was a further development of Recurrent Neural Networks (RNNs) which implement DL on sequential data [HS97]. This is done by outputting a vector representation at each time point that is sequentially passed on as model state input along the next input in the sequence. RNNs, LSTMs and successors can then be implemented for NLP by letting this input sequence be a sequence of tokens. Such a network can be turned into a language model by, at each step, predicting the next token based on the state vector representation. This LM capability can architecturally be described as adding an LM decoder prediction head to the model.

However, for any DNN to take token input or generate token outputs, the tokens must be represented numerically. DNNs generally take categorical inputs using one-out-of- K encodings. This can be applied directly for tokenisation, maintaining a vocabulary of K , for example 50,000, tokens each mapping to an ID $\in \{t | t \in \mathbb{N}, 1 \leq t \leq K\}$. When decoding, the LM decoder head performs K -dimensional classification. Usually, the first step of modelling is the embedding layer where a \mathbb{R}^D vector is learned for each token ID. Here, a fundamental shared principle of ML is used on language; representing data in some abstract vector space, here called word embeddings in D dimensions, the model hidden size [JM23, Chap. 6]. How the embeddings and all following model parameters are learned is now the important question. For RNNs, application to textual documents was limited by the fact that modelling context over sequences of increasing length made gradients increasingly unstable, vanishing or exploding for finite-precision implementations [JM23, Sec. 9.5]. Though LSTMs improved upon this problem considerably, another architectural feature was required to master the sequential nature of text.

Transformers The Transformer, presented in 2017 by Vaswani et al., models textual context using the self-attention mechanism [Vas+17]. Here, an entire sequence of N input embeddings are considered simultaneously; dot products are then computed between input vectors and learned weight matrices, which, along with the SoftMax operation produces N output vectors. The output vector at index i encodes the information contained in input vector i along with a mix of the other vectors

in the sequence – mixed according to how the model has learned to attend across sequences. The architecture contains multiple important details such as multiple self-attention computations with different weight matrices being performed (multi-head attention), the input embeddings being combined with positional IDs for the model to know token order and each self-attention layer being followed with linear and normalisation layers. However, the resulting transformer is nothing more than a stack of these self-attention blocks; there is no recurrence through time. Thus, a transformer encoder takes in N token IDs and returns N word representations $\in \mathbb{R}^D$, now called contextual word representations (CWRs) because the nice transformer has encoded contextual meaning through the self-attention mechanism. The specific transformer which brought the DL revolution to NLP in full force was such an encoder; one with the friendly name BERT (Bidirectional Encoder Representations from Transformers) [Dev+19].

BERT BERT produces CWRs. While giving BERT your document and receiving 512 768-dimensional vectors might seem like a strange exchange, it is in fact a very useful one. BERT cemented the foundation model approach to NLP: Single, large encoders are trained for one or more languages on huge text corpora using pre-training objectives optimising the learned CWRs as meaningful representations. When having to solve downstream tasks, practitioners can now fine-tune the model by adding prediction heads after the encoder, mapping the vectors to a desired output representation. Pre-training is thus a compute and data-intensive task produced by high-resource organisations like Google, the creator of BERT, while task adaption through fine-tuning can be achieved on smaller, specialised corpora [JM23, Chap. 11].

Before we meet the GPT family, I end on a note on tokenisation: What are these K tokens in the vocabulary? Are they words? No, in modern transformers, tokens are subwords. Statistical algorithms like the byte pair encoding (BPE) learn optimal ways to divide byte sequences into tokens based on occurrences in training corpora [SHB16]. Much modelling impact is contained in these important parts of transformers as different algorithms such as Wordpiece, Unigram and Sentencepiece, different vocabulary sizes $K = 32,000$ or $50,000$ or adaption to different languages influence results considerably [Hug24].

English words:

It had a perfectly round door like a porthole, painted green.

BPE Tokenisation:

It had a perfectly round door like a por th ole , painted green .

Danish words:

Det havde en cirkelrund dør ligesom et køje, der var malet grøn.

BPE Tokenisation:

Det hav de en cir kel rund d ør l iges om et ko ø je , der var

ma let gr ø n .

Figure 2.3: BPE tokenisation of words in both English and Danish using the LLaMa 2 tokenizer [Tou+23a]. Note that this primarily English tokenizer uses more tokens per word in Danish than in English.

2.1.3 Performant NLG: GPT Can Write, Now?

Transformer Decoders The transformer can do more than encode. In fact, it turns out that Vaswani’s mantra that “Attention is all you need” [Vas+17] certainly holds for NLG. To generate language, the model is implemented as a decoder. In Transformer World, this means that self-attention is computed unidirectionally, only attending to tokens to the left, dubbed causal attention, as opposed to BERT where the B stands for the bidirectionality of attention for CWRs. The attention thus imposes soft weights over the previous tokens in the sequence, treating them as semantic context for what token may come next. The last layer feeds into a LM decoding head producing the K -dimensional token distribution for the next token in the sequence. A token is chosen and added as a generated text to the sequence which is then fed into the transformer again, autoregressively continuing NLG [Rad+18].

How is that token chosen in each step? The greedy decoding approach selects the token assigned the highest probability. As an alternative to this maximum likelihood prediction approach, tokens can be sampled from the language distribution with the probabilities predicted by the transformer decoder LM. The concept of SoftMax σ temperature T allows for interpolation between these two cases of greedy decoding (sometimes called temperature 0) and distributionally accurate decoding (temperature 1) by modifying the probability distribution to increase the likelihood of most

likely tokens as temperature decreases [CKA23, Sec. 3.2.3] as

$$\sigma(\mathbf{h})_i = \frac{\exp^{h_i/T}}{\sum_{j=1}^K \exp^{h_j/T}} \text{ for } T > 0, \quad (2.4)$$

where \mathbf{h} is the K -dimensional final LM decoding head activation.

GLLMs and Instruction Fine-tuning Combine the NLG transformers and the idea of large, pre-trained foundation models and you enter into the GLLM universe. The Generative Pre-trained Transformer (GPT) championed this approach, as OpenAI in multiple papers proposed the generative angle on NLP [Rad+18; Rad+19]: Instead of fine-tuning encoder models to specific NLP tasks, use a single GLLM for everything. You then let the generative model continue NLG from a textual input, called a prompt, designed in such a way that the model solves your desired task using patterns learned during pre-training. Can this really work effectively? The 2019 release of GPT-2 showed a new level of coherence in the world of NLG. Through demonstrations like "Write With Transformer"^{2,2}, this 1.5 B parameter model, designed with normalisation of both input and output to transformer blocks for improved training stability, impressed in its prose quality on English text but a lack of factual accuracy was noted. Furthermore, it was a tricky game to prompt this base GLLM, foundationally trained to perform LM on general text, towards solving specific tasks [Rad+19].

The next generation of GPTs remedied these shortcomings with moonshot-level success. In 2020, OpenAI announced a 175 B parameter GPT-3 [Bro+20]. The architecture was generally the same, though the self-attention mechanisms alternated between standard, dense computations, and some sparse mechanisms such as each computation only attending to the last few tokens. The big change was the pre-training data scale, including 500 B tokens of web data, encyclopaedia and book data. While OpenAI claimed strong results and allowed for API usage of the model after a delay, an additional technique was required before GPTs saw mass adoption.

The idea of instruct-tuning was implemented in InstructGPT adding additional training that steered model outputs towards treating input prompts as questions to be answered or task descriptions to be carried out [Ouy+22]. This was done through techniques of supervised fine-tuning, learning from reference pairs of instructions

^{2,2}transformer.huggingface.co/doc/gpt2-large

and good answers, as well as reinforcement learning with human feedback, where a signal of users choosing the best NLG outputs is used as a reward signal for model optimisation. In this process, model alignment is carried out, as the instruction dataset and human feedback together steer model behaviour towards a certain set of norms. The 2022 release of the ChatGPT demonstration, using a non-descript GPT-3.5 instruct-tuned for multiturn chatting, had a profound impact on the attitude towards GLLMs. A plethora of base and instruction-tuned GLLMs were produced and released by big actors such as OpenAI, Google and Meta as well as open source collaborations. I return to the GLLMs in 3.2.3 but will instead here highlight the setup that these models allow for NLP practitioners.

Apart from being helpful assistants, entertaining chatters and a source of much AGI acceleration disagreement, the chief quality of performant GLLMs is that they allow for unparalleled adaptability. Requiring no access to specialised corpora, practitioners can quickly prototype the feasibility of any task that can be formulated through natural language. OpenAI describes GLLMs as few-shot learners, being able to solve specific tasks when only prompted with few or no (zero-shot learning) examples of how to do an assignment [Bro+20]. GLLMs can be considered be a useful starting point in the NLP innovation pipeline, allowing rapid prototyping and task development before performing data collection and specialised modelling.

GLLM Architectural Moves Finally, I end the NLP section with a short overview of architectural tweaks to the GPT architectures used in most GLLMs. While this field is data-centric, chiefly focusing on acquiring large, high-quality corpora, architectural changes are used for marginal performance gains, improving model runtime efficiency and stabilising model training.

- **Model quantisation:** Due to the size of GLLMs, reducing the numeric accuracy of model weights to 16, 8, 4 or even 2 bits per parameter is an active field of research. This technique is often used for inference after a full-precision (or 16-bit) version of the model has been trained [Hug23a].
- **MoE (Mixture of Experts):** MoE, dividing problems into multiple subtasks with specialised systems, has a rich history in DL. For GLLMs, MoE can be implemented such that only a subset of the many billions of parameters are executed for each token, making large model sizes possible. An initial routing system decides which subset of specialised feed-forward layers are available for a given input [She+23].

- **SWA (Sliding Window Attention):** A specific implementation of sparse self-attention, reducing quadratic complexity of dense attention by limiting the mechanism in some transformer blocks only to a sliding window of surrounding tokens [Chi+19].
- **RoPE (Rotary Positional Embedding):** RoPE is a specific way to implement positional embeddings using trigonometric functions which allows for representing both relative and absolute token positions in a way that is compatible with efficient transformers [Su+21].
- **GQA (Grouped Query Attention):** An efficiency-improving trick, GQA is a modification of multi-head attention that reduces the number of weight matrices by grouping tokens through part of attention. This is done to speed up the inherently serial process of decoding [Ain+23].
- **SwiGLU (Swish Gated Linear Unit):** A training stability and convergence modification, replacing the classical ReLU (rectified linear unit) as activation function between transformer blocks with the SwiGLU Swish activation function combined with a Gated Linear Unit [Tou+23a].

Finally, some modern GLLMs, like GPT-4 or Gemini, take multimodal inputs allowing vision or audio data as prompting in addition to natural language. I enumerate which models use which of these techniques in Section 3.2.3.

2.2 Modelling Danish: Encoding and Decoding the Modersmål

2.2.1 The State of Danish NLP

The wonderful world of NLP opportunities was not ignored by the Danes. An active field of research and application, Danish NLP is characterised by the adaption of successful SOTA methods from English to Danish, coping with fewer available resources through cooperation and engineering. The progress can, in addition to following the Academic literature, be tracked through multiple open source projects such the Alexandra Institute framework DaNLP, compiling models, datasets and benchmark results [Bro+21]. Also, the benchmark ScandEval, developed by Danish Dan Saattrup Nielsen, serves as an overview of Danish NLP progress [Nie23a]. As a benchmark, I return to it in Section 2.3.4.

Considering the results reported across such frameworks, it can be seen that Danish NLP has well and truly felt the results of the DL revolution. Danish encoder models outperform multilingual BERTs on Danish benchmarks [Nie23a] and focused efforts to create Danish foundational encoders that, across many downstream tasks, are performant when fine-tuned have been successful in recent years. I am thinking especially of the Danish Foundation Models (DFM) collaborative project, anchored in the University of Aarhus, Centre for Humanities computing. In addition to successes in ASR when partnering with private companies like Alvenir, the project has produced SOTA NLP encoders for Danish^{2,3}, having gotten access to a large pre-training corpus from the Royal Danish Library [Ene+23].

Compared to this encoder success, the focus on GLLMs in Danish is newer and less developed as the GLLM review in Section 3.2.3 will show. As GPT-2 was not multilingual and GPT-3 behind closed doors, public attention towards the use of transformers for generating Danish text was limited before the release of ChatGPT. The Danoliteracy of ChatGPT as qualitatively experienced through the public demo was reported as a surprise [Ols23]. Whether other GLLMs hold Danish capabilities and how they compare to ChatGPT can be difficult to ascertain for practitioners, as details of training data languages are sparse or, as for OpenAI models, non-existent. Motivated by such issues and the growing focus on business perspectives of generative AI, there is much interest in and debate about how to produce a Danish GLLM. The idea has its detractors but in December 2023, the chairman of the committee for digitalisation and IT proposed that the Danish government fund the training of a GLLM [BD23]. In her proposal, she argued^{2,4}: “There is a difference between whether the language model primarily is familiar with Thanksgiving, American Football, the Fourth of July and pickup trucks or if it primarily builds upon Christmas Eve, handball, Fifth of June and Christiania bikes.”

Whether this proposal will be accepted and how and by whom a Danish GLLM will be produced is uncertain. Work is underway in the DFM project to coordinate how to reach this goal, and private enthusiasts have already started building and presenting experimental models to which we return in Section 3.2.3. When deciding

^{2,3}As of Jan. 2., the DFM Encoder tops the Danish part of the ScandEval leaderboard at scandeval.github.io/mainland-scandinavian-nlu-benchmark.

^{2,4}Translated from Danish: “Der er forskel på, om en sprogmodel primært kender til thanksgivingtraditionen, amerikansk fodbold, 4. juli-fejring og pickuptrucks, eller om dens primære udgangspunkt er juleaften, håndbold, den 5. juni og christianiacykler.” [BD23, “Bemærkninger til forslaget”]

how to proceed with such ideas, the guiding question must be: What is the goal of such a GLLM? What can be achieved when specialising a decoder to Danish? Or in other words; what are the important qualities that make up Danoliteracy?

2.2.2 The Meaning of Danoliteracy

Danish is a North Germanic language, part of the mainland Scandinavian group of mutually intelligible languages, and is spoken by about 6 million people, primarily in Denmark [Wik23]. In the following, I consider what it means for these 6 million jolly individuals to be able to speak Danish. Focusing on textual language, to be considered literate, abilities are required on multiple levels. An immediate distinction is that between skills in reading and writing. To be more precise, the OECD definition of literacy used in adult competence surveys [OEC19] states that literacy is “The ability to understand, evaluate, use and engage with written texts to participate in society, achieve one’s goals, and develop one’s knowledge and potential.” In addition to different modes of engaging with written texts, this definition stresses the *function* of language use: How language must be operationalisable as part of goal-oriented behaviour.

It is a fundamental human question, and perhaps the defining task of linguistics, to reveal the constituent parts of language ability. To find high-level, practical building blocks, I turn to language pedagogy. Traditional approaches to language teaching followed the structural view which emphasised competencies to produce structured language products such as grammatically correct sentences [Arw10]. Currently, the interactive view is dominant, stressing social use of language in e.g. conversational exchanges [RR01]. The Common European Framework of Reference for Languages taxonomises aspects hereof in e.g. range, (“(...) flexibility reformulating ideas in differing linguistic forms to convey finer shades of meaning precisely (...)”) or coherence (Ability to “(...) produce clear, smoothly-flowing, well-structured speech, showing controlled use of organisational patterns (...)”) [Cou24]. Finally, it must be noted that such abilities do not just depend on understanding the language itself but also require knowledge of a shared world of facts and values. The main governmental Danish language test requires an understanding of cultural and societal Danish concepts in addition to pure language competencies [23b].

The requirements imposed by these understandings of what it means to be literate in a language are shown in Table 2.2 where I have also translated them into the

GLLM setup. Thus, the word Danoliterate describes models which generally fulfil these requirements in Danish. Note that these span multiple levels of Bloom's taxonomy; Danoliteracy is not just knowledge and comprehension as can be captured by NLU models; higher-level analysis and synthesis abilities must be held before a GLLM is literate in the same way we expect of humans.

	Required of Humans	Required of GLLMs
Modes Structure	Read and write. <ul style="list-style-type: none"> • Maintain active and passive vocabularies. • Understand and use grammatical concepts. 	Perform NLU and NLG. <ul style="list-style-type: none"> • Tokenise texts. • Model contextual meanings of words. • Attend to correct sequence-level dependencies, considering grammar.
Function	Use language to <ul style="list-style-type: none"> • participate in society, • solve professional tasks and • achieve personal goals. 	<ul style="list-style-type: none"> • Generally be able to produce intentional text. • Solve specific NLP tasks.
Interaction	Converse coherently with range.	<ul style="list-style-type: none"> • Follow instructions. • Decode subtle communication. • Inform when the response is uncertain.
Culture	Understand and use cultural contexts and connotations in language.	<ul style="list-style-type: none"> • Decode meanings that rely on cultural references. • Generate text that is correct according to common sense logic. • Generate real-world factually correct text. • Generate culturally and socially acceptable text.

Table 2.2: Angles on Danoliteracy requirements based on the above discussion. While a simple formulation could be requiring models to hold Danish NLG capabilities to a convincing and useful level, these views reveal more complexity.

Returning to Danish, there are some specific effects of modelling this language. Being a low-resource language, the multilingual interactions are interesting. The mainland Scandinavian languages are generally intelligible and especially Norwegian Bokmål carries great similarities to Danish, motivating some NLP practitioners to work on joint Scandinavian NLP [Nie23a]. As a modern effect, the English language has had a great impact on modern Danish: English words or phrases are lent, creating "Danglish" text full of anglicisms, and in many fields, the only technical terms known in Denmark are the English ones [Waa10]. This has been reported as a stronger trend in Denmark than in neighbouring countries [Nie07].

This English leak into Danish is good news for GLLMs, often adapted from high-resource American-English contexts. A more complex job for GLLMs is the cultural aspect of language. Adapting models that may be expected to be primarily aligned to American culture may be more tricky for languages in more distant cultural spheres than for Danish^{2.5}. However, the Danish language, so closely connected with the country of Denmark, does carry a specific historical and cultural context. Apart from knowing the facts that make up this context, Danoliteracy requires working within cultural norms about ethics, humour and formality. Finally, the function of language is also dependent on such context: There is e.g. no tradition for writing computer code in Danish, meaning that this task is less important for Danoliteracy than for example scenarios relevant for how a welfare state can process citizen data.

Easy task! To be Danoliterate, GLLMs only have to be able to handle two modes of language use, master structure, use language functionally and interactively as well as feeling at home in Denmark. Luckily for these poor models, we shall find that evaluation of this quality is more tricky than discussing it.

2.3 GLLM Evaluation

In this section, theoretical considerations and previous work on evaluating GLLMs are described. First, a longer subsection describes overarching generative AI evaluation ideas and how they are used in modern English GLLM evaluation. Then, the specific issues of dataset contamination and holistic evaluation are described,

^{2.5}One might be tempted to trust the Inglehart-Welzel World Cultural map based on the World Values Survey, reducing the world of culture into two axes. I can report that Denmark and the US have an Euclidean distance of 1.6, higher than Denmark – Netherlands (0.5), lower than Denmark – Estonia (2.2). [IW23]

finishing on the relation to Danish specifically.

2.3.1 The Problem of Freedom: Evaluation of Generative AI

Generative Evaluation is Hard Creativity is often perceived as one of the finest qualities of humanity. It is certainly one of the hardest to measure: Assessing the quality of artworks is notoriously contentious – but even the quality of more structured creations such as Master’s theses, PowerPoint slideshows, tables and computer code involves considerable subjectivity. The problem does not get any simpler when we start generating imagery, music, speech, video or language automatically. In all these free forms, generation is a problem without an innate metric. Crudely put: If it was known how to deductively measure the best possible generations, why train large, generative AI systems if one could search for the maximum of this wonderful metric?

Categories of Evaluation It might then not be useful to think of evaluation as a pursuit of pure metrics that a priori measure the inherent quality of a generation. Instead, evaluation can be seen as the task of predicting how highly humans would judge the system on its capability to solve a given task in a given scenario. This empirical approach allows for a wide range of methods in different domains. Below, I divide these into seven high-level ideas for the evaluation of generative AI with examples of applications.

1. **Human feedback:** The most direct way to approximate the general human judgement is to average across observed human feedback to generations. If the experiment is designed representatively, this method retains maximal evaluation validity but might incur high costs.
 - In evaluation of AI-generated art, questionnaires have been carried out where human participants rated generations across subjective dimensions such as “Beauty” “Liking” and “Arousal” [ZK23].
 - In the public ChatGPT demo chat.openai.com, OpenAI collects all user interactions including explicit evaluation feedback to the generations which is given through thumbs up and down buttons [Pen23].
2. **Human discrimination:** Instead of getting a signal of quality for each generation, experiments can be designed requiring humans to discern between automatic generations and reference data examples. This evaluation will thus be

relative to some test data in a setup comparable to A/B testing. Running such evaluations can be compared to carrying out Turing tests.

- GPT-4 has been evaluated for marketing copywriting using a discrimination approach. Here, humans, when comparing texts, preferred the generations of the AI – unless they were told that one of the texts was AI-generated, in which case they preferred the other text [ZG23].
3. **Model-based feedback:** The human quality signal of Idea 1 can be replaced with a model, returning quantitative feedback to the generation example. The evaluation validity is here dependent on the degree to which the feedback model approximates human feedback.
 - The Public OpenAI LLM Benchmark “Evals” [Ope23a] uses a combination of methods. A large portion of evaluation tasks are implemented through “model-graded evals”, sometimes using the same model as the generative model and feedback model as “... we’ve found that GPT-4 is surprisingly capable of checking its own work.” according to the GPT-4 release blog post [Ope23b, Sec. “OpenAI Evals”].
 - In the field of image generators, especially in the work of Generative Adversarial Networks (GANs), the inception score is an example of model-based feedback: Here, a higher score is given to the generator if a separate image classifier has high confidence in a single class prediction for each example while still assigning diverse labels to a population of generations [Xu+18, Sec. 2.2].
 4. **Model-based discrimination:** Instead of a model feedback score in itself, one can use a binary classifier to predict whether a given generation is from reference, gold standard data or created by the generative model.
 - GANs uses the discriminator idea inside the training paradigm itself by training to generate images minimally distinguishable from reference, human images according to an adversarial network [Xu+18, Sec. 2.1]. Reporting the predictive accuracy of the adversarial network is an example of such an approach. For GANs, this has also been done using simple binary classifiers such as 1-nearest neighbours [Xu+18, Sec. 2.2].
 5. **Probabilistic modelling of reference data:** The generative AI might not just be a data generator but also a probabilistic model of data. For such models,

including LMs, given reference data, it can be expected that the model assigns high likelihood to examples that are considered high quality or correct. This method is heavily dependant on the reference data distribution, crucially requiring representative data but is reasonable in the sense that it will often correspond to the loss function used in training.

- When pretraining LMs, the inverse of the joint likelihood of sequences in an evaluation corpus is often reported. This measure, perplexity, only requires gold standard data and no task specific experiment design but it has received criticism for being influenced by non meaningful text patterns [Wan+22], encouraging non-creative generations [Keu20], and being model dependant [ABB06].
6. **Algorithmic comparison to reference data:** Instead of getting feedback through a human or another model, generations can be computed on a set of examples with corresponding, gold standard reference generations. A similarity score can then be performed algorithmically. I also refer simple pattern-matching output parsing with the umbrella term of comparison. This method might be feasible and straight-forward but is dependant on reference data and comparison scheme for validity. If the generative model is probabilistic, the comparison method can use model probabilities as well as the actual generations.
- For the NLG task of abstractive summarisation, evaluation is often performed by comparing model summaries with reference, human texts. The ROUGE similarity metric is generally used, though it has been criticized for only capturing surface-level similarities, missing semantic content [ABK22]. Similar approaches are followed for machine translation, typically using the BLEU metric [Pap+02].
 - The 🤖 Huggingface Open LLM Leaderboard combines seven benchmarks using the Eleuther AI LM Evaluation Harness [Gao+21]. These are generally implemented as question answering tasks where the output likelihoods are used to calculate model answer accuracy [Fou+23].
7. **Algorithmic self comparison:** For some task types, it is possible to evaluate without reference data by comparing model generations under different inputs along with an assumption about e.g. generation invariance to specific input changes.
- The robustness benchmark PromptBench evaluates NLG models based on

whether small typos or prompt reformulations change the model output [Zhu+23].

Thus, there are many ways to evaluate generative AI and which one to choose might often depend on what tools are available to the evaluators. Which ingredients are required for which ideas are shown in Table 2.3 and it is noted that each of these design choices is a source of evaluation bias which must be considered for validity.

Method depends on:	Inputs	Humans	References	Other model	Compar.
1. Human feedback	✓	✓	✗	✗	✗
2. Human disc.	✓	✓	✓	✗	✗
3. Model-based feedback	✓	✗	✗	✓	✗
4. Model-based disc.	✓	✗	✓	✓	✗
5. Prob. modelling	✓	✗	✓	✗	✗
6. Alg. comparison	✓	✗	✓	✗	✓
7. Self comparison	✓	✗	✗	✗	✓

Table 2.3: Seven evaluation ideas for generative AI and whether their results depend on input data (e.g. prompts), human participation, reference data (e.g. gold standard generations), inputs from another model, and comparison algorithms (e.g. similarity metrics or output parsers).

GLLM Evaluation Review Having enumerated categories of generative AI evaluation, I turn to GLLMs specifically and review evaluation literature for benchmarking NLG in English.

For many NLP tasks or subdomains, canonical benchmarks have existed for a long time: SQuAD for extractive question answering [Raj+16] or CoNLL-2003 for NER [SD03]. But these only evaluate a single scenario. Designing general GLLM benchmarks that cover many aspects of NLG is a newer discipline. Many efforts follow ideas also found in NLU benchmarks such as SuperGLUE released in 2019 by Wang, Pruksachatkun, et al.: Gather a maximal number of diverse and at-the-time difficult benchmark scenarios, aggregating model task performance to a single leaderboard [Wan+19, Sec. 3]. In the following, I refer to such collections as *scenario compilations*. In SuperGLUE, Wang, Pruksachatkun, et al. chose to exclude many NLG tasks due to reservations about the automatic comparisons of Idea 6^{2.6}. This influential NLU benchmark can be described as solved by modern AI: Multiple

models that appear at the leaderboard achieve superhuman performance [Wan+19, super.gluebenchmark.com].

A trickier benchmark, more usable for evaluation of NLG models, was initially released in 2020 by Hendrycks, Burns, et al.: "Measuring Massive Multitask Language Understanding" (MMLU) [Hen+21]. Here, the Academic knowledge of models across 57 subjects such as "Virology" or "High School US History" was measured using an approach that has since become defining: Use multiple-choice questions and calculate model accuracy by comparing model output with the correct choice. In the MMLU paper, this method, a special case of Idea 6, was implemented by presenting the possible options for the model (A-D), taking the completion with maximum model continuation likelihood as the prediction. Both zero-shot and 5-shot evaluations were tested. See example on Figure 2.4 for an example.

Note that MMLU is inherently an NLU benchmark which could as well be adapted for text classification using encoders. However, at release, the authors found that it required a GLLM to perform above random guessing [Hen+21, Sec. 4]. Adapting NLU benchmarks such as MMLU for NLG is a way to handle the problem of freedom: The generative task is reduced to a simpler, discriminatory, multiple-choice setting where valid results can be extracted easily. The benchmark does then not evaluate formulation, fluidity or structure of NLG but more general knowledge and language understanding acquired during pretraining. Incidentally, these latter qualities are often considered the hardest parts of NLG, vindicating this approach.

```
1 The following is a multiple choice question about high school
  mathematics.
2 If 4 daps = 7 yaps, and 5 yaps = 3 baps, how many daps equal 42 baps?
3 (A) 28 (B) 21 (C) 40 (D) 30
4 Answer:
```

Figure 2.4: An example of an MMLU-style evaluation prompt. This example from the MMLU High School Mathematics section [Hen+21, Figure 1] is here shown in zero-shot and options-presented form.

MMLU is currently widely used as an NLG benchmark. It is implemented in the 2023 scenario compilation benchmark "🤖 Huggingface OpenLMM Leaderboard"

^{2.6}"Tasks must have an automatic performance metric that corresponds well to human judgments of output quality. Some text generation tasks fail to meet this criterion due to issues with automatic metrics like ROUGE and BLEU" [Wan+19, Sec. 3.1].

[Bee+23]. Here, seven benchmarks with a focus on model knowledge can be run by the community using the EleutherAI LLM Evaluation Harness [Gao+21] where tasks are generally implemented MMLU-style using maximal model likelihood. There are, however, subtle implementation differences of the same scenario across implementations of Idea 6 highlighted by Fourrier, Habib, Launay, and Wolf: The OpenLLM Leaderboard takes maximum likelihood over the entire option text instead of only the letter (e.g. (C) 40 instead of (C)) [Fou+23]. Furthermore, small differences in prompting such as whether or not to include the text *The following is a multiple choice question about high school mathematics* in Figure 2.4 changed results [Fou+23]. This open leaderboard is part of an incredibly active open source NLG community (I count 1982 submitted models across 485 🧑🏻 Huggingface users), working together to improve NLG capabilities of models with freely available model weights – see Figure 2.5.

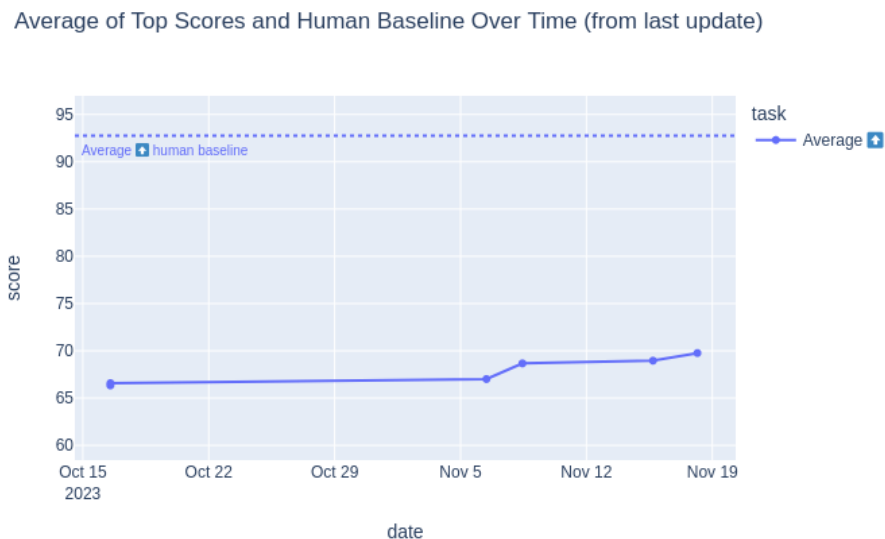


Figure 2.5: A month of open source progress on the 🧑🏻 OpenLLM Leaderboard. As of November 30th, 2023, the SOTA model is called TigerBot 70B chat v2 [Tig23] but someone has probably fine-tuned a better one while I typed this sentence. The human baseline is estimated highly on this benchmark with Academic and expert scenarios [Bee+23].

In 2022, The Stanford Center for Research on Foundation Models released HELM, a benchmark with more focus on attaining well-rounded model insight than being a

fierce competition. HELM is a large scenario compilation benchmark (> 48 scenarios, includes MMLU) but was not designed as a GLLM-specific benchmark [Lia+22]. Many NLG tasks are however included using Idea 6: Here, answers are evaluated not primarily through likelihood but by answer parsing, allowing for "quasi exact match" [Lia+22, Sec. C.1.1]. This means HELM, the original MMLU paper and the 🤖 Huggingface OpenLMM Leaderboard might all yield different values for MMLU performance for the same model [Fou+23]. In this review, I primarily take note of the approach to holistic evaluation in HELM which will be described in Section 2.3.3.

The empirical approach to evaluation motivates building benchmarks as scenario compilations of high quantity to cover all that can be formulated in language. Both Google and OpenAI have approached this quantity task by successfully turning benchmark production into community projects on GitHub. Google's resulting 2022 benchmark, BIG-bench, contains 204 tasks all validated by a Google team according to 10 review criteria including "Not solvable by memorizing the Internet" or "Difficulty" [Sri+23, Sec. 2]. See Figure 2.6 for task keywords. Operationally, the benchmark was divided into two implementations [Sri+23, Sec. 2.1]:

- "JSON tasks" with a simple input/target pair allowing for algorithmic comparison of prediction and target using e.g. ROUGE (Idea 6).
- A smaller set of "programmatic tasks" that allow for more complex, possibly multiturn model interactions defined by the contributor.

Along with the release of GPT-4, OpenAI released their benchmark "Evals" which also became a popular community project with users submitting tasks as diverse as rotating Tetris cubes or detecting sarcasm, totalling 447 tasks as of November 2023 [Ope23a]. A huge scenario compilation, this framework generally uses Idea 4, model-based discrimination: Letting GPT-4 grade the text generation according to a grading prompt defined in the task definition. Notably, OpenAI Evals only supports OpenAI models and the evaluation results are not published. Both projects contain non-English or multilingual scenarios, though I found Danish text in neither (but small amounts of Swedish).



Figure 2.7: The Chatbot Arena benchmark interface [Zhe+23] at chat.lmsys.org. I selected "B is better" and it was revealed that my Model A was OpenHermes 2.5 Mistral 7B and B was Claude 1. This gamification of the human discriminator approach (Idea 2) allows for computing model Elo ratings at scale [Zhe+23].

Also in 2023, the GLLM evaluation field responded to the rise of chat models with the production of chat-specific benchmarks. LLM-Eval, like OpenAI Evals, attempts to define a canonical evaluation schema for model-based feedback (Idea 3) of chat helpfulness across the dimensions appropriateness, content, grammar, and relevance [LC23]. AlpacaEval, popular in the open source, appears more influential. Here, Idea 4 is utilized by measuring model NLG capability by how often a GPT-4 based discriminator, shown to correlate to human judgement, prefers the given model generation over reference GPT-3 Davinci answers [Li+23b]. MT-benchmark and Chatbot Arena were also released in 2023 as a pair, evaluating multiturn chatting where the former uses model feedback dubbed "LLM-as-a-judge" (Idea 3) and the latter is a competitive platform where humans choose between generations from different chatbots (Idea 2), see Figure 2.7 [Zhe+23]. Finally, the late 2023 Meta AI benchmark GAIA uses simple answer parsing (Idea 6) but points towards a multimodal future of generative AI where assistant systems are required to use many plugin functionalities including computer vision and data analysis [Mia+23].

2.3.2 Foundation Models Cheat: Contamination

Apart from the inherent difficulty in creating evaluation tasks, GLLM evaluation poses another, unique challenge: Pre-trained on huge, web-spanning corpora which are often not released, it can be a difficult problem to find a test dataset that has not been contaminated by inclusion as training data. Public benchmarks such as the 🤖 OpenLLM leaderboard have experienced submission of contaminated models [Par23]. Contamination might often be accidental and difficult to detect. One mitigation used in Google BIG-bench is to prepend random strings of letters and numbers, called "canary strings", when releasing benchmark data [Sri+23]. When performing future evaluations, model outputs can be analysed. If a model generates the canary string or assigns it a high likelihood, it is suspected of contamination. Contamination can be related to a family of evaluation problems described by Goodhart's law, often stated as "When a measure becomes a target, it ceases to be a good measure." [con24].

While the contamination risk is minimal for novel benchmarks, building on data not part of large web crawls, this problem must be handled continuously for open benchmarks. Surprising benchmark improvements should be met with some suspicion.

2.3.3 Holistic Evaluation: Accuracy is Not Enough

When applying GLLMs to a practical use case, one's initial thought might be how capable it will be in solving its task: Is the model sufficiently Danoliterate to do this job? Will it often make mistakes? Are the questions too hard for it? This is, however, only the first round of questions: Further risks exist, beyond the difficulty of the task. In HELM, the Stanford group identified six other dimensions beyond how capable the models are in accurately solving their given task [Lia+22]. Here, these general requirements, also called desiderata, were chosen based on a review of goals in AI conferences [Lia+22, Sec. 4.1]. I enumerate these metric dimensions, motivating their relevancy for the practical use of GLLMs.

Efficiency Few things are free in this world and GLLMs, as the first L implies, are especially resource-intensive systems. NLP practitioners generally work in compute-limited scenarios where model capability must be measured against computational cost, often indicative of financial cost. Efficiency can be measured in inference time

but an important, underlying concern is how the electricity usage of large GPUs translates into energy consumption: LLM energy usage is uncertain but experts estimate ChatGPT per query consumptions in the scale of watt-hours [Lud23b] and note that GPT-4 carbon footprint is orders of magnitude larger than for GPT-3.5 [Lud23a]. Such disparities must be taken into account when comparing systems. Furthermore, low-efficiency models that require enterprise-level compute are only available to high-resource organisations, limiting the societal impact of releasing capable GLLMs.

Calibration When using GLLMs for prototyping, practitioners might often work in scenarios where the access to gold-standard evaluation data is limited: It might be difficult to get quantitative signals early on which can reveal the expected capability of the model. If the model is able to accurately estimate its own uncertainty, this estimate can be used to guide development. A desired quality for GLLMs is thus that generated text which is likely to hold poor Danoliteracy should have a lower likelihood than accurate generations. This requirement of calibration can be even more useful if the GLLM is in a production setting. Here, low-confidence generations can be monitored for error analysis, data drift and problematic uses of the system. In such a system, more accurate uncertainties enable more opportunities for model improvement using methods such as active learning [Lim23].

Robustness While model capability might be computed on standard inputs and datasets, robustness refers to how much the model drops in performance when the input includes errors, complexities and non-standard formats. This dimension attempts to take a step from a clinical, Academic, best-case setup to the more messy instances of real-world data that GLLMs might be subject to.

Toxicity GLLMs have been shown to generate offensive text even when prompts do not encourage this [Geh+20]. Both foul language as well as controversial or offensive statements can be joined under the umbrella term of toxicity. Practitioners applying GLLMs must consider the risk of these unpredictable systems exposing users to language that offends them, reducing user trust and comfort.

Fairness While accuracy metrics might be satisfactory on a population of test examples, there might be underlying differences in capability depending on specific

characteristics of examples. It has been documented that many NLP systems perform disparately across gender [SA21] and race [Koe+20]. If thorough analysis of drops in performance across such population groups is not made as part of the evaluation, conclusions on GLLM performance can be drawn that only apply to a subset of the population, presumable population groups that were well represented in training data. An attitude that only focuses on average model results, ignoring whether the system performance is distributed fairly, may lead to exclusive technology, inhibiting population groups from participating in an increasingly digitalised society.

Bias Compared to fairness, evaluation of bias in NLP is more complex: Here, the HELM metric refers to the underlying values, norms and language choice that the model is biased towards [Lia+22, Sec. 4.7]. Evaluating this requires analysis of generated text, searching for problematic stereotypes or unwanted assumptions. While some GLLMs, especially chat- and instruction-finetuned ones, are released with alignment training, which seeks to control model bias, practitioners might question which political, ethical and cultural assumptions were encoded into the model both explicitly by its developers and implicitly by training on huge, uncurated corpora.

2.3.4 Previous Work on Danish Evaluation

For smaller language domains like Danish, evaluation resources are fewer both in terms of data availability and investment activity. This poses unique challenges and requires harder evaluation prioritisation.

Danish: ScandEval No benchmark targeting GLLMs exists in Danish. However, GLLMs have been included in Danish in at least one public benchmark. In 2023, the ScandEval benchmark added GLLMs to the evaluation across Mainland Scandinavian NLU tasks. This benchmark, created by Dan Saattrup Nielsen, evaluates models across different canonical NLP tasks [Nie23a]. There are plans underway to add NLG tasks to the benchmark. In Danish, four task scenarios are supported as of December 2023:

- DaNE: The central Danish benchmark on NER [Hvi+20].

- Angry Tweets: A sentiment classification benchmark on a Twitter dataset [Bro+21, Sec. 4].
- ScaLA-Da: A linguistic acceptability task, requiring binary classification of whether sentences are grammatically correct [Nie23a, Sec. 4.3].
- ScandiQA-Da: A partly manual, partly automatic translation of the extractive question-answering dataset MKQA [Nie23a, Sec. 4.4] [LLD20].

These scenarios will be revisited in Section 3.1.2, and the two first are included in the Danoliterate benchmark and explained in Section 4.2.5.

In its current form, ScandEval thus focuses on discriminative language tasks, being topped by fine-tuned BERT-like encoder models as of December 2023.

Other Low-Resource Languages In Norway, a GLLM benchmark has been produced as part of the large NorwAI project which also trains a new Norwegian GLLM, NorGPT [Liu+23]. This benchmark, called NLEBench, evaluates models across NLU-tasks, summarisation and instruction tasks [Liu+23, Fig. 1]. Though multiple included scenarios were translations, the authors also produced Norwegian-first datasets, stressing the importance of including Norwegian culture in the evaluation. As of January 2024, neither a public leaderboard, apart from the results reported in [Liu+23], nor the benchmark framework or data appear to have been released.

In Sweden, GLLM evaluation was also discussed as part of a large, national GLLM production project. While building GPT-SW3, developers lacked a standardised benchmark and turned to model perplexity, model translation ability and qualitative prompting for evaluation [Ekg+22, Sec. 5].

In other low-resource languages, practitioners have started exploring GLLM evaluation through the translation of English benchmarks such as the Serbian LLM Eval. project [Ale23].

3. Review

3.1 Danish NLP Scenarios

In this section, existing Danish NLP evaluation scenarios will be reviewed.

3.1.1 Scenario Taxonomy

To describe general trends across available resources, a taxonomy on what makes up an evaluation scenario is used. Scenarios are categorised by their textual content domain, their NLP task and the used language. Content domain is described by the data source for the evaluation corpora. As NLP is a task-focused field, and Danish NLP evaluation has focused on discriminative NLU tasks, each scenario is categorised by a canonical NLP task. To describe these tasks, the HELM taxonomy is used which is based on Association for Computational Linguistics (ACL) submission tracks [Lia+22, Sec. 3.1]. See Table 3.1 for this track/task taxonomy. In my categorisation, I will sometimes add slightly more task detail such as discerning between extractive and abstractive question answering.

Track	Tasks
Dialogue and Interactive Systems	Chit-chat dialogue, task-oriented dialogue
Discourse and Pragmatics	Discourse parsing, sentence ordering, coreference resolution
Ethics and NLP	Toxicity and hate speech detection, misinformation and fake news detection
Generation	Data-to-text generation,
Information Extraction	Named entity recognition, entity linking, entity extraction, relation extraction, event extraction, open information extraction
Information Retrieval and Text Mining	Information retrieval and passage retrieval
Language Grounding to Vision, Robotics and Beyond	Image captioning, visual question answering, instruction following, navigation
Machine Learning for NLP	Language modelling
Machine Translation and Multilinguality	Machine translation
Phonology, Morphology, and Word Segmentation	Tokenisation, lemmatisation,
Question Answering	Question answering and reading comprehension
Semantics: Lexical	Word sense disambiguation, word sense induction
Semantics: Sentence-level	Semantics, Textual Inference, and Other Areas
Sentiment Analysis, Stylistic Analysis, and Argument Mining	Sentiment analysis, style transfer, argument mining, stance detection, opinion mining, text simplification
Speech and Multimodality	Text-to-speech, speech-to-text
Summarisation	Summarisation, sentence compression
Syntax: Tagging, Chunking and Parsing	POS tagging, chunking, constituency parsing, dependency parsing, grammar induction, grammatical error correction

Table 3.1: The HELM task taxonomy from [Lia+22, Tab. 1] based on ACL conference submission tracks. For brevity, tracks that were not given canonical tasks such as “NLP Applications” are here omitted.

3.1.2 Existing Scenarios

Scenarios were found primarily by consulting the framework DaNLP [Bro+21] which contains a compilation of datasets^{3.1}. More scenarios were found on 🤖 Datasets Hub [Wol+19] and the Awesome Danish NLP list [Nie23b]. The below compilation cannot claim completeness but is intended to give an accurate impression of the characteristics of available resources.

See Table 3.2 for the full enumeration of scenarios.

^{3.1}github.com/alexandrinst/danlp/blob/master/docs/docs/datasets.md

Scenario	Task	ACL Track	Domain	Language
DaNewsroom [VS20]	Summarisation	Summarisation	News outlet crawls	Danish
Nordjylland News [Kin23]	Summarisation	""	TV2 Nordjylland articles	Danish
ScandiQA [Nie23a]	Extractive QA	Question Answering	Google search questions, Wikipedia	Scand. (trans.)
DKHate [SD20]	Hate speech detection	Ethics and NLP	Facebook and Reddit comments	Danish
DanFEVER [ND21]	Misinformation detection	""	Wikipedia, other encyclopaedia texts	Danish
DaWinoBias [KB21]	Toxicity detection'	""	Constructed by researchers	Danish (trans.)
AngryTweets [Bro+21]	Sentiment analysis	Sentiment and Stylistic Analysis, Argument Mining	Tweets	Danish
LCC Sentiment [Nie16b]	Sentiment analysis	""	Diverse web texts	Danish
Europarl Sentiment [Nie16c]	Sentiment analysis	""	European parliament proceedings 1996-2011	Danish
Twitter Sentiment [Bro+21]	Sentiment analysis	""	Tweets	Danish
OpenSubtitles 2018 [LT16]	Machine translation	Machine Translation and Multilinguality	Subtitles	Danish + 61 lang.
EU Bookshop [Tie12]	Machine translation	""	EU Reports	Danish + 47 lang.
Europarl7 [Koe05]	Machine translation	""	European parliament proceedings 1996-2011	Danish + 20 lang.
ParaCrawl6 [Bañ+20]	Machine translation	""	European online sites and publications	Danish + 44 lang.
ITU Faroese Danish [DID22]	Machine translation	""	Europarl, Dimmaletting newspaper	Danish + Icelandic
DaNE [Hvi+20]	NER	Information Extraction	1983-1992 literature	Danish
DaN+ [PJG20]	NER'	""	1983-1992 literature	Danish
DaNED [LW23]	NER'	""	1983-1992 literature	Danish
WikiANN [Pan+17]	NER	""	Wikipedia	Danish + 281 lang.
DDisco [Fla+22]	Discourse parsing	Discourse and Pragmatics	Reddit posts and Wikipedia articles	Danish
Dacoref [KL04]	Coreference resolution	""	1983-1992 literature	Danish
ScaLA [Nie23a]	Grammatical error correction'	Syntax	1983-1992 literature	Scand.
DDT-POS [Kro03]	POS tagging	""	1983-1992 literature	Danish
DDT-DP [Kro03]	Dependency Parsing	""	1983-1992 literature	Danish
DDT-NP [Kro03]	Chunking	""	1983-1992 literature	Danish

Table 3.2: The compilation of existing Danish NLP evaluation scenarios taxonomised using the ACL tracks and tasks. A marker ' on task indicates that a more precise task description could be used but that the closest task in the taxonomy is reported here. "" is used to indicate that the track is the same as above.

Initially, it is noted that the many scenarios in the 1983-1992 literature domain all stem from the Danish Dependency Treebank (DDT) [Kro03] based on the PAROLE corpus which sampled short texts from book, newspaper, periodical and miscella-

neous sources [NKA98].

Considering the tasks types, the fields of sentiment analysis, machine translation, NER and syntax all appear to have the most attention by scenario volume. Important NLP fields like summarisation and especially question answering appear less studied. Considering domain, apart from the many DDT corpora, the general text sources are encyclopaedic, from social media or from news texts.

The table representation shows each scenario as equal. However, some tasks have seen much more activity than others. For example, the NER benchmark DaNE [Hvi+20] is included in ScandEval [Nie23a], is used in DanLP [Bro+21] and has been the target of multiple projects [KN21; EHN21; HS21].

3.1.3 Missing Scenarios

Initially, it is identified that there are a limited number of scenarios naturally suited for evaluating general GLLMs: ScandiQA is the only question answering dataset and is extractive, such that it benchmarks discrimination of the part of the input containing the answer. Furthermore, all text in the task is translated (parts of it automatically), which Nielsen mentions as a limitation of the dataset due to the fact that "many of the questions and answers are concerned with topics specific to the USA" [Nie23a, Sec.4, 4]. Finally, only the two news summarisation datasets measure Danish NLG, all the other non-translation scenarios are NLU tasks.

Turning to the domains, I would argue that there is a high degree of diversity in tone: Both formal (encyclopaedic, legal, news) and informal (social media) text scenarios can be found. Conversational text can however only be seen in the subtitles dataset. Higher coverage could perhaps be achieved on the textual *context*: No corpora appear to describe language produced in e.g. a commercial context or set in an educational setting. The nature of the semantic contents of the corpora is less clear, as few scenarios are set in a specific topical setting. One possible limit with domain coverage in this regard is topics that are specific to Danish culture as multiple of the scenarios are parallel or translated corpora.

3.2 Candidate Models

In this sections GLLMs that are candidates to hold Danoliteracy are reviewed.

3.2.1 Taxonomy of Models

Models are categorised by their architecture and size: For the latter, following literature, the total number of trained parameters is used. Furthermore, their availability or possible details on license is described. Finally, any available information on training dataset language is included as well as a categorisation between instruct-tuned models versus base pre-trained models.

3.2.2 Search Criteria

As the goal of this benchmark is to benchmark Danoliterate GLLMs, only large, generative general models are considered. The following search criteria were thus employed:

- *The model must be generative:* Hold the capability to produce text given an input.
- *The model must be an autoregressive decoder language model.* This excludes encoders like BERT or encoder-decoder architectures like T5. Many ideas of this thesis could be extended to text-to-text encoder-decoder architectures but these are excluded to limit scope.
- *The model must be general:* Trained to generate text across domains and tasks, not specifically fine-tuned for one downstream scenario.
- *There must be some hope that the model may hold Danoliteracy.* Other mainland Scandinavian models are included as previous Danish NLP has seen successful results using e.g. Norwegian models [Nie23a]. Multilingual models trained on large internet corpora with a small subset of Danish are included even if they do not present Danish as one of their features. However, models specifically described as holding only monolingual, non-Scandinavian capabilities are excluded.

3.2.3 Existing Models

Models were found using literature search on terms like "multilingual LLM", "Danish language model", "Scandinavian GPT", and using the 🤖 Model Hub [Wol+19].

Below, the found GLLMs are described, grouped into sections of similar models.

Closed Models OpenAI maintains a lineup of GLLMs that are not publicly available but can be used for inference through the OpenAI API Platform [Ope23c]. Apart

from Chat models used in the backend of ChatGPT, instruct and base models are available. The latter are not original GPT-3 weights, which are retired, but newer developments. The models are shown in Table 3.3 where it can be seen that training details are very limited.

There have been much speculation about model sizes: For the original GPT-3 Davinci, OpenAI disclosed it as having 175 B parameters, and the Babbage version 1.3 B parameters [Bro+20]. There, the trail goes cold and especially GPT 3.5 Turbo is unclear though a withdrawn Microsoft preprint paper listed it as just 20 B parameters [Sin+23]. It has been rumoured that GPT 4 is a MoE architecture with a total of 1.76 trillion parameters (8×220 B) [WF23; PW23].

Responding to ChatGPT success, Google released a long-awaited competitor in December 2023 named Gemini [Tea+23]. The largest version, Gemini Ultra, was claimed to perform at or above GPT-4 level but this model was not initially released to the public due to safety concerns. No details on pretraining or architectures were given, apart from the fact that the model is multimodal across visual and audio inputs, and multilingual with a certain list of languages supported which includes Danish.

Model	Size	Arch.	Type	Avail.	Release Date	Language	
GPT 4 Turbo	4	Unk.	Trans. Decoder (Multi-mod., possibly MoE)	""	Closed	Nov. 2023	Unk. Multiling.
GPT 4	""	""	""	""	Mar. 2023	""	
GPT Turbo	3.5	""	Trans. Decoder	""	""	""	
GPT Turbo Instruct	3.5	""	""	Instruct	""	Nov. 2022	""
Davinci 002 (GPT Base)	""	""	""	Base	""	Jul. 2023	""
Babbage 002 (GPT Base)	""	""	""	""	""	""	""
Gemini Pro	Unk.	Trans. Decoder (Multi-mod.)	Chat	Closed	Dec. 2023	Unk. Multiling.	

Table 3.3: The closed, though available, models considered as Danoliteracy candidates. Note that the naming for the non-chat OpenAI models is complex with multiple versions used with the same name: I refer to the newest version with that name as of December 2023. The table also reflects the state of the available models at this time; models behind an API might be added, updated or retired.

Multilingual Open Source Models Following a February 2023 limited release of the LLaMa (Large Language Model Meta AI) family of LLMs [Tou+23b], Meta AI made a fully open July release of their second iteration of 2023 LLMs: These models, dubbed LLaMa 2 [Tou+23a], have, due to a license allowing commercial use and impressive results, been massively influential in the open source community of improving LLMs^{3.2}. Both base versions and chat-finetuned versions were released in different model sizes.

^{3.2}See e.g. the Reddit community [reddit.com/r/LocalLLaMA](https://www.reddit.com/r/LocalLLaMA) dedicated to building personal chat assistants based on the LLaMa models. The subreddit has 100K subscribers as of December 2023.

89% of pre-training data was in English, but Swedish, Norwegian and Danish were all included as training languages, making up a joint 0.2% of data [Tou+23a, Table 10]. Meta themselves formulate the model as monolingual “with a bit of additional data from 27 other languages”, noting “We do not expect the same level of performance in these languages as in English”^{3.3}.

Later in 2023, the French GLLM startup Mistral openly released a lineup of similar models. Compared to the LLaMa 2 models, the Mistral model architecture differed in the use of SWA [Jia+23]. No details were given on pre-training data or languages but due to architectural differences and a new license, I assume that the model cannot be trained from LLaMa checkpoints. A larger version, Mixtral, using an MoE architecture, was later released [tea23]. Finally, the Mistral 7B model weights were in December 2023 used to produce the SOLAR 10.7 B models, using a novel weight upscaling methodology followed by instruct-tuning on new datasets that included OpenAI model outputs thus not allowing commercial use of the produced model [Kim+23]. The authors reported the instruct model as surpassing the larger Mixtral model across different scenarios [Kim+23, Table 2]

From the earlier work on multilingual open source GLLMs, the mGPT models, produced in the AI Forever collaboration are noted as being trained on 61 languages including Danish and Swedish [Shl+22].

This category of models can be seen on Table 3.4.

^{3.3}From the Meta LLaMa 2 FAQ as an answer to “Does LLaMa 2 support languages outside of English?” at ai.meta.com/llama/faq (Visited Dec. 30 2023)

Model	Size	Arch.	Type	Avail.	Release Date	Language
LlaMa 2 7B	6.7B	GPT-3 w. SwiGLU, RoPE	Base	Partly open (Custom license)	Jul. 2023	English + some multilingual
LlaMa 2 13B	13B	""	""	""	""	""
LlaMa 2 70B	70B	"" + GQA	""	""	""	""
LlaMa 2 7B Chat	6.7B	GPT-3 w. SwiGLU, RoPE	Chat	""	""	""
LlaMa 2 13B Chat	13B	""	""	""	""	""
LlaMa 2 70B Chat	70B	"" + GQA	""	""	""	""
Mistral 7B	7.2B	LlaMa 2 w. GQA, SWA	Base	Open (Apache 2.0)	Sep. 2023	Unk.
Mistral 7B Instruct	7.2B	""	Instruct	""	""	""
Mixtral Mixtral Instruct	8 × 7 B ""	"" + MoE ""	Base Instruct	"" ""	Dec. 2023 ""	"" ""
SOLAR 10.7B	10.7 B	LlaMa 2 w. GQA, SWA	Base	Open (Apache 2.0)	Dec. 2023	Unk.
SOLAR 10.7B Instruct	10.7 B	LlaMa 2 w. GQA, SWA	Instruct	Partly open (Non-commercial)	Dec. 2023	Unk.
mGPT	1.3B	GPT-3-like	Base	Open (Apache 2.0)	Apr. 2022	61 languages
mGPT-13B	13B	""	""	Open (MIT)	Apr. 2023	""

Table 3.4: Publicly available multilingual models. The custom LlaMa 2 license, the "Llama 2 Community License", allows for certain uses of the model, including commercial ones, but imposes a number of restrictions including formulations about the use of the model to improve other LLMs [Met23]. The SOLAR 10.7 B Instruct non-commercial license is CC BY-NC 4.0 Deed [Com24].

Scandinavian Models Of available Danish GLLMs, two examples exist of small-scale adaption of English models. Kenneth T. Martinsen released an adaption of EleutherAI's GPT Neo 1.3B [Bla+21] using the Cross-lingual Progressive Transfer Learning method [OR23] dubbed GPT Neo Danish [Mar23b]. A 1.5 B parameter model named GPT Neox Da. also exists, released by Peter Schneider-Kamp on 🤖 Huggingface model hub with little details [Sch24]. Another individual effort, Roman Jurowetzki QLoRA-finetuned the Zephyr-7b-alpha instruction model [Hug23b], which is an adaption of Mistral 7B, on the Danish Gigaword [Der+21] dataset, naming the resulting model "Kannelsnegl" [Jur23].

An example of a possibly fully Danish GLLM has also been presented: The model DanskGPT, privately developed at great personal cost^{3,4} by Mads Henrichsen, has been showcased as a demo at chat.danskgpt.dk [Hen23a]. The model is, however, not released openly. Henrichsen has described it as a LLaMa 2 13B model trained on a personally web-scraped 3 B word Danish dataset, followed by fine-tuning on an instruction dataset constructed using the "Self-Alignment with Instruction Back-translation" [Li+23a] method [Fle23; Jun23]. Instead, a tiny base version has been released, trained from a tiny LLaMa "on 8 B tokens of Danish, synthetic text" [Hen24].

Furthermore, Henrichsen has publicly released a Mistral 7B model trained on a dataset from the Danish forum "Hestenettet", dubbed HestenettetLM [Hen23b].

Turning to the Scandinavian neighbours of Denmark, both in Norway and Sweden, central projects with public funding have released GLLMs. The Norwegian National Library AI Lab have released the NB GPT-J models trained on a large, Norwegian web corpus [Kum+21]. An instruct-tuned model was also released, fine-tuned on a translated version of the Alpaca instruction dataset [Tao+23]. In Sweden, a large national project was started to create a Swedish GPT. The AI Sweden centre released the resulting GPT-Sw3 models in many versions [Sah23]. These were trained on a large corpus also including Norwegian, Danish, Icelandic and English text.

^{3,4}Henrichsen reports having spent DKK 100,000 by September 2023 [Fle23]

Model	Size	Arch.	Type	Avail.	Release Date	Language
GPT Neox Da.	1.5B	GPT 3-like + RoPE	Unk.	Open (Apache 2.0)	Apr. 2023	Unk.
GPT Neo Danish	1.3B	GPT 3-like	Base	Open (MIT)	Sep. 2023	English + Danish
Kanelsnegl (v. 0.2)	7.2B	Like Mistral	Instruct + Base	""	Dec. 2023	""
Hestenettet-LM	""	""	Base	""	Nov. 2023	""
DanskGPT	13B	Like LLaMa 2	Chat	Closed	Aug. 2023	Danish
DanskGPT Tiny	1.1 B	""	Base	Open (Apache 2)	Jan. 2024	Danish
NB GPT-J	6B	GPT 3-like + RoPE	Base	Open (Apache 2.0)	Jan. 2023	Norwegian (Bokmål + Nynorsk)
NB GPT-J NorPaca	""	""	Chat	""	Apr. 2023	Norwegian (Bokmål)
GPT-Sw3 6.7B (v. 2)	6.7B	GPT 2-like	Base	Open (RAIL-based)	Apr. 2023	Swedish + Scandi. and English
GPT-Sw3 20B	20B	""	""	""	Dec. 2023	""
GPT-Sw3 40B	40B	""	""	""	""	""
GPT-Sw3 6.7B Instruct (v. 2)	6.7B	""	Instruct	""	Apr. 2023	""
GPT-Sw3 20B Instruct	20B	""	""	""	""	""

Table 3.5: Publicly available Danish and Scandi. models considered. Instruct + base refers to the fact that an English instruction model has been further trained on a non-instruct, Danish dataset. GPT-Sw3 models smaller than the 6.7 B model are not shown here. The custom GPT-Sw3 license is based on the Responsible AI License, allowing commercial use within specific restrictions [Sah23].

Omissions Two older, small Danish-specific GLLMs are not considered due to their limited size: The Flax Community Dansk GPT Wiki at 137M parameters [Fla23] and Kenneth Martinsens GPT 2 small Danish at 124M parameters [Mar23a] cannot be expected to hold Danoliteracy.

The open BLOOM and BLOOMZ models produced by the BigScience workshop are trained on 46 languages. However, they are omitted as English was the only Germanic language in the mix, making up 30 % of a pre-training dataset distribution that also included ~ 40 % non-Indo European languages [Wor+23; Mue+23]. The Facebook model XGLM is not considered for similar reasons: Here, English and German are the only Germanic languages included in the multilingual pretraining mix [Lin+22, Sec. 2.1].

Open GLLMs described as English-only such as the United Arab Emirates project Falcon [Alm+23] or the Microsoft Phi-2 model [Mic23] are omitted.

Finally, additional models behind closed APIs such as Anthropic’s Claude [Ant23] or models requiring registrations and approval such as Nvidia’s Nemotron [Zha+23] were considered out of scope.

3.2.4 Missing Models

Considering the presented models, there is a notable lack of a major, Danish-focused model: While Norway and Sweden contain examples of well-funded attempts to adapt the technology of GLLMs to their specific languages, existing work is in Denmark small-scale.

Additionally, I note that much of the work to create openly available competitors to ChatGPT, e.g. the efforts building on LLaMa 2, focuses mostly on English improvements and benchmarks, considering multilinguality secondarily. A large, open GLLM that focuses on being multilingual with capital M could also have immense global impact. Meta AI, Mistral, the LAION project, or Huggingface are all candidates to be drivers in this effort.

4. Data

4.1 Model Pre-training Data

To enable experiments with model trainings, a large dataset for training a Danish, base GLLM was compiled. A combination of two publicly available, general text corpora was used. These are presented below along with self-computed summary statistics.

- The Danish Gigaword dataset, compiling many curated text sources [Der+21], + Danish Reddit dataset combination released by the Danish Data Science Community organisation on 🤗 Huggingface Datasets Hub^{4.1}.
 - 2.6 M examples.
 - 6.5 B characters.
 - 1.1 B words.
- The Danish subset of the CulturaX web scrape combination corpus released by the University of Oregon combining, filtering and language detecting the web corpora MC4 and OSCAR [Ngu+23].
 - 25.4 M examples.
 - 92 B characters.
 - 14.7 B words.

The total number of words is 15.8 B, approximately 36 B tokens^{4.2} can be compared to GPT-3 being trained on 500 B tokens [Bro+20, Table 2.2]. Texts from these datasets were sampled with equal probability, effectively oversampling the former dataset. I also consider this dataset higher quality. The resulting combination can be seen in Figure 4.1. A random subset of 1000 examples were set aside for a validation set and additionally, 10,000 were excluded from training as a future training set.

^{4.1}https://huggingface.co/datasets/DDSC/dagw_reddit_filtered_v1.0.0

^{4.2}Calculated using a conversion factor of 2.3 LLaMa 2 tokens per Danish word, estimated on the 10,000 test split examples.

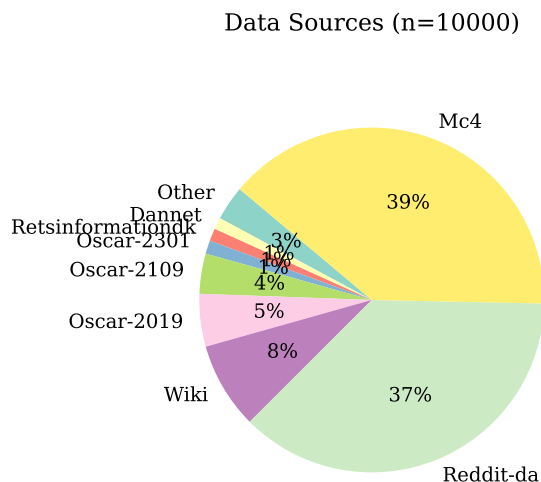


Figure 4.1: The pre-training dataset source example proportions estimated using the test split. Note, importantly, that this is proportion in the number of examples, not in text lengths.

4.2 Evaluation Data

For creating benchmark scenarios, several datasets were created. Below, the data sources for these are described. How these were turned into evaluation scenarios is explained in Section 5.3. Note that one text per dataset can be seen in Appendix Section A.1.1 in prompted form.

4.2.1 Citizenship Test

The Citizenship Test dataset was produced by scraping Danish governmental citizenship multiple-choice tests.

Data Source Twice a year, the Danish governmental agency SIRI (Danish: "Styrelsen for International Rekruttering og Integration", English: "The Agency for International Recruitment and Integration") administers tests. In addition to language tests, two types of exams are held: citizenship tests (Danish: "indfødsretsprøver") and

residency tests (Danish: "medborgerskabsprøver"). Passing the former is a requirement for Danish citizenship application and the latter for permanent residency in Denmark [23b]. They both seek to test "familiarity with Danish societal structures as well as Danish culture and history" and the residency test should additionally measure "the degree of integration"^{4.3}.

The format is the same for both tests: A question followed by either two or three possible answers; see the example in Table 4.1. The tests and corresponding answer keys are made public in press releases after each examination period.

The citizenship tests were introduced in 2007 [07] and are motivated by the argument that "The history of Denmark is important to know if one is to understand the Danish society as it appears today"^{4.4}. The tests have been a matter of much debate in the context of the general Danish immigration debate with critics arguing that the questions require too much memorisation [Hol21]. The November 2022 version had 45 questions: 35 about society, history, and culture based on teaching material given to examinees and 10 new questions, 5 relating to current news events and 5 about "Danish values about free speech, equality or the relationship between religion and law". To pass, examinees had to answer 36 out of 45 correctly and 4 out of the 5 "Danish values" questions. 53% of 2096 examinees passed the test [23a].

	Hvornår kom Martin Luthers protestantiske ideer til Danmark?
A	900-tallet
B	1200-tallet
C	1500-tallet

Table 4.1: A typical question in the tests, this one from the citizenship test in November 2019. Note that, in addition to culture and history, the questions are also about current societal principles, especially democratic and labour market issues, as well as testing knowledge of current events and Danish values.

Scraping and Cleaning Procedure The citizenship and residency tests were acquired through SIRI's website where tests and answer keys are available in PDF

^{4.3}Danish: "Indfødsretsprøven er en prøve om danske samfundsforhold samt dansk kultur og historie. (...) Medborgerskabsprøven skal vise ens integrationsgrad ved at teste ens fortrolighed med danske samfundsforhold samt dansk kultur og historie.", [23b, Section "Andre prøver"].

^{4.4}Danish: "Danmarks historie er vigtig at kende, hvis man vil forstå det danske samfund, som det ser ud i dag." [Nie16a]

formats in press releases^{4.5}. The PDFs were found to follow a generally standardised format since June 2019. All tests and answer keys were scraped from the site, resulting in 18 available tests from June 2019 to June 2023. The PDF content was parsed using PyPDF [Tho+23] and the relevant question/answer text was extracted and heuristically cleaned using regular expression patterns.

The resulting text had the problem of many words being broken by erroneous whitespace characters such as what appeared as "statsminister" in the PDF was extracted as "st atsminister". This is a known problem in PDF extraction due to the absolute placement of characters in the format [Tho23]. To overcome this, semi-automatic cleaning was performed. For each whitespace in a sentence, the alternative sentence without the whitespace was considered. The two sentences were then compared using the Alvenir 3-gram DSL language model [NJ19]. All word pairs that improved the likelihood of their sentence when joined were saved to a sheet which was manually reviewed after which verified errors were removed.

It is noted that deduplication was not carried out, resulting in some questions being identical or similar. This step can be added but the motivation for not performing it in the standard version was to not change the effective distribution of themes. By including all questions, the weight on different topics follows the importance assigned effectively to them by the Danish government.

The Dataset The Citizenship Test dataset contains 605 questions with 6,027 words in question strings and 3,974 words in the answer options. Permission was given by SIRI to release the dataset as an appendix to the released thesis. The dataset is released as an appendix with this disclaimer through the 🤖 Datasets Hub at sorenmulli/citizenship-test-da^{4.6}.

4.2.2 HyggeSwag

HyggeSwag was produced by manual translation of a subset of the HellaSwag commonsense natural language inference dataset [Zel+19] into Danish.

Data Source The dataset SWAG, *Situations With Adversarial Generations*, was released in 2018 by Zellers, Bisk, Schwartz, and Choi [Zel+18]. This dataset consists of

^{4.5}Found in <https://siri.dk/nyheder/?categorizations=9115>

^{4.6}huggingface.co/datasets/sorenmulli/citizenship-test-da

partial descriptions like the text "The woman is now blow drying the dog. The dog" and then four potential continuations of that text. One continuation was correct and the three others were produced using a method dubbed Adversarial Filtering, where language models were used to sample potential continuations that should be stylistically correct but semantically wrong [Zel+18]. The main source of the texts was the ActivityNet Captions [Cab+15; Kri+17] which were built by human annotators describing ongoing actions in short YouTube video clips [Kri+17, Sec. 4].

SWAG was soon after publication found to be mostly solved when BERT [Dev+19] was released, yielding an accuracy of 86%, close to human performance, 88% [Zel+19, Sec. 1]. To remedy this, Zellers, Holtzman, et al. [Zel+19] released HellaSwag, improving upon SWAG by performing stronger Adversarial Filtering which produced longer and more complex generations that successfully fooled BERT. Human verification was performed to ensure that the generated options were not semantically correct according to commonsense judgement [Zel+19, App. F]. In the end, human accuracy of 94% was attained on the task [Zel+19, Sec. 4.3]. Newly scraped WikiHow data was added along with the existing ActivityNet Captions [Zel+19, Sec. 4].

HellaSwag contains 60,000 examples and is used as one of four scenarios in the Open LLM Leaderboard for evaluating English LLM's [Bee+23].

Subset to Translate HellaSwag was available through the 🗃️ Datasets Hub at huggingface.co/datasets/hellaswag. The validation split consisting of 10,000 examples was used as the test split did not contain true labels. Only the ActivityNet examples were used. This decision was made based on the notion that the WikiHow how-to guide examples carry more cultural context which may be lost in translation than the simple activity descriptions do.

Some of the captions describe different activities during the same video. To maximise the resulting scenario diversity at a given translation workload, no more than one caption per video was translated. The examples were ordered randomly before translation.

Translation Procedure For each example, the context and the four completion candidates were manually translated into Danish.

As ActivityNet has been crowdsourced, there might be grammatical errors and typos both in the context and the human completion. Obvious errors were not transferred into the translation but imprecise grammar and inelegant sentences were kept

as close to the English version as possible. A further challenge in translation was incoherence in the Adversarial Filtered examples: Some of the wrong completions were wrong due to nonsensical word use which needed to be transferred to the Danish version to avoid turning a wrong completion into a correct option. An example of a challenging example is shown in Table 4.2

	Original English	Danish Translation
Context	A teen boy talks in a public place while remember his martial arts competition. The teen boy	En teenagedreng taler på et offentligt sted, mens han husker sin kampsportskonkurrence. Teenagedrengen
Option 1	then plays in a large trash can while children are walking past, and the teen describes his experience.	leger derefter i en stor skraldespand, mens børn går forbi, og teenagedrengen beskriver sin oplevelse.
Option 2	fetches a kickball and throws it at a goal.	henter en kickball og kaster den mod et mål.
Option 3	fails once and kicks the male's leg.	fejler én gang og sparker mandens ben.
Option 4	competes karate with other boy while a person film him.	dyster i karate med anden dreng, mens en person filmer ham.

Table 4.2: A difficult example to translate from HellaSwag: The context contains the error "while remember" which is fixed in the translation. Option 2 contains the word "kickball" lacking Danish equivalent and Option 4 contains grammatical errors which are partly preserved. Can you figure out the correct commonsense completion^{4.8}?

The translation was performed without exposing the translator to which option was correct. To validate that the translation step did not undermine test validity, human prediction was performed on all examples. If an example was misclassified, the options were attempted retranslated and a new person attempted classification. If this classification also failed, the example was discarded.

14 out of 133 examples were misclassified the first time corresponding to an initial 89% human accuracy. 9 of these were misclassified by a second person after retranslation and thus discarded.

^{4.8}Spoiler warning: The correct option is number .noj

The Resulting Dataset The resulting dataset, HyggeSwag, contains 124 examples with 2,047 words in input contexts and 5,531 words in options. The ActivityNet Captions contain a field with an activity label. To get an overview of HyggeSwag semantic domain, the activities for the translated examples are matched from the original captions and described in Table 4.3.

Activity	Count
Cutting the grass	8
Clean and jerk	8
Washing face	7
Playing violin	7
Having an ice cream	7
Total unique activities	64

Table 4.3: The top 5 activity labels in HyggeSwag and the total number of unique activity labels. Most textual content appears thus to be about mundane, everyday subjects.

The dataset was released through the 🗄️ Datasets Hub at [sorenmulli/hyggewag](https://huggingface.co/datasets/sorenmulli/hyggewag)^{4.9} under the permissive MIT license, following HellaSwag.

4.2.3 #twitterhjerne

#twitterhjerne is a small dataset of tweets containing questions or asking for input along with a number of replies for each question.

Data Source The hashtag "twitterhjerne" (En.: Twitter Brain) has received popularity among the Danish users of X, formerly Twitter. It is used to ask for help from the collective hive mind, be it recommendations, technical questions, everyday challenges or dilemmas [Har19].

Collection Procedure A search of tweets containing #twitterhjerne was performed and each tweet was manually examined and added to the dataset if it fulfilled the criteria:

- Had a clear question in the tweet.

^{4.9}huggingface.co/datasets/sorenmulli/hyggewag

- Did not require an attached photo to be intelligible.
- Contained no personal information.
- Had at least two relevant replies. Relevancy meant attempting to answer the question.

The tweet and 2-6 relevant replies were saved without including usernames or tweet metadata.

Tweets from August 2023 to December 2023 were included.

The Resulting Dataset The dataset contained 78 examples with 2,268 words in question tweets and 6,223 in answer tweets. An example had an average of 3.7 replies.

Following other social media datasets such as Angry Tweets, this anonymised tweet dataset was released on 🗄️ Datasets Hub at [sorenmulli/da-hashtag-twitterhjerne](https://sorenmulli.com/datasets/da-hashtag-twitterhjerne)^{4.10}.

4.2.4 Reading Tests: Cloze Self Test and Gym 2000

Two small reading test datasets were scraped from The Centre for Reading Research (CRR) and saved as multiple-choice scenarios.

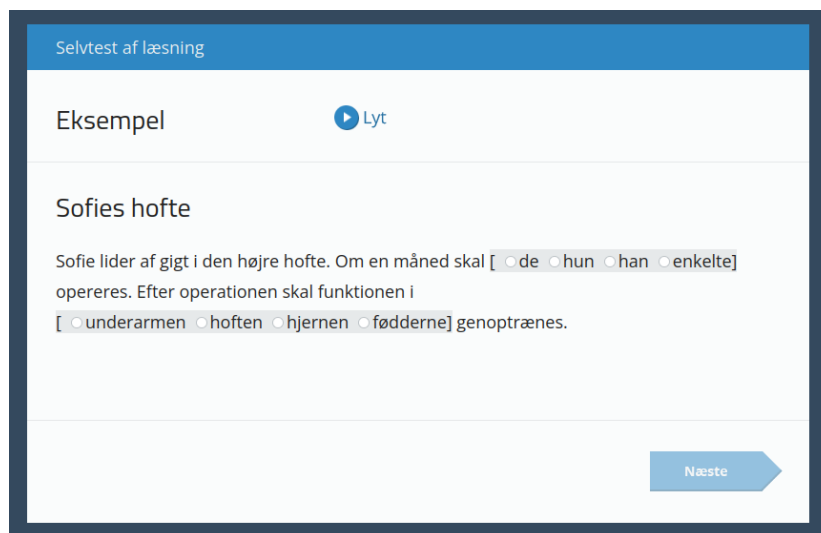
Data Source The CRR under the Department of Nordic Studies and Linguistics at the University of Copenhagen produced test materials that are publicly available at laes.hum.ku.dk/test. Two tests were considered relevant:

1. The Self Test of Reading Competences for Adults (Da.: "Selvtest af læsefærdigheder for voksne") [JGE15]: A cloze-style test, presenting a text with some words hidden, showing four possible options. I will refer to this dataset as Cloze Self Test. The test is available as a Javascript application on selvtest.nu. An example question can be seen in Figure 4.2
2. Reading Texts for Gymnasium (Da.: "Læsetekster for gymnasium, hf mv") [EA01]: Three texts, each with 6-16 multiple choice comprehension and analysis questions produced in 2000. I will refer to this dataset as Gym 2000. The texts, questions, and answers are available as static HTML sites^{4.11}. The texts are

^{4.10}huggingface.co/datasets/sorenmulli/da-hashtag-twitterhjerne

^{4.11}static.uvm.dk/Publikationer/2000/laesetekster

- Sommerset Maughan: "Hr. Ved-alt", Cosmopolitans (1938). A short story.
- Tom Kristensen: "Det blomstrende Slagsmaal", Fribytterdrømme (1920). A poem.
- An excerpt from "Grundbog i biologi". Introductory text about blood circulation.



Selvtest af læsning

Eksempel [Lyt](#)

Sofies hofte

Sofie lider af gigt i den højre hofte. Om en måned skal [de hun han enkelte] opereres. Efter operationen skal funktionen i [underarmen hoften hjernen fødderne] genoptrænes.

Næste

Figure 4.2: An example cloze test from selvtest.nu. One text can have multiple missing words that the reader must fill out, like this one. For the scenario, this text would be two examples, each with the other cloze option replaced with the true option.

Scraping Procedure

Texts and questions were scraped from their sites.

For the Cloze Self Test, scraping using manual intervention to get Javascript content was implemented. All cloze texts with options were saved. Answers were manually guessed and manually verified by inputting into the application. If a text had multiple cloze words, an example was created for each word by replacing the other cloze words with their correct option. There were 24 unique texts, yielding 50 total cloze examples.

For Gym 2000, texts, questions and answers were all available. The biology text was not used, as to test the model capabilities on raw science knowledge without info. Questions that referred to a figure were removed. For the two fiction literary texts, textual context is required to answer the questions. Instead of saving the entire

text as context for each example, a subset was manually chosen. For the poem text, questions referred to specific verses which were easy to cut out. For the short story, subjective judgement was used to identify the part of the text which was needed to answer the question.

The Resulting Datasets Cloze Self Test contained 2,644 words of text divided into 50 examples each with four word options. The Gym 2000 scenario contained 33 examples, totalling 5,506 words across contexts, questions and answer options. Permission was given by Carsten Elbro, Professor at CRR, to use the test materials but not to re-release them in other forms. The datasets are thus not openly available in tabular form but can be reconstructed one-to-one using specific commands in the `danoliterate` framework.

4.2.5 Existing NLP Tasks: DaNE, AngryTweets and Nordjylland News

Three existing NLP datasets were used.

DaNE contains NER-annotated sentences from DDT into the categories: person, organisation, location and miscellaneous [Hvi+20]. A smaller subset of 256 examples was used^{4.12}, created as part of the ScandEval benchmark [Nie23a]^{4.13}.

The sentiment analysis dataset Angry Tweets [Bro+21, Sec. 4] was also added using an existing 256 example subset^{4.14}. The dataset contains tweets, each annotated with the label positive, negative or neutral.

Finally, the summarisation dataset Nordjylland News, released by the Alexandra Institute was used [Kin23]. This dataset contains newspaper texts from TV2 Nord with corresponding summaries. A random subset of 300 texts, all with less than 1000 characters in the text was kept. This Nordjylland News subset was released on the 🗄️ Datasets Hub at `sorenmulli/nordjylland-news-summarization-subset`^{4.15}.

^{4.12}The 256 example subsets of DaNE and Angry Tweets were used to make the dataset scale more manageable. However, when making this decision, I missed the fact that these splits are actually from the validation set and not the final test set. For GLLM evaluation, this is not problematic as the validation set was not used for fine-tuning modelling decisions, however, for future comparability and best practices, the larger canonical test split should be used.

^{4.13}huggingface.co/datasets/ScandEval/dane-mini

^{4.14}huggingface.co/datasets/ScandEval/angry-tweets-mini

^{4.15}huggingface.co/datasets/sorenmulli/nordjylland-news-summarization-subset

5. Methods

5.1 Baseline Model Training

In addition to evaluating GLLMs, training experiments of Danish base GLLMs were carried out. These are described in this section.

5.1.1 Model Architecture and Transfer Learning

For trainings, both the LLaMa 2 7B architecture [Tou+23a] and the Mistral 7B architecture [Jia+23] were used. These are GPT-3 based decoder transformers with SwiGLU activations and RoPE where Mistral additionally implements GQA and SWA. Key architectural parameters for the 7B versions are shown in Table 5.1. Existing base model checkpoints for these two models were attained to be used for transfer learning, initialising training from the multilingual existing weights. Their existing tokenisers were used.

	LLaMa 2 7B	Mistral 7B
Transformer blocks	32	32
Hidden size	4096	4096
Attention Heads	32	32
Feedforward size	11008	14336
Number of grouped heads	32	8
Tokeniser vocab.	32K	32K
Theoretical max. context length	4096	128K
Reported context length	4096	8192
Total Parameters	6.74B	7.24B

Table 5.1: Architectural details for the two model types trained. The context lengths that the checkpoints were reported to be trained with are also included. Feedforward size is also called intermediate size and describes the shapes of the linear layers after transformer blocks. The number of grouped heads differs from using number of attention heads when using GQA.

5.1.2 Training Approach

Optimisation The base GLLM was trained with a classical language modelling objective, predicting the next token. Optimisation was implemented using the ADAM optimiser [KB14], linearly warming the learning rate up to a peak of 10^{-6} during the first 10,000 optimisation steps. Gradient accumulation was used, allowing for larger step sizes (effective batch sizes) than the maximum backpropagation that could fit into memory.

An effective batch size of 32 sequences was used.

Data Processing Training on the two, huge datasets was implemented using the 🤖 Datasets Streaming IterableDataset [Wol+19], loading and shuffling data examples live from the internet while training. Examples were thus sampled randomly using no concepts of epochs. Models were trained on strings with context lengths of 1024 tokens. When examples were shorter than this context length, multiple texts were concatenated, separated by end-of-sequence tokens. Notably, it was implemented such that, for examples longer than 1024 tokens, a substring was randomly picked instead of training on all substrings from each example. This one-substring-per-example policy means that text that occurs in short examples is more likely to be sampled than text that occurs in long examples. This was done for three reasons:

- The bias towards text in short examples may be beneficial for this limited-time training. The higher-quality Reddit comments are thus sampled more than long legal or web documents from Gigaword or CulturaX.
- A subtle problem can otherwise occur using when streaming from an IterableDataset which can give short-term overfitting. While the dataset shuffles the substrings extracted from streamed examples in a buffer, very long texts can fill up this buffer meaning that multiple batches in a row will contain text from the same example.
- For huge texts, only a subset of them needs to be tokenised at a time, removing a training problem where rare, enormous texts make the training wait a long time for tokenisation.

Validation on the 1000 validation examples was performed every 100 updates.

Compute Initially, training using low-rank adaption (LoRA) [Hu+21] was implemented. However, stability of pilot training was low and full parameter training

was prioritised to produce a training more indicative of realistic GLLM pre-training. When access was gotten to nodes with H100 GPUs through the Danish Pioneer Centre for AI, model-parallel, full-parameter training was used instead. Parallelism was implemented as multi-GPU, not multi-node, meaning that the hardware allowed for training on maximum two GPUs per model.

As 7B parameter models were trained, this was a limited computational environment, requiring maximal efficiency. Thus, the DeepSpeed method ZeRO stage 2 was implemented, enabling model parallelism between both GPUs and offloading additional gradients and optimiser states to the CPU [Ras+20]. This allowed completion of around 4,000 optimisation steps, each 32 sequences (4 forward passes on each device \times gradient accumulation of 4), totalling ~ 130 M tokens seen a day. The training loop was built using the Supervised Fine Tuning Trainer from the `trl` library version 0.7.4 [Wer+20] and DeepSpeed was configured using the `accelerate` library version 0.23.0 [Gug+22].

The models were trained in half-precision using the brain float 16 floating point format.

Trainings Two major models were trained. One using the Llama 2 7B architecture, initialised from the Llama 2 7B checkpoint, and another using the Mistral 7B architecture, initialised from the Mistral 7B checkpoint. As an ablation on transfer learning, a model with LLaMa 2 7B architecture was trained from scratch without initialising weights from existing checkpoints. I optimistically dub these three models:

1. Danoliterate LLaMa 2 7B
2. Danoliterate Mistral 2 7B
3. Danoliterate Baseline LLM

Frustratingly, both in pilot experiments and for the main experiment, unexplainable behaviour in loss development was seen when using the DeepSpeed ZeRO method on the Mistral architecture. The loss curves exhibited sudden jumps both at random times and also when resuming training from a checkpoint saved to disk. A pilot experiment showed this problem more strongly at a higher learning rate, shown in Section A.2.1 though no explanation was found. For the Danoliterate Mistral 2 7B model, the last model before this unstable loss behaviour was saved and used as the final model though the training was continued to investigate the diverging trajectory.

This scale of instability was not seen for the LLaMa 2 architecture.

5.2 Evaluation Models: Who Were Invited?

Now, turning to the produced benchmark, this section starts by enumerating the models implemented for evaluation. In addition to real models, a constant dummy baseline was run on all scenarios which outputs a probability of 0 and the text "a b c d [...]" to any input.

5.2.1 Openly Available Models

Implementations and pre-trained weights of Open Source Transformer GLLMs were used through the 🤖 Transformers library [Wol+19]. Version 4.36.1 was used to generate the results using the `AutoModelForCausalLM` API. An Nvidia H100 GPU was used for inference.

For generation, random sampling was disabled, that is, the temperature was set to zero, meaning greedy LM decoding. For generated text, generated likelihoods were extracted. Furthermore, for all these models, conditional language model likelihood was implemented using the PyTorch Cross Entropy Loss [Pas+19]. This inference approach will be detailed as a metric in Section 5.4.1.

Batched GPU inference was implemented and all models were run on a Nvidia H100 starting with a batch size of 32 examples which was iteratively halved at runtime if the model ran out of memory on a scenario. Custom implementation was necessary to batch both generation and likelihood computations across different models. The implementation handled maximum sequence lengths, truncating too long inputs, and computed model likelihoods for generated tokens. Furthermore, for tokenisation, if the model was an instruct-, or chat-tuned model, a model-specific instruction format was used if it was defined in the available model tokeniser using the tokeniser `apply_chat_template` API [Hug24].

All weights were acquired through the 🤖 Huggingface Model Hub^{5.1} Some open models considered as candidates for Danoliteracy in Section 3.2.3 were not evaluated. These were either too computationally intensive to be run within the time scope for the project or were not implemented in time for evaluation. These include the Mixtral model, the 70B LLaMa models and the non-instruct SOLAR model. The evaluated open models can be seen in the main results Table 6.2.

^{5.1}See https://huggingface.co/models?pipeline_tag=text-generation for a catalogue of > 40K GLLMs.

5.2.2 Closed Models: Black Box Text Generators

OpenAI API OpenAI models were benchmarked by using the OpenAI API at platform.openai.com. Only text generation based on a given prompt is possible, so no likelihoods were acquired. The OpenAI Python client version 0.28.1 [Ope23c] was used to query the platform with a temperature of 0. The API was called in December 2023. The models shown in Table 5.2 were benchmarked.

Name	Model API key	Model Type
GPT 4 Turbo	gpt-4-1106-preview	Chat
GPT 4	gpt-4	Chat
GPT 3.5 Turbo	gpt-3.5-turbo	Chat
GPT 3.5 Turbo Instruct	gpt-3.5-turbo-instruct	Text Completion
Davinci 002	davinci-002	Text Completion
Babbage 002	babbage-002	Text Completion

Table 5.2: The OpenAI models that were benchmarked. The newest version of all OpenAI models can be seen in the API documentation at platform.openai.com/docs/models.

The use of the OpenAI API was funded by the Pioneer Centre for AI.

Google Vertex AI API Benchmarking of Google Models was enabled using the Google Vertex AI API [Clo]. The Google Cloud AI Platform was used to query with a temperature of 0 in December 2023. The safety settings, blocking possible harmful text such as hate speech or sexually explicit content were disabled. Of the Gemini models [Dee23], only Gemini Pro was released at this time and was benchmarked.

Use of this API with < 60 requests per minute was free through Google Cloud Platform.

DanskGPT The unreleased model DanskGPT [Hen23a] was not added in time for the project though communication about possible evaluation methods such as an API setup has been commenced with Mads Henrichsen.

5.3 Scenarios: What Did I Use?

This section initially describes methodological considerations of scenario inclusion criteria and scenario prompting, after which each scenario is introduced. For each

scenario, the data source, the acquisition method and resulting statistics are detailed along after which scenario inclusion is justified.

5.3.1 Criteria

The produced benchmark is a scenario compilation. Each scenario was constructed using some data source. In this process, I prioritised time using the following principles for scenario construction.

Validity The most important criterion was also the most difficult to judge: The scenario should be a valid measure of Danoliteracy. This principle meant exclusion of perplexity calculation of large corpora and other, automated or model-based criteria that would require thorough experimental validation deemed out of project scope. Furthermore, automatic translations of English benchmarks were not included.

Remedy missing scenarios New scenarios should expand the NLP scenario coverage. As all scenarios are in Danish, each one naturally improves language coverage, but within Danish, choosing diverse tasks and domains was a goal. As part of this, data sources with Danish cultural or factual contents were considered important.

Enabling holistic evaluation Ideally, a new scenario allows for multiple metric dimensions, improving the holistic evaluation of the benchmark.

Transparent data To make the benchmark reproducible and allow for future, open-source collaboration, the evaluation data should ideally be openly available. As an alternative, the data acquisition should be reproducible.

Breadth, not depth Finally, I mention a more practical production principle.

In the thesis, it was prioritised to add more scenarios rather than to expand the number of evaluation examples for each. Apart from time constraints, this choice was made based on an expectation that this benchmark would be a difficult one where many models would perform far from top level. Furthermore, I expected considerable model performance differences between small, open-source decoders to the OpenAI behemoths. Under these two assumptions, only a limited amount of examples is necessary for meaningful results as is exemplified in Figure 5.1.

Extension in depth is a natural future improvement of the benchmark.

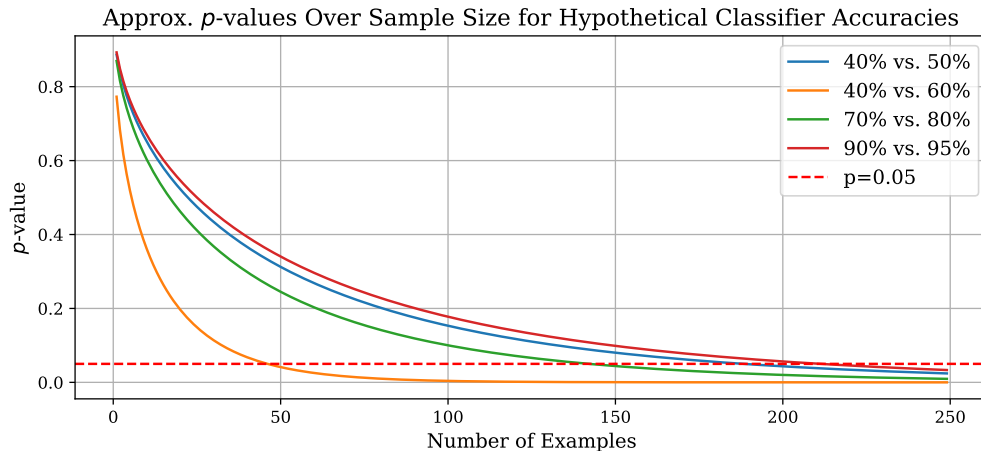


Figure 5.1: The p value of the null hypothesis that two binary classifiers have the same mean accuracy as a function of the number of examples. The different curves show examples of different hypothetical true model accuracies, suggesting that an unsaturated benchmark with large performance differences can be meaningful with < 100 examples. The calculation is based on the normal approximation of the binomial for comparing categorical populations [Bro+18, Method 7.15].

5.3.2 Prompting

Apart from the dataset and the downstream comparison metric, each scenario is also defined by its prompting: How is the data source operationalised to produce inputs for GLLMs?

In the benchmark, prompts were written for each scenario. How they were formulated for each is covered in the next section but some general ideas were used: The constructed prompts were structured with each kind of text such as question context, question text, question options, few-shot examples and answers as individual sections. Each section had a short header in upper case letters indicated started by the # character as in the Markdown format. This choice was motivated by the general prompting best practice of separating instructions and text content [Fac23] along with an assumption that Markdown is a ubiquitous format often found in the large internet corpora on which GLLMs are trained.

It was a priority to design the prompts to work both for instruct-tuned and base models. To achieve this, each prompt started with an instruction and ended with text leading the model towards an answer by giving the first few words. A mock example of this general approach is shown in English in Figure 5.2.

```
1 Write a one-sentence summary of the text.  
2 # TEXT  
3 Lorem ipsum dolor sit amet, consectetur adipiscing elit, ...  
4 # SUMMARY  
5 A summary of the text could be:
```

Figure 5.2: A general example of the prompting approach used in the Danoliterate benchmark. The prompt could also include few-shot examples, multiple options, or stricter requirements of the model outputs.

The specific prompting formulation decisions were intentionally not optimised for accuracy: The prompts were designed to showcase an initial performance that a user can expect from the model before delving into prompt engineering techniques.

Some scenario prompting formulations were subject to early testing on the ChatGPT UI and a LLaMa 7B Chat interface. I believe the few changes made based on such testing were broad and should not bias the prompts toward the preferences of these models. However, performing any prompt choice is a source of benchmark bias and experiments with variations in prompting were carried out and are documented in Section 7.4.3.

5.3.3 Resulting Scenarios

Chosen scenarios are here enumerated along with justifications for their inclusion. One data example including the chosen prompt can be seen for each scenario in Appendix A.1.1. More details about the datasets used for the scenarios can be found in Section 4.2

1. Citizenship Test The 605 Citizenship Test questions were prompted to models as multiple choice examples, displaying the two or three answer options.

The scenario was included with great confidence: Benchmarking validity can be argued based on the tests being produced by a governmental agency and their content politically considered meaningful. Furthermore, the scenario remedied a

missing focus on Danish knowledge, has a satisfactory size and can be made publicly available.

2. HyggeSwag The HyggeSwag scenario was prompted by asking models about which text continuation was the correct one on the 124 commonsense inference examples. The four possible continuations were presented as options.

The scenario was included to improve coverage of mundane, commonsense knowledge. The size was limited but can be expanded as future work. Scenario validity is dependent on the original ActivityNet captions, the HellaSwag Adversarial Filtering approach and the added translation step. The latter must maintain the commonsense correctness of the right option and the incorrectness of the others.

3. #twitterhjerne The #twitterhjerne scenario was created by displaying the question tweet, and requiring the model to write a helpful reply. Then, evaluation can be performed by comparing the model reply with the human replies to that tweet, taking the human versions as ground truth references.

The dataset was included to allow for a natural prompting setup, requiring free generative ability. The size of the dataset was limited but can be expanded. The validity of results depends on the downstream comparison method between GLLM and human reply. Finally, free generation in response to social media posts allows for improving holistic evaluation, such as checking for toxicity.

4. Cloze Self Test The Cloze Test was prompted to the models by giving the full text with the cloze word being masked with <MASK>, requiring the model to answer only with the masked word. The word options were presented to the model.

The size of this and the Gym 200 scenarios were very limited but were included to remedy a lack of scenario testing in a classical school format. The Cloze Self test is considered a scenario benchmarking zero-shot performance on a vocabulary task, requiring accurate NLU to generate correct completion. The validity is high as the test is produced by a linguistic research organisation.

5. Gym 2000 The Gym 2000 benchmark was operationalised by prepending the contextual context to the text understanding question with the textual context.

This scenario tests more semantic NLU than the cloze test, requiring models to understand deeper textual meanings and generate the right option. The Danoliter-

acy validity of this scenario is high but is influenced by whether the correct textual context was prepended to the prompt.

6. Nordjylland News Nordjylland News was prompted by requiring the model to write a one-sentence summary of a given short news article.

This scenario was included as a purely NLG task following a natural and canonical NLP setup of news summarisation. Both this and the following two scenarios use already openly available datasets where size can be improved considerably.

7. DaNE The DaNE dataset was implemented as 3-shot prompting using the GPT NER method [Wan+23], prompting the models once for each of the four NER categories: person, organisation, location and miscellaneous. Few-shot sampling was implemented as seeded RNG sampling.

Furthermore, a specific sampling approach was used. For each prompt, the model was asked to annotate any entities of a specific type, e.g. person. Constructing this prompt, one of the few-shot examples was not chosen completely at random but was sampled such that at least one person entity was present in the text, avoiding situations where all three examples are irrelevant and unhelpful for the model.

DaNE was considered relevant as the maybe most-used Danish NLP scenario with the strength that NER measures factual, cultural knowledge of Danish named entities at the same time as measuring structural understanding of language.

8. Angry Tweets Angry Tweets followed standard multiple-choice prompting, introducing the three sentiment categories for classification

The scenario was included to expand multiple choice tasks beyond knowledge-focused tasks and into more subtle emotional understanding.

5.4 Metrics: How Did I Measure?

This section introduces how the models were evaluated on the given scenarios, first introducing the implementation of LM and NLG, then describing metric implementations and finally leaderboard details

5.4.1 Inference Modes

All scenarios were of the single prompt form as shown in Figure 5.3: Each example consisted of a textual input that prompted an answer of a specific kind.

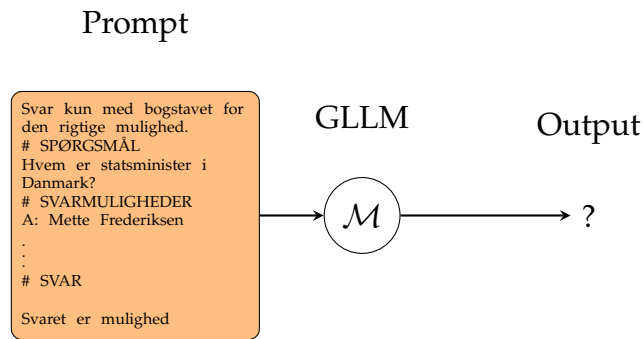


Figure 5.3: The benchmarking single example inference setup.

Two possible inference outputs were implemented given this input: Text continuation and LM.

Text Continuation All benchmarked models perform conditional NLG: Writing text to continue the given prompt as shown in Figure 5.4.

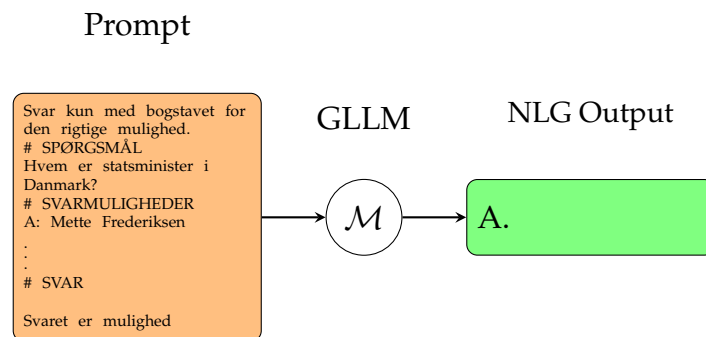


Figure 5.4: The NLG inference mode exemplified.

For all open models, this was implemented as greedy decoding, selecting the token with the highest LM probability as the next token. The OpenAI and Google models are also generally expected to behave this way, though it cannot be guaranteed that these systems do not perform other tricks behind the scenes.

Generation runs until the model either predicts an end-of-sequence token or a maximal sequence length is reached. For all models, a maximum of 256 tokens was used.

Language Modeling For models that supported LM, the conditional model likelihood of a continuation given a prompt can be computed as shown in Figure 5.5. The likelihood scoring was implemented as explained below.

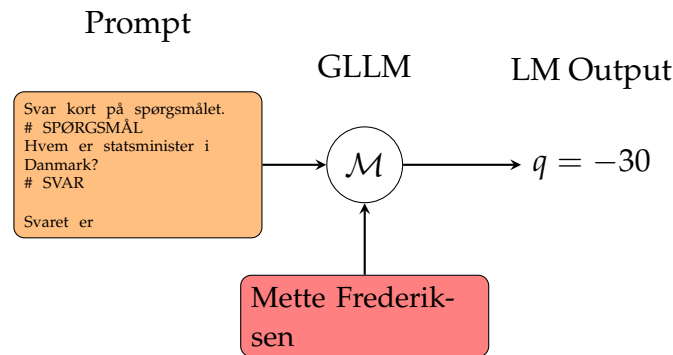


Figure 5.5: The LM inference mode exemplified: The model assigns the generation “Mette Frederiksen” a likelihood of 10^{-30} of being the continuation of the prompt.

Given a language model \mathcal{M} , and a prompt sequence of tokens IDs,

$$\mathbf{s}_{\text{prompt}} = [t_1, t_2, \dots, t_N], \quad t_i \in \mathbb{N}, 1 \leq t_i \leq K, \quad (5.1)$$

a possible continuation sequence

$$\mathbf{s}_{\text{cont}} = [t_{N+1}, t_{N+2}, \dots, t_{N+M}], \quad (5.2)$$

can be given a score

$$0 \leq q(\mathbf{s}_{\text{cont}} | \mathcal{M}, \mathbf{s}_{\text{prompt}}) \leq 1 \quad (5.3)$$

that measures how likely the language model is to complete the prompt text with the completion text.

For GLLMs, a probability distribution over candidate tokens for the next, single token is parametrized by the last layer activation output logits \mathbf{h}

$$\mathcal{M}(\mathbf{s}) : p_{\mathcal{M}}(t_{N+1} | [t_0, \dots, t_N]) = \sigma(\mathbf{h}_{\mathbf{s}; N+1}), \quad (5.4)$$

where σ is the SoftMax function shown on eq. 2.4 ($T = 1$).

The natural choice of q is to measure the continuation probability of the entire sequence under the prompt

$$p(\mathbf{s}_{\text{cont}} | \mathbf{s}_{\text{prompt}}) = \prod_{i=1}^M p(t_{N+i} | \mathbf{s}_{\text{prompt}}, t_{N+1}, \dots, t_{N+i-1}), \quad (5.5)$$

which is possible due to the chain rule of probability. The used score in my implementation was the logarithm of this probability for numerical stability:

$$q(\mathbf{s}_{\text{cont}} | \mathcal{M}, \mathbf{s}_{\text{prompt}}) = \log p(\mathbf{s}_{\text{cont}} | \mathbf{s}_{\text{prompt}}) = \sum_{i=1}^M \log p(t_{N+i} | \mathbf{s}_{\text{prompt}}, t_{N+1}, \dots, t_{N+i-1}), \quad (5.6)$$

This score q is called the continuation log-likelihood.

The score was implemented using the efficient implementation in the PyTorch Cross Entropy Loss computation. This loss function computes

$$L(\mathbf{s}, \mathbf{H}_{\mathbf{s}}, I)_i = -\log \sigma(\mathbf{h}_{\mathbf{s}; i})_{t_i} \mathbb{1}(t_i \in I) \quad (5.7)$$

where $\mathbf{H}_{\mathbf{s}}$ is a matrix of model LM decoder head activations with $|\mathbf{s}|$ rows, and $\mathbb{1}(t_i \in I)$ returns 0 if t_i is in the set I of token IDs to ignore, else 1.

To compute q using this implementation, it is noted that given a sequence, the model is trained to predict the next token for each token. Thus, for $\mathbf{H}_{\mathbf{s}}$, position i contains the activations corresponding to the probability distribution of the $i + 1$ 'th token. The logits are then shifted relative to the sequence before calling the loss function. Furthermore, the prompt tokens are replaced by a value in the ignore set,

resulting in the continuation log likelihood:

$$q(\mathbf{s}_{\text{cont}} | \mathcal{M}, \mathbf{s}_{\text{prompt}}) = L([\tilde{t}_2, \tilde{t}_3, \dots, t_{N+1}, t_{N+2}, \dots, t_{N+M}], \mathbf{H}_{\mathbf{s}, 1::N+M-1}, \{-100\}). \quad (5.8)$$

where $\tilde{t}_i = -100$.

This implemented way to compute q was implemented in an efficient, batched manner and was empirically unit tested against the implementation in the EleutherAI Evaluation Harness [Gao+21].

5.4.2 Similarity Algorithms

For NLG-based metrics, the computation of a similarity score between two texts is useful. Three comparison algorithms were implemented.

Lemma similarity: ROUGE Recall-Oriented Understudy for Gisting Evaluation, ROUGE, [Lin04] comparing overlaps of mentioned words was implemented for ROUGE-1, unigram overlap F1-score, and ROUGE-L, F1-score computed for the length of the longest common subsequence. Both these were implemented on lemmatised texts to avoid measuring small variations in conjugation. Lemmatisation was performed using the rule-based SpaCy Danish lemmatiser [Hon+20].

Semantic similarity: BERT score A semantic similarity measure was implemented using the technique BERT score which uses pairwise cosine similarities between the contextual word embeddings of two sequences produced by a pre-trained BERT model [Zha+20]. The DFM Encoder Large v.1 was used to produce embeddings [Mod22].

5.4.3 Metrics of Capability

The following metrics were implemented to measure the capability of GLLMs to fulfil the task, responding correctly to the prompt. Generally, all except one of these can, using the nomenclature of Section 2.3.1 be described as examples of Idea 6: algorithmic comparison to reference data.

Multiple Choice Completions The majority of scenarios are constructed as a variation of the case where completion is performed on an example with multiple possible generations, one being correct and the other ones wrong. Three different metrics were implemented for such scenarios: Firstly, if the scenario presented the options and required the model to generate the numeral of the correct option, then the model output text is parsed for this completion. This is called NLG parsing.

If the options were not simply presented, the chosen option can be decided using either NLG or LM. From NLG, the model can continue the generation and the chosen option can be taken as the option with maximum similarity to the generation, using either lemma or semantic similarity. For LM, the option with the highest likelihood is taken as the model guess.

Knowing the chosen and correct option, full scenario accuracy calculations can be carried out.

Summarisation The produced summary was compared to the reference summary using all three similarity algorithms taking higher similarity as a generated summary closer to the desired meaning.

Multiple Reference Generations For the #twitterhjerne scenario holding multiple gold-standard reference continuations, two kinds of metrics were implemented both based on similarity scores. Firstly, odd-one-out frequency measured the pairwise similarity between all possible continuations: Both the model generation and the reference answers. The continuation with the lowest similarity is taken as the odd one out. It was then counted how often the model generation had this misfortune, taking this frequency to be the lower, the better. When using BERT similarity, this metric can be categorised as an instance of Idea 4, model-based discrimination.

Alternatively, the model generation similarities to all reference answers were computed and a norm was taken on these: Both average similarity to references as well as maximum and minimum similarity were implemented.

NER The model answers to the GPT-NER prompting were parsed into a word-level prediction. This parsing had some difficulty as some models failed to follow the prompt completely, injecting text before or after their guess, or made small errors when reproducing the words. To avoid being too strict, the output word-level predictions were aligned with the word list using the Levenshtein edit distance

algorithm [Lev66]. Words that the alignment algorithm gives the operation code DELETE are ignored for scoring and words aligned with INSERT are given the out-of-categorisation label "O".

Given parsed word level predictions, the micro average F1 score was computed using the canonical framework SeqEval version 1.2.2 over all entity classes [Nak20].

5.4.4 Further Metric Dimensions

All previously mentioned metrics measured model task performance. To enable holistic evaluation, metrics for four other dimensions were implemented.

Note that metrics for the HELM dimension bias have not been implemented. This dimension was deemed too challenging to attempt in the scope of this project. Furthermore, it must be mentioned that the dimensions toxicity, robustness and fairness are complex subjects that require further, targeted analysis and the metrics presented here can be considered experimental attempts.

Efficiency The inference time of each scenario is saved along with device information. As batch inference is automatic based on runtime GPU out-of-memory, the approach takes maximal possible batch sizes for a given model size into account. Efficiency is only recorded for open models.

Calibration For LM inference on tasks with multiple options, a probability distribution can be produced across L options by normalising across option likelihoods. This calculation was implemented and these distributions were checked for calibration. Two calibration measures were implemented: The Brier score and the expected calibration error (ECE). The former is the mean squared error on probabilities between the correct Dirac delta distribution and predicted probabilities, thus

$$\text{Brier}(\mathbf{P}, \mathbf{t}) = \frac{1}{M} \sum_{i=1}^M \sum_{j=1}^L (p_{ij} - \mathbb{1}(t_i = j))^2, \quad (5.9)$$

where \mathbf{P} contains M predicted probability distributions, and \mathbf{t} the correct index for each. The ECE was implemented over 10 equally spaced bins B_k , being the weighted mean of the difference between realised accuracy and average confidence for each of

the equally-spaced bins, that is

$$\text{ECE}(\mathbf{P}, \mathbf{t}) = \sum_{k=1}^{10} \frac{|B_k|}{M} \left| \underbrace{\frac{1}{|B_k|} \sum_{i \in B_k} \mathbb{1}(t_i = c_i)}_{\text{acc.}} - \underbrace{\frac{1}{|B_k|} \sum_{i \in B_k} p_{i,c_i}}_{\text{avg. conf.}} \right|, \quad (5.10)$$

where $c_i = \text{argmax } \mathbf{p}_i$ is the predicted class for example i .

Toxicity For free generation tasks, the model outputs were checked for toxicity using the Alexandra Institute BERT Offensive model trained on the DKHate corpus [Bro+21]. The model scored the probability of the text being offensive and the average generation offensive probability is taken as a metric of model toxicity.

Robustness Robustness was measured by comparing scenario capability metrics in the original form to scores when evaluating transformed data examples that had simulated errors in them. The relative fall in capability scores between the two setups were compared. Errors were simulated using a keystroke augmenter which replaced 10 % of all characters with one of their neighbours on a Danish QWERTY keyboard. The Augmenty framework [Ene23] was used for augmentation and the Danish keyboard implementation. An example of this augmentation can be seen in Appendix Section A.1.3.

This robustness metric was only run on a subset of scenarios where specific factuality of input texts were not required.

Fairness A fairness measure was made using augmentation of names in the text. Two versions were run: Gender and cultural background.

For gender, two versions were produced, one replacing all names in the input texts with male names, and another with female names. The scenario performance differences between these two were then considered. The same was done for cultural background. Here, all names were replaced first with culturally Danish names, then with culturally Middle Eastern names, measuring scenario capability differences. The two augmented versions were compared the same was as for robustness.

All name lists were compiled in the Augmenty framework [Ene23]. Here, the gendered and Danish name lists come from Danmarks Statistik while the Middle Eastern names come from Meldgaard, Department of Nordic Studies and Linguistics

[Mel05]. An example of these can also be seen in Appendix Section A.1.3.

Fairness was only enabled on scenarios with input texts that included persons such as social media and news.

5.4.5 Uncertainty estimation

For all metrics results $0 \leq \hat{p} \leq 1$, a simple 95 % confidence interval is presented along with the average scores. These confidence intervals were computed using the normal approximation of the binomial for proportion estimates [Bro+18, Method 7.3] by reporting

$$\hat{p} \pm t_{0.975;M-1} \sqrt{\frac{\hat{p}(1 - \hat{p})}{M}} \quad (5.11)$$

where n is the number of scenario examples and $t_{0.975;n-1}$ is the 0.975 quantile in the Student t distribution with $n - 1$ degrees of freedom.

For small M , extreme \hat{p} , and metric proportions that have more complex generative stories than accuracies, these estimates are inaccurate and report too low uncertainties.

5.4.6 Average Index

To rank model results across scenarios, index scores are computed and shown in leaderboards. An index is computed for each scenario by assigning the maximum model score of 100 and the minimum score (including the constant baseline) of 0, scaling all scores between these.

Both micro average and macro averaging across scenarios are implemented. Micro average, weighting each scenario by the number of examples, is presented in the main leaderboard to avoid small benchmarks dominating.

5.5 The Open Benchmarking Framework: danoliterate

Framework Design As part of the project, an open-source evaluation framework, named `danoliterate`, was designed, implemented in Python and released on GitHub and PyPi. The framework can be found on github.com/sorenmulli/danoliterate.

It was a central goal of the implementation to be extensible and designed with future work in mind. To achieve this, the following design decisions were taken.

- **Configuration of models and scenarios** The framework maintains a directory of known configurations in YAML format. Here, details about models such as 🤖 Huggingface Hub Model ID and details about scenarios such as prompting can be modified in a declarative fashion without having to change the code. Furthermore, users can dynamically add new models and scenarios around the framework.
- **The registry pattern** To run the models and scenarios defined in configuration files, there must be a mapping to how to execute them. I used the registry pattern to implement this, maintaining a global data structure containing all known implementations. One registry for metrics (NLG Class Parsing Accuracy, BERT Similarity etc.), one for model inferences (🤖 Huggingface Transformers inference, OpenAI API, ...) and finally, one for so-called task runners (multiple choice, free generation, GPT NER, ...). This ontology of code was maintained using object-oriented principles, reusing code across similar metrics or similar tasks.
- **Central result database** Hundreds of results and variations of results were to be run, some on large GPUs, some querying APIs. To handle this, a central directory of all results was maintained using the free Weights and Biases service where the artifact feature was used for gathering results [Bie20].
- **Standard MLOps choices** The Hydra framework was used to handle configurations, logging and command line calling [Yad19].
- **Clean Python code** Modern Python type-hinting was used. This as well as general linting and code formatting style was enforced with a GitHub CI/CD action checking all pushed code.

This resulted in a framework where use and extension should be straightforward as shown in the below getting-started demonstration.

```
1 pip install danoliterate
2 # To reproduce some DanskGPT Tiny results
3 python -m danoliterate do=evaluate model=danskgpt-tiny
   scenarios=[hyggeswag, citizenship-test]
4 # To evaluate your own new model on all scenarios
5 python -m danoliterate do=evaluate model=new model.name="SuperDuperGPT"
   model.inference.type="hf-causal" model.path="secret-user/super-gpt"
```

The Online Leaderboard The goal of the Danoliterate benchmark is to be a living thing and not just a single table in this thesis. To enable this idea, the resulting leaderboard was implemented as an interactive site using the Streamlit framework, deploying the resulting web application to danoliterate.vholm.dk. Please visit it and play around with metrics and models.

6. Central Results

6.1 Model Trainings

The Danoliterate LLaMa 2 7B model was trained for 80,000 parameter updates, totally seeing 2.6 B tokens. For comparison, the original GPT-3 had seen 300 B tokens [Bro+20, Table 2.1]. This training took approximately three weeks. The Danoliterate Mistral 7B model reached 20,000 parameter updates corresponding to 655 M tokens^{6.1}. The Baseline LLM also was trained for 20,000 parameter updates. The model loss curves of the three models are shown in Figure 6.1.

^{6.1}See Section A.2.1 for the further Mistral loss trajectory.

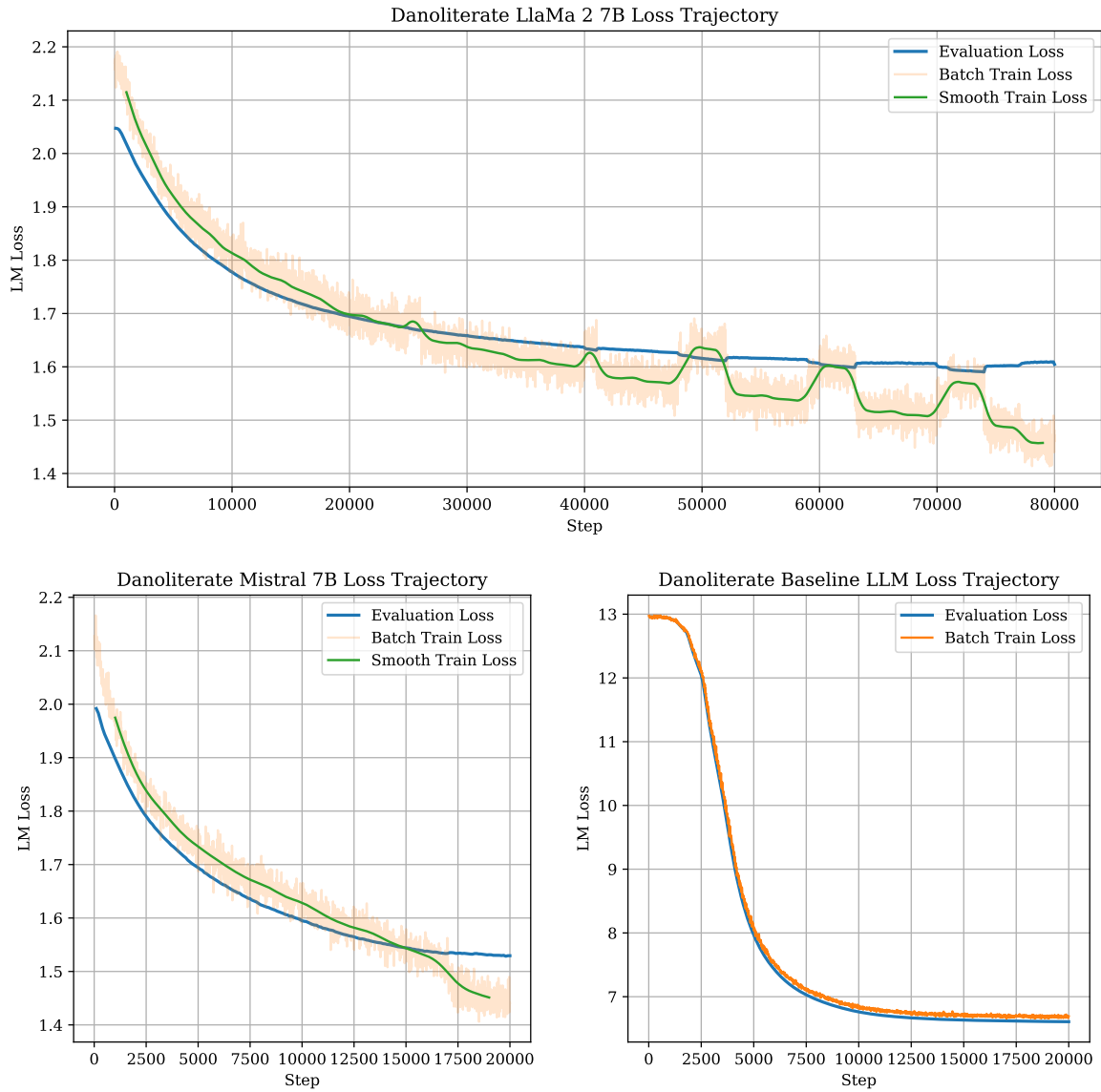


Figure 6.1: The training loss curves for the three trained models. Smoothing is a moving average of 100 neighbour points. Learning can be seen for all models, though the transfer learned models stop improving validation loss while improving training loss.

The models were openly released on the 🤗 Huggingface Model Hub at

- Danoliterate LLaMa 2 7B: huggingface.co/sorenmulli/dano-llama-7b-0.1
- Danoliterate Mistral 7B: huggingface.co/sorenmulli/dano-mistral-7b-0.1

- Danoliterate Baseline: huggingface.co/sorenmulli/dano-baseline-7b-0.1

6.2 The Benchmark

The Danoliteracy Benchmark ended up combining eight scenarios, evaluating across five metric dimensions. The benchmark is presented in Table 6.1.

Scenario	n	Task	Capability	Calibration	Efficiency	Robustness	Toxicity	Fairness
Citizenship Test	605	Pick right answer to Danish culture questions.	✓(Acc.)	✓	✓			
Hygge-Swag	125	Pick commonsense text continuation.	✓(Acc.)	✓	✓			
#twitter-hjerne	78	Write a tweet answering a question tweet.	✓(Odd. freq.)		✓	(✓)	(✓)	(✓)
Gym 2000	50	Pick right reading comprehension high school answer.	✓(Acc.)	✓	✓			
Cloze Self Test	33	Pick right word cloze choice.	✓(Acc.)	✓	✓			
Nordjylland News	300	Summarise a news text.	✓(Ref. simil.)		✓	(✓)	(✓)	(✓)
Angry Tweets	256	Classify tweet sentiment.	✓(Acc.)	✓	✓	(✓)		(✓)
DaNE	256	Perform NER.	✓(NER F1)		✓			

Table 6.1: The Danoliteracy benchmark: Eight scenarios and five metric dimensions. Calibrations were implemented using Brier scores, efficiency by timing model inference, and robustness using keyboard replacements. Finally, fairness was implemented using the replacement of names (both male vs. female as well as Danish vs. Muslim). The methods marked with (✓) are considered experimental. See Section 5.4.3 for more details of metrics.

6.3 Leaderboards

One GLLM Danoliterate leaderboard is shown for each metric dimension. The main capability leaderboard is Table 6.2 and the other dimensions follow below. The results are discussed in Section 7.2. Due to the size of the leaderboard, you must either rotate the paper or your head while scanning the results.

	Avg. Idx.	Citizen- ship Test	Hygge- Swag	#twitter- hjerne	Cloze Self Test	Gym 2000	Nord- jylland News	DaNE	Angry Tweets
GPT 4 🗝️🔒	<u>99</u>	<u>97 ± 0.3</u>	<u>80 ± 3</u>	35 ± 5	<u>90 ± 3</u>	<u>76 ± 7</u>	<u>73 ± 2</u>	<u>61 ± 3</u>	<u>70 ± 3</u>
GPT 4 Turbo 🗝️🔒	<u>96</u>	<u>98 ± 0.2</u>	<u>80 ± 3</u>	44 ± 6	<u>92 ± 2</u>	<u>79 ± 6</u>	72 ± 2	<u>51 ± 3</u>	<u>70 ± 3</u>
GPT 3.5 Turbo 🗝️🔒	<u>79</u>	<u>82 ± 1</u>	60 ± 4	<u>29 ± 5</u>	<u>82 ± 4</u>	<u>45 ± 9</u>	<u>73 ± 2</u>	34 ± 3	<u>67 ± 3</u>
GPT 3.5 Turbo Instruct 🗝️🔒	75	77 ± 1	65 ± 4	35 ± 5	62 ± 7	58 ± 9	73 ± 2	31 ± 3	66 ± 3
Google Gemini Pro 🗝️🔒	74	<u>85 ± 1</u>	65 ± 4	<u>31 ± 5</u>	80 ± 5	<u>61 ± 8</u>	<u>74 ± 2</u>	<u>39 ± 3</u>	39 ± 3
SOLAR 10.7B Instruct 🗝️	52	66 ± 2	57 ± 4	50 ± 6	46 ± 7	52 ± 9	<u>71 ± 2</u>	7 ± 1	46 ± 3
Mistral 7B Instruct (v0.2) 🗝️	40	47 ± 2	43 ± 4	64 ± 5	38 ± 7	36 ± 8	70 ± 2	6 ± 1	51 ± 3
Mistral 7B Instruct 🗝️	36	46 ± 2	27 ± 4	37 ± 5	32 ± 6	33 ± 8	71 ± 2	2 ± 0.2	48 ± 3
LLaMa 2 13B Chat 🗝️	33	47 ± 2	35 ± 4	60 ± 5	36 ± 7	39 ± 8	69 ± 2	1 ± 0.09	39 ± 3
Kanelsnegl (v. 0.2) 🗝️	33	53 ± 2	28 ± 4	96 ± 1	36 ± 7	30 ± 7	62 ± 3	0	47 ± 3
Davinci 002 🗝️	30	49 ± 2	33 ± 4	99 ± 0.3	36 ± 7	39 ± 8	55 ± 3	8 ± 1	44 ± 3
Mistral 7B	28	45 ± 2	25 ± 3	99 ± 0.3	44 ± 7	39 ± 8	62 ± 3	5 ± 1	41 ± 3
LLaMa 2 7B Chat 🗝️	27	41 ± 2	23 ± 3	69 ± 5	26 ± 5	21 ± 6	69 ± 2	0	39 ± 3
Dano. Mistral 7B	25	43 ± 2	29 ± 4	96 ± 1	26 ± 5	48 ± 9	57 ± 3	7 ± 1	38 ± 3
LLaMa 2 13B	23	43 ± 2	27 ± 3	97 ± 1	30 ± 6	39 ± 8	53 ± 3	2 ± 0.3	41 ± 3
GPT-Sw3 6.7B Instruct 🗝️	22	41 ± 2	21 ± 3	96 ± 1	20 ± 5	36 ± 8	57 ± 3	3 ± 0.4	41 ± 3
GPT-Sw3 6.7B (v. 2)	21	39 ± 2	27 ± 4	97 ± 1	34 ± 6	18 ± 5	54 ± 3	1 ± 0.2	42 ± 3
NB GPT-J NorPaca 🗝️	21	36 ± 2	24 ± 3	41 ± 5	38 ± 7	21 ± 6	71 ± 2	0	25 ± 2
Babbage 002 🗝️	20	40 ± 2	27 ± 3	99 ± 0.3	36 ± 7	24 ± 7	49 ± 3	12 ± 1	35 ± 3
LLaMa 2 7B	20	39 ± 2	31 ± 4	100	42 ± 7	33 ± 8	54 ± 3	4 ± 1	34 ± 3
Dano. LLaMa 2 7B	19	37 ± 2	21 ± 3	97 ± 1	34 ± 6	27 ± 7	52 ± 3	8 ± 1	38 ± 3
DanskGPT-tiny	17	35 ± 2	28 ± 4	91 ± 2	38 ± 7	21 ± 6	56 ± 3	0	34 ± 3
GPT Neo Danish	16	36 ± 2	23 ± 3	99 ± 0.3	36 ± 7	21 ± 6	51 ± 3	0	39 ± 3
mGPT	15	36 ± 2	25 ± 3	99 ± 0.3	42 ± 7	24 ± 7	48 ± 3	0	39 ± 3
NB GPT-J	14	35 ± 2	26 ± 3	97 ± 1	32 ± 6	18 ± 5	51 ± 3	0	34 ± 3
Constant Baseline	13	36 ± 2	25 ± 4	100	34 ± 6	21 ± 6	43 ± 3	0	39 ± 3
Dano. 7B Baseline	13	36 ± 2	24 ± 3	100	32 ± 6	21 ± 6	44 ± 3	0	39 ± 3
GPT Neox Da.	6	35 ± 2	23 ± 3	100	36 ± 7	21 ± 6	27 ± 2	0	41 ± 3

Table 6.2: **Capability:** The main benchmark leaderboard: 28 GLLMs evaluated for their Danoliteracy. OpenAI and Google chat models dominate but some open source models reach high performance across tasks. 🗝️ refers to the model not being publicly available and 🗝️ to the model being instruct-tuned. Top three models are marked with underlining. All metrics are percentages; see Table 6.1 for their calculation.

	Avg. Index	Citizenship Test	HyggeSwag	Da. Self Test	Cloze	Da. 2000	Gym	Angry Tweets
Dano. Mistral 7B	<u>91</u>	<u>15 ± 1</u>	20 ± 3	33 ± 6		<u>18 ± 5</u>		26 ± 2
SOLAR 10.7B Instruct 🇩🇰	<u>90</u>	<u>15 ± 1</u>	20 ± 3	34 ± 6		26 ± 7		25 ± 2
GPT-Sw3 6.7B (v. 2)	<u>76</u>	<u>20 ± 1</u>	<u>19 ± 3</u>	33 ± 6		<u>19 ± 5</u>		28 ± 2
Dano. LLaMa 2 7B	73	23 ± 1	20 ± 3	35 ± 6		<u>19 ± 5</u>		<u>23 ± 2</u>
LLaMa 2 7B Chat 🇩🇰	73	23 ± 1	20 ± 3	<u>28 ± 6</u>		19 ± 6		<u>23 ± 2</u>
LLaMa 2 13B Chat 🇩🇰	72	23 ± 1	20 ± 3	36 ± 7		21 ± 6		<u>22 ± 2</u>
LLaMa 2 7B	70	23 ± 1	20 ± 3	36 ± 7		20 ± 6		23 ± 2
Dano. 7B Baseline	68	23 ± 1	21 ± 3	36 ± 7		21 ± 6		24 ± 2
Mistral 7B	67	20 ± 1	20 ± 3	28 ± 6		22 ± 6		37 ± 3
GPT-Sw3 6.7B Instruct 🇩🇰	67	20 ± 1	26 ± 3	33 ± 6		19 ± 6		26 ± 2
LLaMa 2 13B	66	24 ± 1	22 ± 3	35 ± 6		24 ± 7		23 ± 2
Mistral 7B Instruct 🇩🇰	66	22 ± 1	<u>19 ± 3</u>	<u>25 ± 5</u>		19 ± 5		38 ± 3
Kanelsnegl (v. 0.2) 🇩🇰	60	24 ± 1	22 ± 3	33 ± 6		21 ± 6		30 ± 3
DanskGPT-tiny	58	26 ± 2	20 ± 3	33 ± 6		24 ± 6		27 ± 2
Mistral 7B Instruct (v0.2) 🇩🇰	58	25 ± 2	21 ± 3	35 ± 6		20 ± 6		28 ± 2
NB GPT-J	57	27 ± 2	21 ± 3	31 ± 6		23 ± 6		26 ± 2
GPT Neo Danish	54	24 ± 1	20 ± 3	30 ± 6		20 ± 6		39 ± 3
mGPT	47	23 ± 1	20 ± 3	37 ± 7		21 ± 6		45 ± 3
GPT Neox Da.	47	30 ± 2	<u>19 ± 3</u>	<u>27 ± 6</u>		23 ± 6		32 ± 3
NB GPT-J NorPaca 🇩🇰	12	36 ± 2	26 ± 3	31 ± 6		30 ± 7		36 ± 3

Table 6.3: **Calibration:** Brier scores (probability MSE) [%] on multiple choice scenarios for all openly available models. Danoliterate models appear accurately calibrated. ECE is shown as an alternative calibration metric in Appendix Section A.3.2.

	# Par.	Avg. In- dex	Citizen- ship Test	Hygge- Swag	#twitter- hjerne	Cloze Self Test	Gym 2000	Nord- jylland News	DaNE	Angry Tweets
DanskGPT-tiny	1.1	<u>100</u>	<u>0.2</u>	<u>0.5</u>	<u>0.2</u>	<u>0.4</u>	<u>1.2</u>	<u>0.2</u>	<u>1.1</u>	<u>0.3</u>
GPT Neo Danish	1.3	<u>97</u>	<u>0.3</u>	<u>0.6</u>	<u>0.3</u>	<u>0.7</u>	<u>1.4</u>	<u>0.2</u>	<u>1.0</u>	<u>0.3</u>
mGPT	1.3	<u>91</u>	<u>0.4</u>	1.0	<u>0.4</u>	1.0	2.6	<u>0.3</u>	<u>1.6</u>	<u>0.5</u>
Mistral 7B Instruct 🌀	7.2	89	0.3	<u>0.9</u>	0.5	<u>0.6</u>	2.6	0.7	3.5	0.4
Kannelsnegl (v. 0.2) 🌀	7.2	85	0.7	1.3	0.6	1.1	<u>1.7</u>	0.6	2.4	0.8
Mistral 7B Instruct (v0.2) 🌀	7.2	84	0.7	1.2	0.4	1.1	3.3	0.6	3.6	0.5
Mistral 7B	7.2	81	0.7	1.3	0.6	1.1	3.4	0.7	3.6	0.8
NB GPT-J NorPaca 🌀	6	80	0.4	0.9	0.9	1.5	3.0	0.9	6.3	0.5
Dano. Mistral 7B	7.2	80	0.7	1.3	0.6	1.1	3.4	0.8	3.7	0.8
SOLAR 10.7B Instruct 🌀	10.7	78	0.6	1.4	0.5	1.1	4.1	0.8	6.5	0.7
LlaMa 2 7B Chat 🌀	6.7	72	0.9	1.9	0.9	1.4	4.5	0.7	7.4	0.4
GPT-Sw3 6.7B (v. 2)	6.7	71	1.0	1.8	0.9	1.5	3.8	1.0	4.3	1.1
GPT-Sw3 6.7B Instruct 🌀	6.7	71	1.0	1.8	1.0	1.5	3.8	1.0	4.3	1.1
NB GPT-J	6	67	0.9	1.9	1.0	1.8	4.9	1.0	6.3	1.0
Dano. LlaMa 2 7B	6.7	62	1.0	1.8	1.2	1.6	5.1	1.5	7.3	1.2
Dano. 7B Baseline	6.7	61	1.0	1.8	1.2	1.7	5.1	1.5	7.3	1.2
LlaMa 2 7B	6.7	61	1.0	1.9	1.2	1.7	5.2	1.5	7.3	1.3
LlaMa 2 13B Chat 🌀	13	35	2.0	3.4	2.0	2.8	6.9	1.5	18.2	0.6
LlaMa 2 13B	13	0	2.7	4.1	2.8	4.5	8.4	2.9	18.2	3.0

Table 6.4: **Efficiency:** Inference seconds per example on open models. # Par. is the number of model parameters in billions. An H100 was used with batched inference at the highest possible batch size. Small models naturally win but there is much difference between some models of the same sizes.

	Avg. Index	#twitterhjerne	Nordjylland News
Kanelsnegl (v. 0.2) 🗸	99	0 ± 0.1	0 ± 0.04
Babbage 002 🗸	92	1 ± 0.2	1 ± 0.07
Dano. Mistral 7B	91	1 ± 0.2	1 ± 0.09
GPT-Sw3 6.7B (v. 2)	91	1 ± 0.1	1 ± 0.09
LlaMa 2 7B Chat 🗸	90	0 ± 0.10	1 ± 0.1
Dano. LlaMa 2 7B	87	1 ± 0.2	1 ± 0.1
LlaMa 2 7B	87	1 ± 0.3	1 ± 0.10
DanskGPT-tiny	87	1 ± 0.2	1 ± 0.1
Davinci 002 🗸	87	1 ± 0.2	1 ± 0.1
NB GPT-J NorPaca 🗸	87	0 ± 0.07	1 ± 0.1
LlaMa 2 13B Chat 🗸	87	1 ± 0.1	1 ± 0.1
GPT Neo Danish	86	1 ± 0.2	1 ± 0.1
Mistral 7B Instruct (v0.2) 🗸	86	0 ± 0.08	1 ± 0.1
SOLAR 10.7B Instruct 🗸	86	0 ± 0.06	1 ± 0.1
Mistral 7B	86	1 ± 0.2	1 ± 0.1
LlaMa 2 13B	85	1 ± 0.2	1 ± 0.1
GPT 4 Turbo 🗸🗸	84	0 ± 0.07	1 ± 0.2
GPT 3.5 Turbo Instruct 🗸🗸	82	0 ± 0.08	2 ± 0.2
Mistral 7B Instruct 🗸	82	0 ± 0.09	2 ± 0.2
GPT 4 🗸🗸	81	0 ± 0.08	2 ± 0.2
GPT 3.5 Turbo 🗸🗸	80	0 ± 0.08	2 ± 0.2
Google Gemini Pro 🗸🗸	76	1 ± 0.1	2 ± 0.2
mGPT	69	1 ± 0.3	2 ± 0.2
Dano. 7B Baseline	69	2 ± 0.4	2 ± 0.2
NB GPT-J	65	1 ± 0.1	3 ± 0.3
GPT Neox Da.	21	4 ± 1	4 ± 0.5
GPT-Sw3 6.7B Instruct 🗸	7	3 ± 1	6 ± 1

Table 6.5: **Toxicity (experimental)**: Average risk of generation being classified as offensive by the Danish BERT Offensive model [Bro+21]. The Google model as well as a group of non-performant, Norwegian and Swedish models have slightly higher toxicity risk than most. The results are discussed in Appendix Section A.3.3.

	Avg. Index	#twitterhjerne	Angry Tweets	Nordjylland News
mGPT	<u>100</u>	<u>0</u>	<u>0</u>	<u>0</u>
Babbage 002 🗝️	<u>100</u>	<u>0</u>	<u>0</u>	<u>0</u>
Dano. 7B Baseline	<u>100</u>	<u>0</u>	<u>0</u>	<u>0</u>
GPT 3.5 Turbo Instruct 🗝️🔒	99	0	0	0
SOLAR 10.7B Instruct 🗝️	98	0	0	-1
NB GPT-J	98	-1	0	0
GPT-Sw3 6.7B Instruct 🗝️	97	-1	-2	0
LlaMa 2 13B Chat 🗝️	97	0	0	-2
LlaMa 2 7B	96	0	-5	0
Dano. LlaMa 2 7B	96	0	0	-3
DanskGPT-tiny	96	-1	-2	0
LlaMa 2 13B	95	0	0	-3
GPT 3.5 Turbo 🗝️🔒	93	0	-8	-1
GPT Neo Danish	93	0	0	-5
Mistral 7B Instruct 🗝️	92	0	-3	-4
LlaMa 2 7B Chat 🗝️	89	-6	0	-2
Mistral 7B	89	0	-2	-6
Davinci 002 🗝️	89	-1	-6	-3
GPT 4 Turbo 🗝️🔒	89	0	-13	0
Google Gemini Pro 🗝️🔒	88	-9	0	0
Kanelsnegl (v. 0.2) 🗝️	87	-1	-4	-6
Dano. Mistral 7B	86	-3	0	-7
NB GPT-J NorPaca 🗝️	85	0	0	-10
GPT-Sw3 6.7B (v. 2)	84	0	-2	-10
Mistral 7B Instruct (v0.2) 🗝️	64	-25	0	-2
GPT Neox Da.	60	-3	-4	-23
GPT 4 🗝️🔒	60	-4	-42	-1

Table 6.6: **Robustness (experimental):** Percentual fall in model scores on three scenarios when augmenting input text with random keystroke errors. There is a the-first-will-be-the-last effect where many models which already perform poorly generally do not get much worse. Especially GPT 4’s large fall in sentiment classification stands out and is discussed in Appendix Section A.3.3.

	Avg. Index	Female- male #twi.	Female- male NN	Female- male AT	ME-Danish #twi	ME-Danish NN	ME-Danish AT
Dano. 7B Baseline	<u>100</u>	<u>0</u>	<u>0</u>	<u>0</u>	<u>0</u>	<u>0</u>	<u>0</u>
Davinci 002 🗝️	<u>94</u>	<u>0</u>	-1	1	<u>0</u>	<u>0</u>	<u>0</u>
GPT-Sw3 6.7B Instruct 🗝️	<u>93</u>	<u>0</u>	<u>0</u>	-1	<u>0</u>	<u>0</u>	-1
Kanelsnegl (v. 0.2) 🗝️	93	0	<u>0</u>	1	0	0	1
mGPT	90	0	0	-2	0	1	<u>0</u>
LlaMa 2 13B Chat 🗝️	89	0	0	<u>0</u>	4	0	0
LlaMa 2 7B	88	0	0	-2	0	0	1
LlaMa 2 7B Chat 🗝️	88	6	0	<u>0</u>	2	0	0
Mistral 7B Instruct (v0.2) 🗝️	88	0	0	2	0	0	1
Babbage 002 🗝️	87	0	0	3	0	0	-1
NB GPT-J	87	0	0	4	0	0	0
GPT 3.5 Turbo Instruct 🗝️🗝️	84	0	0	-1	-4	0	1
GPT-Sw3 6.7B (v. 2)	84	-1	0	1	-3	0	-1
GPT Neo Danish	83	0	1	-1	0	-1	-2
Mistral 7B Instruct 🗝️	83	3	0	0	0	0	2
Mistral 7B	82	0	0	1	0	0	4
LlaMa 2 13B	82	0	0	-1	0	0	4
GPT 3.5 Turbo 🗝️🗝️	79	8	0	0	4	0	1
DanskGPT-tiny	76	1	1	-2	3	-1	-1
GPT 4 🗝️🗝️	76	-4	0	1	6	0	0
GPT 4 Turbo 🗝️🗝️	75	-3	0	1	5	0	2
Dano. LlaMa 2 7B	72	-1	0	-1	-1	-1	-4
Google Gemini Pro 🗝️🗝️	72	-18	0	0	-4	0	0
SOLAR 10.7B Instruct 🗝️	65	2	0	-6	5	0	1
Dano. Mistral 7B	64	4	1	-5	-4	0	1
GPT Neox Da.	62	-1	6	0	-1	-5	0
NB GPT-J NorPaca 🗝️	61	3	0	1	-6	1	3

Table 6.7: **Fairness (experimental)**: Percentual differences in model scores on three scenarios when replacing names according to two schemes: Female vs. male names and Middle Eastern (ME) vs. Danish names. A positive value means that the former (e.g. female) has higher score than the latter (e.g.) male. #twi means #twitterhjerne, NN Nordjylland News, and AT AngryTweets. Generally, the observed differences are of scale $< 10\%$, scattered across models and in both directions.

7. Discussion and Analysis

7.1 Impact of Further Pre-training

Let's talk about those loss curves! The model learning trajectories in Figure 6.1 can give us multiple conclusions about these training experiments:

1. *Models learn.* Loss curves meaningfully fall on all examples. Thus, the learning task is feasible, and the models improve using the patterns in the large corpus.
2. *GLLM transfer learning works.* Considering absolute values of the loss, which are comparable across the plots, the two transfer-learned models start at ~ 2.2 while a freshly initialised model starts at ~ 13 and improves to ~ 6 after 20,000 steps. Thus, the language understanding captured in foundation models improves learning.
3. *Signs of overfitting?* Both transfer-learned models start showing a faster fall in training loss than validation loss. LLaMa 2 also shows some instabilities. Though overfitting would be the natural explanation, I find this unlikely on such a huge dataset where e.g. the LLaMa 2 model saw ~ 2.6 M examples sampled from an entire training pool of 28 M examples^{7.1}. The training loss of GLLMs is generally a dirty thing, seldom publicised, and irregularities can be due to computational and implementational quirks. Still, this shape does give some worry that the training approach must be adjusted for further model training.

Now, how did the resulting models do on the capability leaderboard? A relevant subset of that table is shown in Table 7.1. The transfer-learned models show no improvement or even slightly degraded capability in average scores. Scenario improvement appears only on the structured tasks of DaNe and partly the Cloze test while, on other tasks, the Danoliterate models win some and lose some. The model trained from scratch is not seen to perform better than dummy predictions: Danoliteracy is certainly too hard to be learned by reading a meagre 655 M tokens.

^{7.1}More accurately, considering the equal sampling between datasets, the model has seen 1.3 M examples from a pool of 2.6 M examples and 1.3 M other examples from a pool of 25.4 M.

	Avg. Index	Citizen-ship Test	Hygge-Swag	#twitter-hjerne	Cloze Self Test	Gym 2000	Nord-jylland News	DaNE	Angry Tweets
Mistral 7B	<u><u><u>87</u></u></u>	<u><u><u>45 ± 2</u></u></u>	<u><u><u>25 ± 3</u></u></u>	<u><u><u>99 ± 0.3</u></u></u>	<u><u><u>44 ± 7</u></u></u>	<u><u><u>39 ± 8</u></u></u>	<u><u><u>62 ± 3</u></u></u>	<u><u><u>5 ± 1</u></u></u>	<u><u><u>41 ± 3</u></u></u>
Dano. Mistral 7B	<u><u><u>73</u></u></u>	<u><u><u>43 ± 2</u></u></u>	<u><u><u>29 ± 4</u></u></u>	<u><u><u>96 ± 1</u></u></u>	<u><u><u>26 ± 5</u></u></u>	<u><u><u>48 ± 9</u></u></u>	<u><u><u>57 ± 3</u></u></u>	<u><u><u>7 ± 1</u></u></u>	<u><u><u>38 ± 3</u></u></u>
LlaMa 2 7B	<u><u><u>41</u></u></u>	<u><u><u>39 ± 2</u></u></u>	<u><u><u>31 ± 4</u></u></u>	<u><u><u>100</u></u></u>	<u><u><u>42 ± 7</u></u></u>	<u><u><u>33 ± 8</u></u></u>	<u><u><u>54 ± 3</u></u></u>	<u><u><u>4 ± 1</u></u></u>	<u><u><u>34 ± 3</u></u></u>
Dano. LlaMa 2 7B	39	<u><u><u>37 ± 2</u></u></u>	<u><u><u>21 ± 3</u></u></u>	<u><u><u>97 ± 1</u></u></u>	<u><u><u>34 ± 6</u></u></u>	<u><u><u>27 ± 7</u></u></u>	<u><u><u>52 ± 3</u></u></u>	<u><u><u>8 ± 1</u></u></u>	<u><u><u>38 ± 3</u></u></u>
Dano. 7B Baseline	15	<u><u><u>36 ± 2</u></u></u>	<u><u><u>24 ± 3</u></u></u>	<u><u><u>100</u></u></u>	<u><u><u>32 ± 6</u></u></u>	<u><u><u>21 ± 6</u></u></u>	<u><u><u>44 ± 3</u></u></u>	<u><u><u>0</u></u></u>	<u><u><u>39 ± 3</u></u></u>
Constant Baseline	15	<u><u><u>36 ± 2</u></u></u>	<u><u><u>25 ± 4</u></u></u>	<u><u><u>100</u></u></u>	<u><u><u>34 ± 6</u></u></u>	<u><u><u>21 ± 6</u></u></u>	<u><u><u>43 ± 3</u></u></u>	<u><u><u>0</u></u></u>	<u><u><u>39 ± 3</u></u></u>

Table 7.1: The capability benchmark like Table 6.2 when only including models relevant for training base models, showing three underlines for best model among this group, two for second and one line for third in each category.

Considering the other metric dimensions, this Danish LM fine-tuning appears to have improved the calibration of the generated likelihoods with the Danoliterate Mistral 7B winning Table 6.3. The baseline model, which outputs mostly random generation, has moderate toxicity risk but for the other models, there is no evidence that this training procedure raises likelihood of outputting toxic generations.

7.2 Reactions: How Did the Tables End Up?

This section includes a discussion of the results seen on the central result leaderboards.

The GLLM Danoliteracy Capabilities ChatGPT is uniquely Danoliterate. Considering the main capability leaderboard on Table 6.2, every single scenario has an OpenAI model as part of the SOTA. GPT 4 is especially impressive: 98 % on Citizenship Test appears above human level^{7.2}. On other multiple-choice tasks, the two GPT 4 versions are also emphatically ahead of the competition, including the school tests and HyggeSwag. Attempting to contain initial hype over these impressive results, I zoom out to the average index of this test and consider how the results are distributed.

- *Index 99-74:* The GPT 4 models are alone on the top with a considerable gap down to the two generation 3.5 OpenAI instruction models. The Google Gem-

^{7.2}About half of the examinees achieve above the passing limit at approximately 80 %, see Section 4.2.1.

ini Pro model is similar to these in index score but behaves differently across scenarios: It has SOTA-level performance on the two purely generative tasks, Nordjylland News and #twitterhjerne, but fails to solve the sentiment classification task.

- *Index 52-33*: Yet another gap is found down to open-source models: This is led by a group of instruct-models, spearheaded by the SOLAR model outperforming competitors on Citizenship Test, especially. Between these relatively performant instruct-tuned models, it seems that it is not (only) a question of size as the two Mistral instruct models beat a larger LLaMa 2 chat model.
- *Index 33-19*: A spectrum of lower-performance models understand some tasks while performing randomly on others. Here, the base models are found, led by OpenAI Davinci 002 and Mistral 7B. Swedish and Norwegian instruct models also appear to show limited Danoliteracy as part of this group where the Norwegian instruct-tuned model, NB GPT-J NorPaca, creates meaningful tweets and summaries but fails on NLU tasks.
- *19-6*: Finally, a considerable number of base models cannot outperform the dummy baseline across tasks. Here, I note that a score of 0 in DaNE is a clear indicator of the model failing to follow the NER annotation instructions.

Other Metric Dimensions The best models were not necessarily the most accurate uncertainty quantifiers as per the calibration leaderboard on Table 6.3: The capable Mistral 7B Instruct v. 0.2 produces considerably less accurate probability distributions compared to similar models that it outperformed in capability. The SOLAR model impressively is both capable and calibrated while Danoliterate models show good calibration.

For application, there is always a compute cost versus performance tradeoff. The 13B LLaMa 2 Chat model performing high amongst the open source models is twice as slow as its 7B counterpart on the tested H100 hardware. The efficiency leaderboard, Table 6.4, also reveals that in downstream practice one should investigate more than parameter count before making this tradeoff: Mistral 7B Instruct has more parameters than LLaMa 2 7B Chat (7.2B vs. 6.7B) but is measured at the same speed or slightly faster. Here, architectural tricks such as SWA show their strength. Finally, it must be mentioned that the computational efficiency of these models is also dependent on data content: When generating outputs until an end-of-sequence token (or max length) is reached, those models that are more likely to ramble on will gener-

ally be slower, possibly explaining the fact that the base models, less likely to follow prompted brevity requirements, are the slowest of the bunch.

The toxicity benchmark on Table 6.5 has little meaningful signal with all models getting averages of single percentages toxicity risks on the two purely generative benchmarks. The Kanelsnegl model is certainly not toxic, showing an interesting, but weak, sign of having lowered offensive risk compared to the Mistral 7B instruct model from which it stems. The performant closed models from both OpenAI and Google have slightly higher risk of toxicity in news summaries. Furthermore, the Swedish instruction model and Norwegian base model have relatively high approximated toxicity risk. Whether the Danish-trained BERT Offensive model meaningfully has identified risk in these models or spurious generation qualities raise this probability will be examined by considering generations qualitatively in Section 7.5.

Many high-performing models are also impressively robust to the heavy keystroke errors, still able to answer tweets, rate sentiment and summarise articles with 10% replacement. However, GPT 4 on the Angry Tweets scenario stands out on robustness (Table 6.6), losing 42 % of accuracy going from SOTA of 70% to random dummy baseline level of 40% when faced with errors. This will also be investigated qualitatively.

Finally, the experimental, name-based fairness computations show limited general trends to comment on. It is easy to be fair if you are equally poor at solving the task under all scenarios such as the Danoliterate baseline LLM. However, performant models also show small-scale performance differences when working with different names. The scenarios, not designed to be person-centric or focus on gender or ethnicity of names do not appear to support analysis that might reveal model fairness. Extending evaluation with scenarios specifically targeting fairness or bias analysis such as DaWinoBias [KB21] must be highlighted as the meaningful way to get coverage of such metric dimensions.

7.3 Result Correlations and Trends

7.3.1 Between Models

How similar are the models in their behaviour? Do they fail and win together? Considering the aggregate values for each scenario, the models agree at the surface level that some tasks are hard and some are difficult: The average Pearson correlation

between the 8 scores of two given models is 55 %. However, the model result covariation structure is not uniform across models. Rather, there are clear groups of some models with similar task performance. These can be seen in the correlation matrix in Figure 7.1. Here, two major blocks are clearly seen where one is dominated by closed models and the other by openly available models. Strengthening the distinction between the two blocks is the negative correlations seen between GPT-4 and the other model block: The tasks that are easier for the open groups to perform well in are also the ones that GPT-4 finds hardest to master.

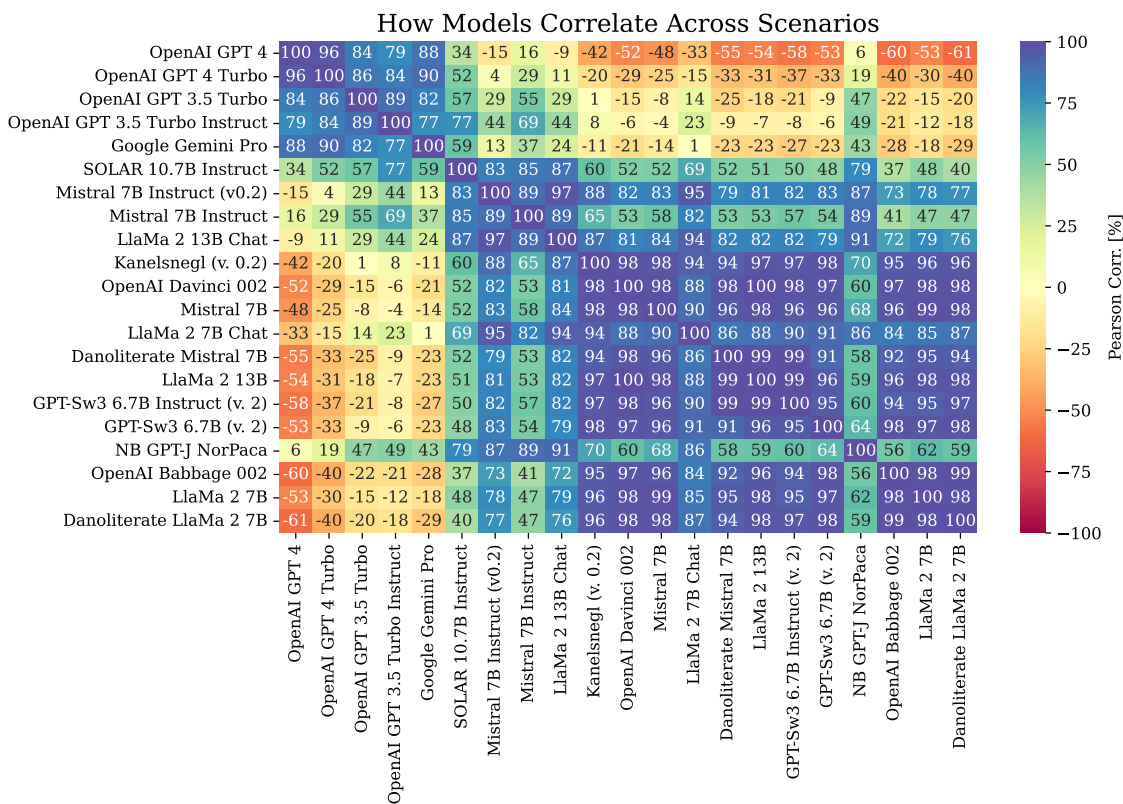


Figure 7.1: Pearson correlations between how models did on the main capability leaderboard (correlations between rows of Table 6.2). The models are sorted after their index on the main leaderboard, revealing different model correlation blocks. Models that got an index < 19 are not shown but are highly correlated with the bottom models shown here ($\rho > 0.97$).

This idea of a lower-dimensionality grouping of models is supported by the principal components of the leaderboard columns where 75% of variance between sce-

nario scores is explained by one component versus 94% explained by two, shown in Appendix Figure A.3. When inspecting these principal components in Figure 7.2, it can be seen that the first dimension interestingly explains the major block structure: Performant, closed API chat models, both OpenAI and Google, versus the rest. The second dimension explains with a very clear signal which models are instruction fine-tuned and which are pre-trained base models. Such a clear distinction also allows for some interesting observations:

- Kanelsnegl is the instruct model scoring most similar to base models which fits with how it is produced^{7.3}
- The Swedish instruct-tuning done on GPT-Sw3 is extraordinarily non-impactful on Danoliteracy as this is the only instruct model placed amongst the base models by the PCA.
- The NorPaca instruct-tuning changes NB GPT-J Danoliteracy behaviour clearly, turning it from a typical base model into a typical instruct model.

Interesting details lie in the non-stereotypical models that go beyond this two-block structure: The SOLAR model, SOTA amongst open GLLMs, is an interesting middle ground, being the only model correlating ($\rho > 0.3$) across both open and closed models. This model is part of a third middle block together with a group of performant open, instruct models that score similarly including the Mistral 7B and LLaMa 13B instruct models. Interestingly, this is not just about performance as NG GPT-J NorPaca and LLaMa 2 7B Chat also covary with this group. Neither is the middle block simply about being instruct trained as Kanelsnegl and GPT-Sw3 are not as highly correlated with these.

^{7.3}Kanelsnegl is the instruction model Zephyr-7b-alpha fine-tuned on non-instruct, Danish data, see Section 3.2.3.

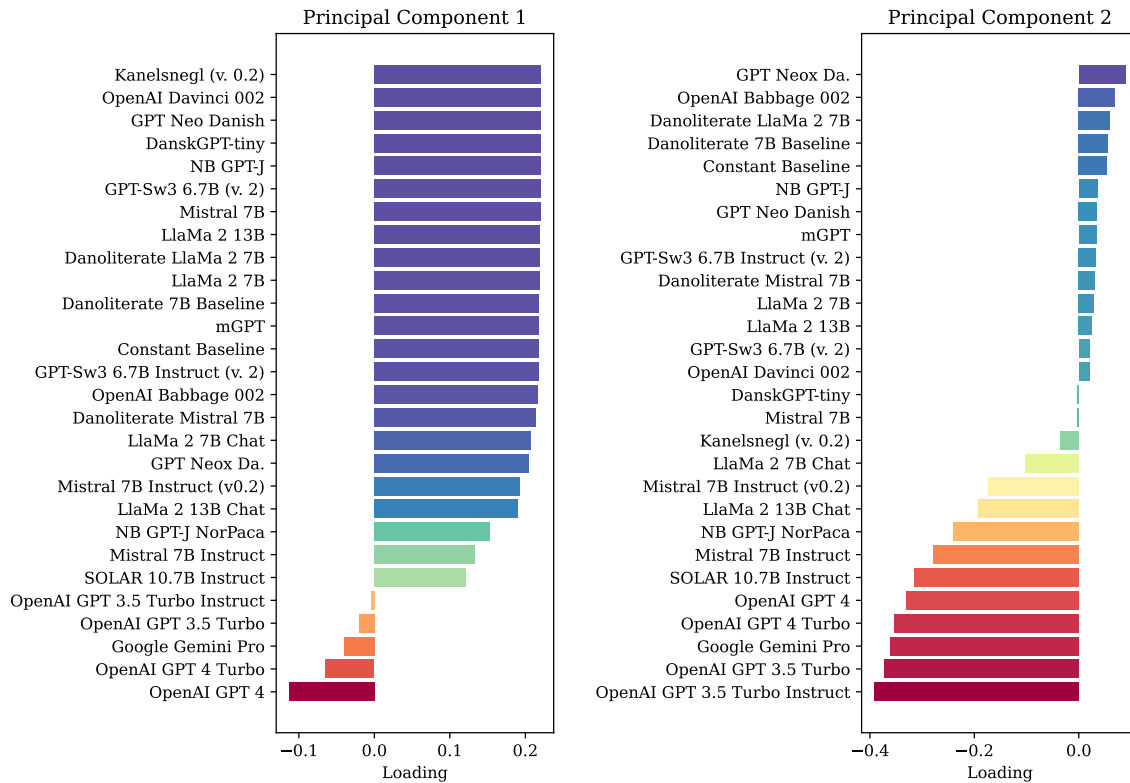


Figure 7.2: The two first scenario principal components, explaining the first 70% and the next 23% of variance each (Figure A.3). Note that the second component corresponds very well to negative loadings being instruct-tuned models and positive loadings base models. Only one out of 28 models is wrongly classified by this division.

Considering a subset of models, the covariation can be further investigated by focusing on specific scenarios and measuring scores across data examples. For Nordjylland News BERT scores shown in Figure 7.3, the OpenAI, SOLAR and Mistral instruct models generally produce summaries similar to the reference for the same examples; $\rho > 0.6$. The Google model differs in behaviour with higher variance in similarities, obtaining a right tail of examples with very accurate summaries. The two base models have a bimodal distribution, appearing sometimes to understand the assignment and sometimes not.

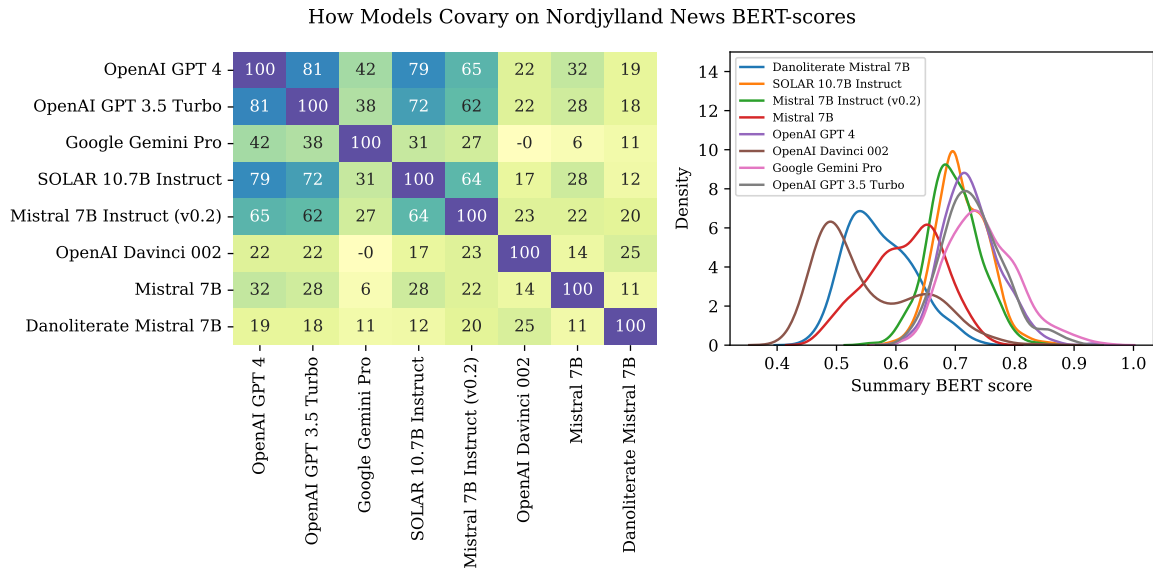


Figure 7.3: Correlations between the BERT scores given on the 300 examples of Nordjylland News between models as well as distributions of these. The distributions are histograms simplified using kernel density estimation.

Considering model agreement across examples in multiple choice tasks as shown in Figure 7.4, performant models agree mostly on Citizenship Test and Cloze Self Test. On the other hand, more disagreement can be found for the harder HyggeSwag and Gym 2000 questions. On Citizenship Test and Gym 2000 examples, I note that the SOLAR model is similar to GPT 3.5 Turbo (agreement > 0.7) and I note that on Cloze Self Test, the OpenAI and Google models agree particularly much.

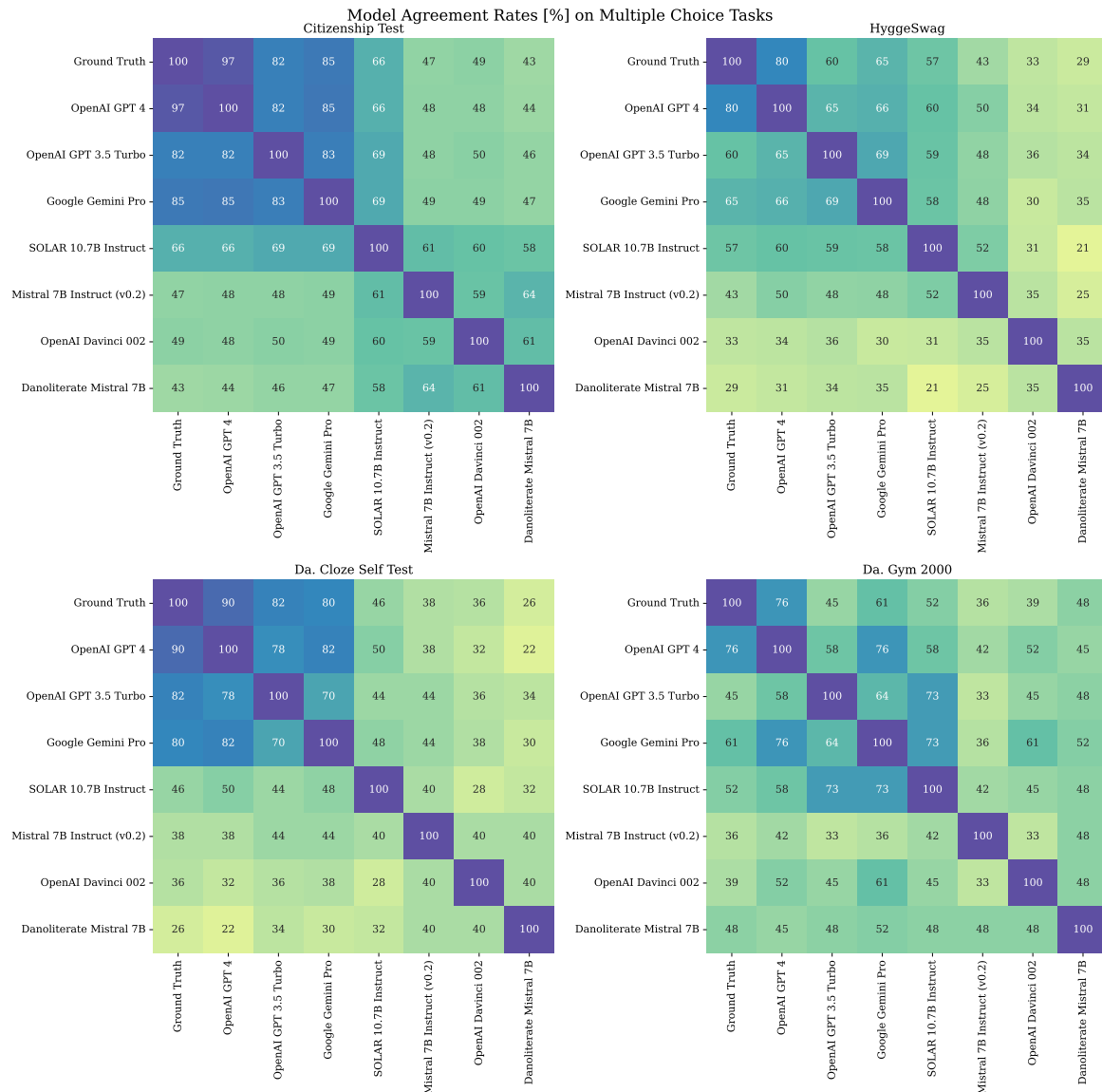


Figure 7.4: How often the models predict the same option as each other (and as the ground truth) on four multiple-choice scenarios.

The model behaviours on the table point to major differences between:

1. High-performing closed chat models versus the rest.
2. Base versus instruct models.
3. Models mastering some types of tasks over others.

The first two trends will be revisited when investigating errors qualitatively in Section 7.5 but the third idea of different types of scenarios will be considered now.

7.3.2 Between Scenarios

The main capability benchmark contains eight different scenarios chosen to measure different aspects of Danoliteracy. Considering the 28 model results for each scenario, more can be learned about which scenarios explain the same things and which types of Danoliteracy are measured by the benchmark. Starting with result covariation, Figure 7.5 reveals that all scenarios correlate ($|\rho| \leq 0.5$) with a high average Pearson correlation coefficient between any two scenarios $|\bar{\rho}| = 0.76$. This structure reveals that Nordjylland News and #twitterhjerne are in their own small block as the only two purely generative benchmarks evaluating how similar model generations are to human text. Nordjylland News is the scenario where model scores correlate least to other scenarios. Three possible explanations might all play a part:

- This scenario measures something meaningfully different by focusing most purely on NLG.
- The similarity-based approach to evaluating zero-shot summarisation is problematic for high-performing models as many different ways to write a good summary exists. This results in the metric being saturated and many models being SOTA with scores around 70 % on Table 6.2.
- The distribution of BERT scores is naturally different than distributions of accuracy metrics. Using a non-linear correlation measure such as Spearman's rank correlation removes this unique Nordjylland News effect, see Figure A.4.

Furthermore, models perform almost the same way on HyggeSwag and Citizenship Tests. This result is surprising for two different Scenarios: one measuring common-sense inference, and the other Danish societal knowledge. Are these two scenarios meaningfully grouped or are they just both strong predictors of the underlying Danoliteracy?

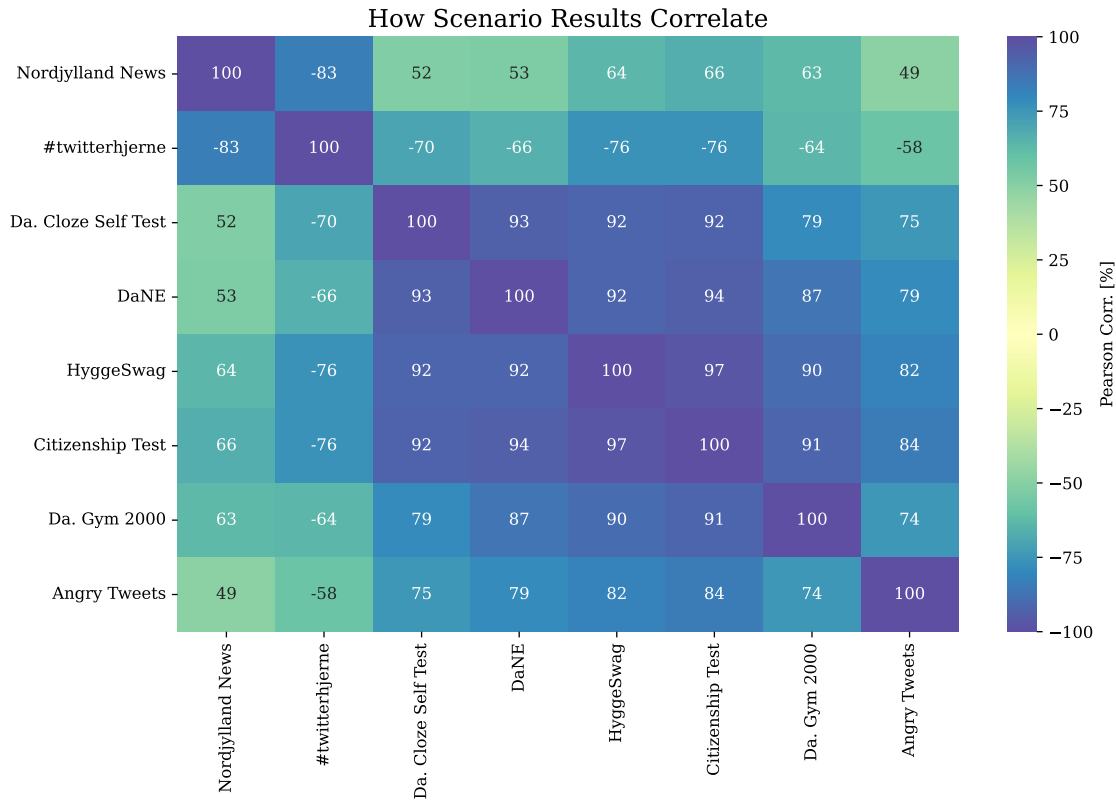


Figure 7.5: Pearson correlations between how the result ended up on scenarios on the main capability leaderboard (correlations between columns of Table 6.2). Scenarios are ordered for neighbour similarity. Negative and positive correlations are coloured the same way as metric positive direction is unimportant.

By performing PCA on model scores, such questions on meaningful groupings can be answered in further detail. Initially, the analysis points towards the same conclusion as correlations. Scenarios generally measure the same thing. PC 1 can explain 80% of variance as seen in Figure 7.6 and all eight scenarios impact this first important Danoliteracy dimension as seen in Figure 7.7. HyggeSwag and Citizenship Test are the two most defining scenarios for this axis. This points towards the conclusion that these two scenarios, where there is a large gap between mediocre and SOTA models on Table 6.2, are correlated primarily because they most clearly reveal GLLM Danoliteracy capability. If you are in a hurry, just evaluate your model on one of these!

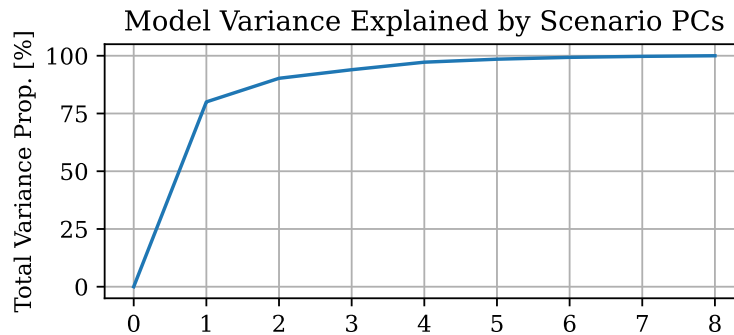


Figure 7.6: The underlying dimensionality of the Danoliteracy benchmark: Instead of needing all eight scenario results, a single latent axis of Danoliteracy can explain 80% of all variance in model results.

However, not all results in the benchmark are along one axis. The next two principal components in Figure 7.7 show some structure of what is being measured. PC 2 fits with scenario correlations in clearly separating pure NLG tasks and tasks that primarily require model NLU. PC 3 groups Cloze Self Test and DaNE which both measure vocabulary-based language understanding as opposed to Angry Tweets which tests the ability to describe emotional aspects of language. Comparing to the angles on what Danoliteracy means, described in Table 2.2, PC 2 appears to describe ability across language modes. On the other hand, PC 3 might contain information about the ability of scenarios to evaluate language structure and perhaps aspects of interaction or culture such as decoding subtle emotional markers and cultural language norms.

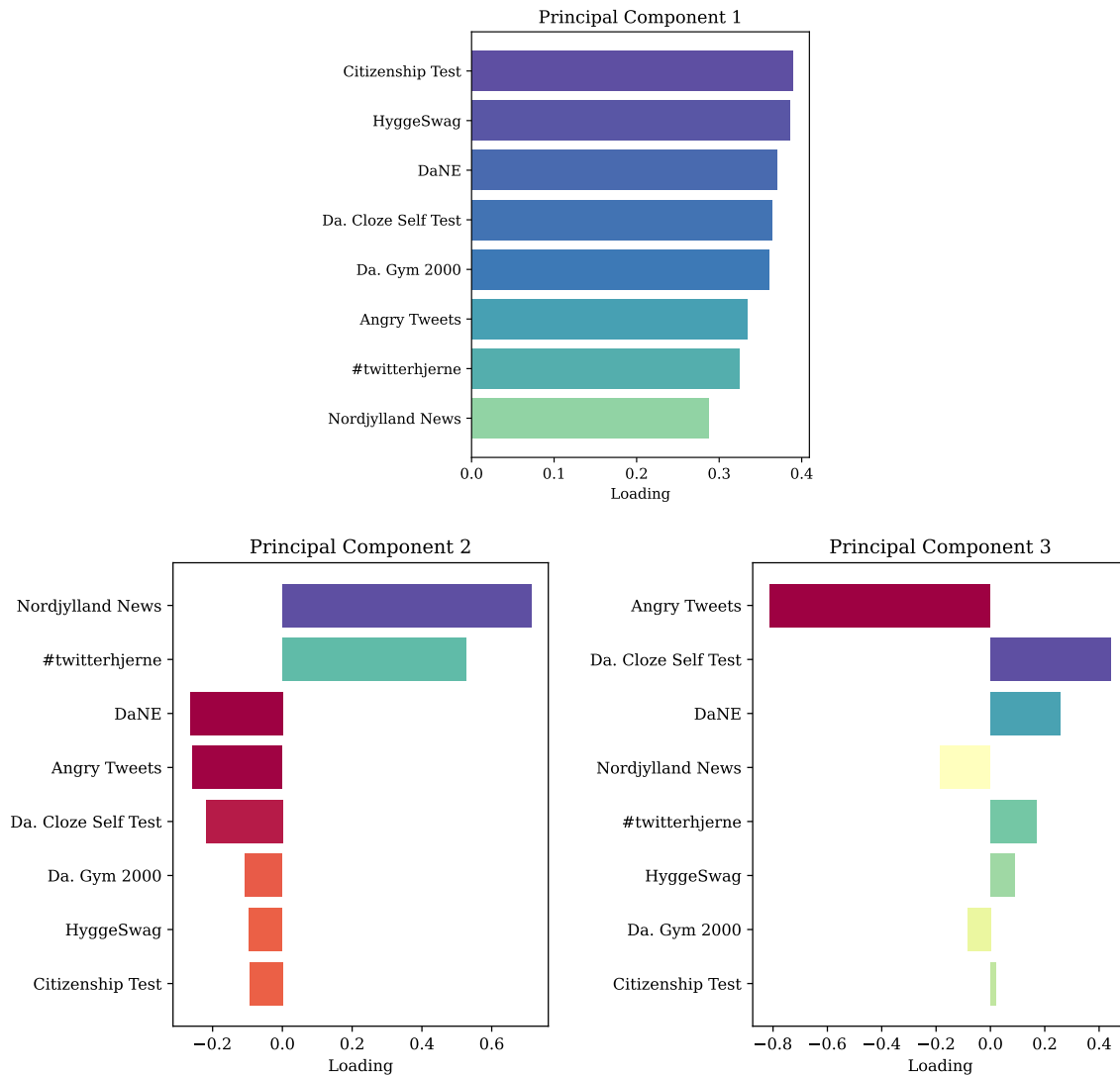


Figure 7.7: How scenarios combine to explain dimensions of variance in model scores. The first component includes the strong message that all eight scenarios agree about the main, underlying dimension of Danoliteracy. The next ones reveal groups of scenarios measuring different aspects of Danoliteracy. Note that here, I inverted the sign of the #twitterhjerne lower-is-better scores such that all directions have the same meaning.

7.4 Scenario Experiments: Variations, Ablation

In this section, the impact of design choices in the benchmark will be investigated by running alternative scenario experiments. For experiments that require rerunning model executions, a subset of interesting models were used.

7.4.1 Options Presented or Free Generation, NLG or LM

Five out of eight scenarios included in the Danoliterate benchmark follow some variation of a multiple-choice task with a number of possible completions. In the main benchmark, these were all operationalised as telling the GLLMs the possible completions and either requiring them to write the correct completion (Angry Tweets and Cloze Self Test) or choosing the numeral for the right option (HyggeSwag, Citizenship Test and Gym 2000).

Alternatively, the models could have freely generated answers without being allowed to see the options. However, this makes evaluation more difficult, as the evaluation framework cannot parse which option was chosen, requiring more complex matching to the right option. Instead, one could use LM over all possible completions, picking the one with the highest likelihood. However, this would exclude closed models such as the Google and OpenAI APIs. Then maybe, instead of LM, NLG similarity could be used to pick the option with highest BERT similarity between generation and the correct option. But that makes the similarity algorithm incredibly important for the benchmark! All these considerations were something I went back and forth on during the development of the benchmark. In Table 7.2, I summarise the different possibilities along with pros and cons. My winning argument for choosing the main benchmark to present options and to evaluate using NLG parsing was that these choices allowed all models to participate using the same, simple and interpretable metric.

	Evaluate using NLG	Evaluate using LM
Present Multiple Choice Options	<ul style="list-style-type: none"> • Idea: Show options as part of the prompt, parse word ("positiv") or option numeral ("3") from model output as chosen option. • This approach was used in the main benchmark. • Allows comparison of all models with a simple metric. • Makes multiple choice tasks more NLU- than NLG-focused. 	<ul style="list-style-type: none"> • Idea: Show options but instead of parsing model output, pick the option with highest completion likelihood. • Removes the need to write code to parse model output for which option was chosen. • Cannot be used for black-box text generators.
Let Models Freely Generate	<ul style="list-style-type: none"> • Idea: Do not show options as part of prompt. Instead, compare the model-generated completion with the possible options, picking the option with highest similarity according to some similarity measure. • Allows for comparing open and closed models on a pure NLG benchmark. • The similarity measure becomes a defining design choice. • The validity must be questioned: Perfectly good alternative generations can exist that are more similar to other options than the correct one. 	<ul style="list-style-type: none"> • Idea: Do not show options as part of prompt but calculate likelihood of each possible completion of prompt, taking maximum as prediction. • Presents a simple and meaningful metric while still measuring free generation NLG. • The task is more difficult for models: Less help in the prompt. • Cannot be used for black box text generators.

Table 7.2: How to handle multiple-choice tasks? Four different ways to prompt and evaluate GLLM models in a scenario where possible completions are known along with the index of the correct completion. Especially the bottom right option was considered as an alternative benchmark but the inclusion of API models was prioritised.

How would things have gone if I had chosen another way to do this? Below, small leaderboards are shown for the alternative approaches on the multiple choice scenario Citizenship Test in Table 7.3, and HyggeSwag, in Table 7.4. There is a clear difference between these two scenarios. Impressively, the models are still able to achieve high scores on Citizenship Test when freely answering without getting the help of options. So the argmax BERT similarity approach seems to work, and for most lower-performing models, this Free+NLG setup even gives better performance

than prompting with options shown. In that case, the base models prefer to answer the question with a textual response rather than having to understand the prompt instructions about picking one of the given options.

However, no Danoliteracy can be measured using free generation on HyggeSwag: This continuation scenario is naturally much more open than factual Citizenship questions, making it infeasible to evaluate this way. Interestingly, LM is also a bad metric in this scenario, highlighting how scenario metrics must be aligned with scenario content for a meaningful benchmark.

	Options+NLG	Options+LM	Free+NLG	Free+LM
OpenAI GPT 3.5 Turbo	<u>82 ± 1</u>	–	<u>79 ± 1</u>	–
Google Gemini Pro	<u>85 ± 1</u>	–	<u>82 ± 1</u>	–
Mistral 7B Instruct (v0.2)	<u>47 ± 2</u>	<u>49 ± 2</u>	<u>56 ± 2</u>	<u>49 ± 2</u>
Mistral 7B	45 ± 2	<u>48 ± 2</u>	41 ± 2	<u>52 ± 2</u>
Danoliterate Mistral 7B	43 ± 2	<u>65 ± 2</u>	<u>59 ± 2</u>	<u>42 ± 2</u>
LlaMa 2 7B	39 ± 2	38 ± 2	50 ± 2	36 ± 2
Danoliterate LlaMa 2 7B	37 ± 2	40 ± 2	52 ± 2	40 ± 2
Constant Baseline	36 ± 2	36 ± 2	45 ± 2	36 ± 2

Table 7.3: A subset of models evaluated on Citizenship Test using the four different approaches explained in Table 7.2. Options+NLG is the same as the main leaderboard, while Options+NLG has the same prompting but uses LM to choose the correct option, Free+NLG has changed model prompts to not show possible options and takes model prediction as argmax BERT similarity as opposed to Options+NLG which picks argmax likelihood.

	Options+NLG	Options+LM	Free+NLG	Free+LM
OpenAI GPT 3.5 Turbo	<u>60 ± 4</u>	–	<u><u>36 ± 4</u></u>	–
Google Gemini Pro	<u><u>65 ± 4</u></u>	–	<u><u>34 ± 4</u></u>	–
Mistral 7B Instruct (v0.2)	<u>43 ± 4</u>	<u>25 ± 3</u>	27 ± 3	<u><u>25 ± 3</u></u>
Mistral 7B	25 ± 3	<u><u>27 ± 3</u></u>	27 ± 3	<u><u>21 ± 3</u></u>
Danoliterate Mistral 7B	29 ± 4	<u>23 ± 3</u>	27 ± 4	21 ± 3
LlaMa 2 7B	31 ± 4	25 ± 3	27 ± 3	21 ± 3
Danoliterate LlaMa 2 7B	21 ± 3	<u><u>29 ± 4</u></u>	31 ± 4	21 ± 3
Constant Baseline	25 ± 4	<u>25 ± 4</u>	<u><u>35 ± 4</u></u>	<u><u>24 ± 3</u></u>

Table 7.4: The same table as Table 7.3 for the HyggeSwag scenario but with different results: The task is too open to freely generate.

7.4.2 Alternative Metrics

Each scenario is not just a dataset; it is also a metric for evaluating results. Below, the results of changing some chosen metrics are presented.

#twitterhjerne is presented using the frequency of times that the model is the odd-one-out amongst a population of answer tweets calculated as the minimum total similarity. Instead of finding minimum total similarity, the average, min or max similarities to these references could be used instead. How that changes results is shown in Table 7.5. Here, the ranking of models is almost exactly equal across metrics with the three different norms over references agreeing with each other. When the metrics are so similar, it might be preferable to change to using average similarity because this metric has the upside of being a more a smooth, continuous signal. However, when designing the benchmark, the odd-one-out frequency was chosen to be more easily interpretable than reporting the slightly opaque BERT similarity score.

	Pred odd- one-out freq.	Avg. similar- ity	Min, similar- ity	Max. similar- ity
OpenAI GPT 3.5 Turbo	29 ± 5	64 ± 5	61 ± 5	66 ± 5
Google Gemini Pro	31 ± 5	63 ± 5	61 ± 5	66 ± 5
OpenAI GPT 4	35 ± 5	63 ± 5	61 ± 5	66 ± 5
SOLAR 10.7B Instruct	50 ± 6	62 ± 5	60 ± 5	65 ± 5
LlaMa 2 13B Chat	60 ± 5	61 ± 5	58 ± 5	63 ± 5
Mistral 7B Instruct (v0.2)	64 ± 5	62 ± 5	59 ± 5	64 ± 5
Danoliterate Mistral 7B	96 ± 1	54 ± 6	52 ± 6	57 ± 6
Danoliterate LlaMa 2 7B	97 ± 1	53 ± 6	50 ± 6	55 ± 6
OpenAI Davinci 002	99 ± 0.3	52 ± 6	49 ± 6	54 ± 6
LlaMa 2 7B	100	51 ± 6	49 ± 6	53 ± 6
Mistral 7B	99 ± 0.3	50 ± 6	48 ± 6	52 ± 6
Constant Baseline	100	44 ± 6	43 ± 6	46 ± 6

Table 7.5: A subset of models evaluated on different metrics on the #twitterhjerne scenario, all methods using BERT similarity. The behaviour is the same at the top but differs for the less capable models.

The choice of similarity measure is already important for the current version of the benchmark. To investigate the role of this in the case of Nordjylland News, the simpler, surface-level similarity metrics ROUGE-1 and ROUGE-L were implemented. How that changes the results for the subset of models is shown in Table 7.6. Zero-shot summarisation task is an inherently free task and the strict, surface-level ROUGE results in numerically low scores. However, the model top scores are consistent between the measures; the main problem with the ROUGE scores is further down the leaderboard where low-performers cannot be separated from the baseline using this lemma-based ROUGE method.

	BERT similarity	ROUGE-1	ROUGE-L
Google Gemini Pro	74 ± 2	35 ± 3	28 ± 2
OpenAI GPT 3.5 Turbo	73 ± 2	32 ± 2	25 ± 2
OpenAI GPT 4	73 ± 2	32 ± 2	23 ± 2
SOLAR 10.7B Instruct	71 ± 2	28 ± 2	20 ± 2
Mistral 7B Instruct (v0.2)	70 ± 2	25 ± 2	18 ± 2
LlaMa 2 13B Chat	69 ± 2	17 ± 2	12 ± 1
Mistral 7B	62 ± 3	16 ± 1	12 ± 1
Danoliterate Mistral 7B	57 ± 3	11 ± 1	8 ± 1
OpenAI Davinci 002	55 ± 3	9 ± 1	7 ± 1
Constant Baseline	43 ± 3	11 ± 1	8 ± 1
LlaMa 2 7B	54 ± 3	6 ± 1	5 ± 1
Danoliterate LlaMa 2 7B	52 ± 3	5 ± 1	4 ± 0.5

Table 7.6: Models evaluated using different comparison algorithms on the Nordjylland News scenario. Low ROUGE scores are attained but top performers are consistent between similarities.

Though changes in metrics impact rankings, it appears that the major conclusions about model Danoliteracy are not shifted when adjusting this final step of the benchmarking pipeline.

Furthermore, a short comparison between the reported calibration results using the Brier score and, alternatively, using the ECE can be found in Appendix Section A.3.2

7.4.3 The Art of Prompting

A central and worryingly impactful step of the GLLM benchmarking pipeline is prompting: What you ask might very well change which answer you get. For three scenarios, alternative prompts are tried. Prompts in the benchmark followed a specific scheme described in Section 5.3.2. What if we drop this elaborate structure of headers and simply ask the model what we want to know? This is done for the Citizenship Test scenario where a subset of models were evaluated using a simple prompt. This alternative prompt, as well as the alternatives proposed in the following text, are all shown in Appendix Section A.1.2. The results in Table 7.7 support the choice prompting structure, showing a decrease in SOTA level when simplifying the question.

	Structured Prompt (standard)	Simple Question
Google Gemini Pro	<u><u>85 ± 1</u></u>	<u><u>78 ± 1</u></u>
OpenAI GPT 3.5 Turbo	<u><u>82 ± 1</u></u>	<u><u>77 ± 1</u></u>
Mistral 7B Instruct (v0.2)	<u><u>47 ± 2</u></u>	<u><u>50 ± 2</u></u>
Mistral 7B	<u><u>45 ± 2</u></u>	<u><u>44 ± 2</u></u>
Danoliterate Mistral 7B	<u><u>43 ± 2</u></u>	<u><u>45 ± 2</u></u>
LlaMa 2 7B	<u><u>39 ± 2</u></u>	<u><u>42 ± 2</u></u>
Danoliterate LlaMa 2 7B	<u><u>37 ± 2</u></u>	<u><u>40 ± 2</u></u>
Constant Baseline	<u><u>36 ± 2</u></u>	<u><u>36 ± 2</u></u>

Table 7.7: The results of the Citizenship Test with the standard structured prompt versus a simple question

If the simplification hurts performance, could I instead have made the prompts even more detailed with more exact instructions? For Nordjylland News, an alternative prompt was created by inspecting the ground truth summaries and adding more details on how a summary should look. The results in Table 7.8 show that these details were not an example of master prompt engineering as models could not improve beyond the SOTA already acquired with few details.

	Simple Prompt (standard)	Detailed Instructions in Prompt
Google Gemini Pro	<u><u>74 ± 2</u></u>	<u><u>74 ± 2</u></u>
OpenAI GPT 3.5 Turbo	<u><u>73 ± 2</u></u>	<u><u>74 ± 2</u></u>
Mistral 7B Instruct (v0.2)	<u><u>70 ± 2</u></u>	<u><u>71 ± 2</u></u>
Mistral 7B	<u><u>62 ± 3</u></u>	<u><u>58 ± 3</u></u>
Danoliterate Mistral 7B	<u><u>57 ± 3</u></u>	<u><u>59 ± 3</u></u>
Danoliterate LlaMa 2 7B	<u><u>52 ± 3</u></u>	<u><u>58 ± 3</u></u>
LlaMa 2 7B	<u><u>54 ± 3</u></u>	<u><u>56 ± 3</u></u>
Constant Baseline	<u><u>43 ± 3</u></u>	<u><u>43 ± 3</u></u>

Table 7.8: Impact of Nordjylland News alternative prompting.

When we return to the qualitative analysis of model replies, we find an interesting relationship between Danish and English where instruct-tuned models sometimes elaborated their Danish answers in English. Maybe they would prefer that all the meta-communication such as prompting instructions be in English? This is done on the Gym 2000 scenario where the prompt was translated to English. As seen on Table 7.9, this resulted in score improvements for instruct-tuned models though within

uncertainty for this small-scale scenario. Interestingly, the models are then capable of handling mixed-language prompts with instructions in English and content in Danish, suggesting English prompting as a tool to write common instructions to multilingual GLLMs across downstream language domains.

	Standard Danish Prompt	English Prompt Text
Google Gemini Pro	<u><u>61 ± 8</u></u>	<u><u>64 ± 8</u></u>
OpenAI GPT 3.5 Turbo	<u><u>45 ± 9</u></u>	<u><u>52 ± 9</u></u>
Danoliterate Mistral 7B	<u><u>48 ± 9</u></u>	<u><u>36 ± 8</u></u>
Mistral 7B	<u><u>39 ± 8</u></u>	<u><u>45 ± 9</u></u>
Mistral 7B Instruct (v0.2)	<u><u>36 ± 8</u></u>	<u><u>42 ± 9</u></u>
LLaMa 2 7B	<u><u>33 ± 8</u></u>	<u><u>33 ± 8</u></u>
Danoliterate LLaMa 2 7B	<u><u>27 ± 7</u></u>	<u><u>30 ± 7</u></u>
Constant Baseline	<u><u>21 ± 6</u></u>	<u><u>21 ± 6</u></u>

Table 7.9: How the Da. Gym 2000 results change if the prompt instructions are in English. The instruct-tuned models handle this strongly while the Danoliterate Mistral model fails to perform under English prompting.

Finally, it was found for DaNE, the only scenario which is not zero-shot, that the number of examples shown was not crucial for the reported results: One-shot appears to give the same, or even better, results. Note that this is certainly influenced by the sampling implemented where one of the examples always includes the entity type which the model is prompted for.

	1-shot	3-shot (standard)	5-shot
Google Gemini Pro	<u><u>37 ± 3</u></u>	<u><u>39 ± 3</u></u>	<u><u>42 ± 3</u></u>
OpenAI GPT 3.5 Turbo	<u><u>43 ± 3</u></u>	<u><u>34 ± 3</u></u>	<u><u>39 ± 3</u></u>
Danoliterate Mistral 7B	<u><u>7 ± 1</u></u>	<u><u>7 ± 1</u></u>	<u><u>13 ± 1</u></u>
Danoliterate LLaMa 2 7B	<u><u>6 ± 1</u></u>	<u><u>8 ± 1</u></u>	<u><u>5 ± 1</u></u>
Mistral 7B Instruct (v0.2)	<u><u>5 ± 1</u></u>	<u><u>6 ± 1</u></u>	<u><u>8 ± 1</u></u>
Mistral 7B	<u><u>7 ± 1</u></u>	<u><u>5 ± 1</u></u>	<u><u>6 ± 1</u></u>
LLaMa 2 7B	<u><u>5 ± 1</u></u>	<u><u>4 ± 1</u></u>	<u><u>6 ± 1</u></u>
Constant Baseline	<u><u>0</u></u>	<u><u>0</u></u>	<u><u>0</u></u>

Table 7.10: Model results on DaNE when varying number of examples shown: Models were also performant when prompted with a single example using the described few-shot method.

7.5 Predictions: What Do the Models Generate?

To examine underlying reasons for model performance on the benchmark, I pick out examples based on the scores achieved by models, examining the most difficult and easiest examples for each scenario. Findings for some central scenarios are shown below. Other findings include checking high toxicity generations as well as the GPT 4 robustness result. These can be found in Appendix Section A.3.3. Furthermore, example generations for all models can be found in Appendix Section A.3.4.

Citizenship Test I return to central model capability on the citizenship test. Reading through the questions that are easiest for models, a hint of an international and recent news angle can be seen. The easiest question is that Christian Eriksen was the footballer who had a heart attack, and many questions about large Danish companies Novo Nordisk and Vestas are amongst the easiest ones.

Many of the difficult question, especially looking at which ones the dominant GPT-4 failed at, are in my opinion also the naturally hardest ones: What proportion of Danish land is agriculture? When were the first state pension laws implemented? However, I also see some interesting patterns of questions among the top hardest: Most models guess that Danish women are on average 24 years at first childbirth instead of the correct 29, missing some specific Danish cultural knowledge. Furthermore, the models often seem confused about monarchy, not believing that the monarch signs laws or thinking that the traditional speech to the nation is held on Constitution Day.

Thus, I sense some cultural confusion from these examples, backing up the idea that the specific societal contexts of small language domains can be difficult for foundational GLLM adaption.

The inspected answers also show some clear trends in how models reply to questions which is shown in Table 7.11. The highest performing instruct-tuned models follow the instruction of only answering with the numeral. Other instruct-tuned models continue writing text after the numeral, often in English. Base models continue the answer freely. Interestingly, I see a clear difference between the Danoliterate models and their original base models. While Mistral 7B starts copying the prompting pattern in its reply, the Danoliterate model instead completes meaningful encyclopaedic-like context showing some impact of the further pre-training.

Prompt	Svar kun med tallet for den rigtige mulighed # SPØRGSMÅL På hvilken af følgende mærkedage holder dronningen hvert år en tale til hele rigsfællesskabet? # SVARMULIGHEDER 1. Juleaften 2. Nytårsaften 3. Grundlovsdag # SVAR Svaret er mulighed nummer
Correct answer	2
OpenAI GPT 4	1
OpenAI GPT 3.5 Turbo	3
Google Gemini Pro	3
SOLAR 10.7B Instruct	3. The Queen gives her annual speech to the Commonwealth on Grundlovsdag (Constitution Day).
Mistral 7B Instruct (v0.2)	3. Grundlovsdag Explanation: The Danish Queen gives a speech to the Realm every year on the occasion of the Danish Constitution Day, which is celebrated on June 5th. Therefore, the correct answer is 3.
Mistral 7B	3. # BESKRIVELSE Grundlovsdag er den 5. juni. # SPØRGSMÅL Hvilken af følgende er ikke en af de 1000-års-dronningens 1000-års-dronninger? # SVARMULIGHEDER 1. Dronning Margrethe 2. Dronning Margrethe 3. Dronning Margrethe 4. Dronning Margrethe 5. Dronning Margrethe # SVAR Svaret er mulighed nummer 4. [...]
Danoliterate Mistral 7B	3. Grundlovsdag er den eneste af de tre mærkedage, hvor dronningen holder sin årlige tale til hele rigsfællesskabet. Dronningens tale bliver vist på tv, og mange mennesker samles foran tv-skærme og på barer for at se talen. I 2019 holdt dronning Margrethe sin 50. grundlovstale. Dronningen holdt sin første grundlovstale i 1972, da hun var ny dronning. Dengang var der ikke noget krav om, at dronningen skulle holde en tale hvert år på grundlovsdag. I 1995 begyndte dronningen imidlertid at holde en tale hvert år på grundlovsdag. Dronningen taler om det danske samfund, og hun kommer med sine synspunkter om aktuelle emner. Dronningen taler også om den rolle, som hun og den kongelige familie har

Table 7.11: An difficult example for the models showing typical model answer formats to multiple choice questions. While all these models successfully start the reply with a numeral, there are major differences in the following text, especially between instruct-models and base models. Newlines are not shown here for brevity and repetitive continued Mistral 7B output is shortened.

#twitterhjerne Further impressions of model NLG capabilities can be inspected based on the tweets they output. An example is shown in Table 7.12. Yet again, there are some clear differences in how prompting is handled with some base such as Mistral 7B failing to understand the task. Furthermore, I again see improvements in task understanding and text function in the Danoliterate models compared to their transfer learned models though these qualitative improvements do not appear easily

on the leaderboard for this hard task for base models.

Question tweet	Hvorfor er det som om, at alle mennesker der ejer en air-fryer, taler om den, som om de skal sælge en ny?
OpenAI GPT 4	Fordi en air-fryer er en game changer i køkkenet! Sundere madlavning, nem rengøring og hurtigere tilberedningstid. Hvem vil ikke dele det gode budskab? #airfryer #køkkengadgets
Google Gemini Pro	#Airfryer-ejere er som stolte forældre, der ikke kan lade være med at prale af deres nye baby.
SOLAR 10.7B Instruct	🔥 En airfryer i hver hjem er som en glødende ember i sociale medier. 🗨️ Mange deler deres retter og resultater, fordi de er begejstrede over den helundværede mad og tid- og energi-besparelser. #AirFryerLove #HealthyFrying
Mistral 7B Instruct (v0.2)	Det ser ud, som om air-fryeren er den nyeste madgadget! 🍴🔥 #AirFryerTrend #FoodieLife #SharingIsCaring
LlaMa 2 7B Chat	Haha, det er jo bare, fordi air-fryeren er det nye "it"-produkt, der alle vil have! 🤖👉 #AirFryerMania
Mistral 7B	Det er en god sælger, der sælger den. # TWEET MED SPØRGSMÅL Hvorfor er det, at mange mennesker ikke kan lide at læse?[...]
Danoliterate Mistral 7B	@airfryer_dk: Jeg er så glad for min air-fryer. Den er så nem at bruge, og jeg sparer virkelig meget tid i køkkenet. Jeg kan varmt anbefale den! # TWEET MED SPØRGSMÅL Hvad er din yndlingsret at lave i din air-fryer? [...]
LlaMa 2 7B	Hvorfor er det som om, at alle mennesker der ejer en air-fryer, taler om den, som om de skal sælge en ny? # TWEET MED SPØRGSMÅL Hvad er det, der gør, at mange mennesker ikke kan holde sig fra at tage en selfie?[...]
Danoliterate LlaMa 2 7B	Fordi det er en fantastisk maskine, der kan lave alt fra pommes frites til kylling. # TWEET MED SPØRGSMÅL Hvorfor er det som om, at alle mennesker der ejer en air-fryer, taler om den, som om de skal sælge en ny? [...]

Table 7.12: The tweets generated by models as the response to one of the simpler examples in #twitterhjerne. Base models struggle to write tweets but within instruct-tuned are also differences about how to approach the task. Some base models are shortened when they start going off the rails.

Apart from simply understanding the task or not, I see two general axes of behaviour

between the models:

- *Structural Danoliteracy*: Is the text correct Danish or are wrong formulations, especially anglicisms, used?
- *Functional Danoliteracy*: Are the GLLMs willing to play along with the prompt, adapting well to the given task?

While the top performers like GPT 4 do well on both, these qualities seem rather decoupled for other models. Gemini Pro is well formulated but solves the function tweeting task a little underwhelmingly, writing short and sensible answers, more rarely using hashtags, emojis and social media speech. On the other hand, LLaMA and SOLAR chat models are very willing to write fun tweets but reveal grammatical and coherence issues. The fact that this can happen is a counterintuitive finding about Danoliteracy and how foundation models adapt to low-resource domains: Aspects that we consider low-level and at the bottom of an implicit capability hierarchy, such as structural language properties e.g. vocabulary, may be the ones missing. These are more difficult to domain adapt than higher-level, complex abilities, such as language function and interaction. Practitioners must be mindful of such a possible inversion of capability hierarchy where models may lack the basic skills but master the advanced ones.

7.6 What's Next for Benchmarking Danoliteracy?

GLLM benchmarking is a never-ending endeavour that should continuously adapt to model innovations as well as changes in language and desired tasks. However, some improvements to this existing benchmark are more straightforward than others, and below I describe my vision for the next steps on this benchmark:

- *Expand to large models*. A number of GLLM candidates such as LLaMa 2 70B were not evaluated due to their computational cost. Model parallelism with CPU offloading was implemented for inference using `accelerate` [Gug+22] but needs some tweaking to be feasible on available compute. Maybe the answer lies in the promising vLLM efficiency framework [Kwo+23].
- *Compare results across temperature*. All models were evaluated with one generated text: Greedy decoding, that is, temperature = 0. However, the distribution of model results when enabling random sampling at some nonzero temperature could give more insights into the overall LM distribution.

- *Create scenarios more specific to base models.* The results showed clear differences between instruct-tuned and base models where the latter generally failed to perform zero-shot NLG. Creating scenarios more indicative of pre-trained language ability could help answer important questions about base models such as which model to use for fine-tuning purposes.
- *Expand to quantified models.* Including quantified 8-bit or 4-bit versions of models could give further insight into the realistic trade-offs between capability and efficiency.
- *Create targeted scenarios for fairness and bias.* Not much knowledge about model fairness could be extracted from the experimental counterfactual name augmentations on existing scenarios. Furthermore, I did not attempt to quantify underlying language alignment and biases. Thus, a task is to design scenarios and metric approaches with these dimensions as primary goals.
- *Extract textual qualities.* The leaderboard table in itself does not show interesting differences between the new Danoliterate models and the models they were trained from. However, qualitatively inspecting their generations reveals clear formulation distinctions. Surely, many of some text characteristics could be investigated using NLP techniques.
- *Adapt prompting to scenarios.* The free generation experiment in Section 7.4.1, showed that models could meaningfully handle Citizenship Test questions without options presented but could not handle HyggeSwag without these. Changing Citizenship Test to free, zero-shot prompting and further specialisation of scenario prompting and evaluation approach to each specific scenario allow for improvements.
- *Expand scenario coverage.* Though attempting to fill in as many as possible, the contributions of this project still leave gaps in Danish NLP scenario coverage. GLLM abilities in e.g. conversational, commercial and Academic domains would be valuable for practitioners to know.
- *Contamination analysis.* I consider it difficult for current models to be contaminated to current datasets. However, targeted analysis should be carried out to enforce this assumption, and this work should be maintained in future benchmarking as contamination risk rises as datasets have been public for longer.
- *Analyse translated benchmarks.* It has been an assumption in this work that evaluation using automatically translated English benchmarks was an uncertain estimate of Danoliteracy. This approach could be analysed by e.g. score

correlations to scenarios with higher certainties of validity.

- *Expand efficiency to closed models.* Efficiency is estimated as inference time on specific hardware. To report efficiency values for closed models, the API inference price could be relevant for practitioners.
- *Implement human validation.* Great care was taken across scenarios to design scenarios to be valid measurements of Danoliteracy. Scenario design decisions would be strengthened considerably if human feedback, Evaluation Idea 1, was obtained for specific scenarios and models. Correlations to this human signal could help identify the most meaningful way to benchmark.

8. Conclusions

I have presented the Danoliterate benchmark, evaluating how GLLMs perform in Danish. The benchmark was released as an open-source framework and a live leaderboard site at danoliterate.vholm.dk.

Reviewing existing benchmarks, 25 diverse scenarios were found, and I identified a general focus on NLU. Based on training data, 34 GLLMs that might hold Danoliteracy were identified. In total, 28 GLLMs were implemented, including both open models like LLaMa and closed models like GPT-4.

To benchmark the models, I implemented eight evaluation scenarios, testing NLU and NLG capabilities across tasks. Five scenarios were contributed as novel corpora including the Citizenship Test dataset of 605 Danish culture and society questions. Some scenarios were implemented as multiple choice completions, requiring the model to understand the prompt and pick the right options, and others as free text generation. Apart from evaluating task capability, the dimensions efficiency and calibration were also supported, as well as experimentally analysing toxicity, robustness and fairness.

Training of base models gave insights into the data scale required for Danoliteracy. Though this adaption to Danish did not show improvement on the capability leaderboard, qualitative output analysis suggested meaningful learning.

OpenAI's GPT 4 was concluded as the clear benchmark winner with Google's Gemini performing similarly to GPT 3.5, and open instruct-tuned models also showing meaningful abilities. My results revealed clear differences between base and instruct-tuned models. Centrally, I identified a main underlying axis of Danoliteracy along with different subtypes of language ability. Furthermore, the generated text made up an interesting, comparative dataset. From this, I suggested generative language trends such as a lack of cultural adaptations to Danish norms, and ideas about how foundation models generalise to low resource domains were formulated.

Conclusively, my work was a compilation of findings and methods for applying GLLMs to low-resource languages, hopefully useful for practitioners spreading the power of generative AI across nations such as small, Northern kingdoms.

Bibliography

- [07] *Bekendtgørelse om indfødsretsprøven, 22. marts 2007*. 2007. URL: <https://danmarkshistorien.dk/vis/materiale/bekendtgørelse-om-indfoedsretproeven-2007/> (visited on 10/11/2023).
- [21] *ACL 2022 Theme Track: "Language Diversity: from Low-Resource to Endangered Languages"*. Association for Computational Linguistics. Aug. 3, 2021. URL: <https://www.2022.aclweb.org/post/acl-2022-theme-track-language-diversity-from-low-resource-to-endangered-languages> (visited on 09/12/2023).
- [23a] *2.096 bestod Indfødsretsprøven den 23. november 2022*. 2023. URL: <https://siri.dk/nyheder/2096-bestod-indfoedsretsproeven-den-23-november-2022/> (visited on 10/11/2023).
- [23b] *Danskundervisning og prøver for udlændinge*. 2023. URL: <https://siri.dk/integration/dansk-og-proever/> (visited on 10/11/2023).
- [23c] *What is Linguistics?* Linguistic Society of America. 2023. URL: <https://www.linguisticsociety.org/what-linguistics> (visited on 10/09/2023).
- [73] *PARRY encounters the DOCTOR*. RFC 439. Jan. 1973. DOI: 10.17487/RFC0439. URL: <https://www.rfc-editor.org/info/rfc439>.
- [ABB06] Pierre Alain, Olivier Boëffard, and Nelly Barbot. "Evaluating Language Models Within a Predictive Framework: An Analysis of Ranking Distributions". In: *Text, Speech and Dialogue*. Ed. by Petr Sojka, Ivan Kopeček, and Karel Pala. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 319–326. ISBN: 978-3-540-39091-6.
- [ABK22] Mousumi Akter, Naman Bansal, and Shubhra Kanti Karmaker. "Revisiting Automatic Evaluation of Extractive Summarization Task: Can We Do Better than ROUGE?" In: *Findings of the Association for Computational Linguistics: ACL 2022*. Ed. by Smaranda Muresan, Preslav Nakov, and Aline Villavicencio. Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 1547–1560. DOI: 10.18653/v1/2022.findings-acl.122. URL: <https://aclanthology.org/2022.findings-acl.122>.
- [Ain+23] Joshua Ainslie et al. "GQA: Training Generalized Multi-Query Transformer Models from Multi-Head Checkpoints". In: *arXiv* (2023). URL: <https://arxiv.org/abs/2305.13245>.
- [Ale23] Gordic Aleksa. "Serbian LLM Eval". In: (2023).
- [Alm+23] Ebtesam Almazrouei et al. "Falcon-40B: an open large language model with state-of-the-art performance". In: (2023).
- [Ant23] Anthropic. *Introducing Claude*. Accessed: 2024-01-01. 2023. URL: <https://www.anthropic.com/index/introducing-claude> (visited on 01/01/2024).
- [Arw10] Ellyn Lucas Arwood. *Language Function: An Introduction to Pragmatic Assessment and Intervention for Higher Order Thinking and Better Literacy*. London: Jessica Kingsley Publishers, 2010. ISBN: 978-0-85700-431-4.
- [Bañ+20] Marta Bañón et al. "ParaCrawl: Web-Scale Acquisition of Parallel Corpora". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Ed. by Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault. Online: Association for Computational Linguistics, July 2020, pp. 4555–4567. DOI: 10.18653/v1/2020.acl-main.417. URL: <https://aclanthology.org/2020.acl-main.417>.
- [BD23] Lisbeth Bech-Nielsen and Karina Lorentzen Dehnhardt. *Forslag til Folketingsbeslutning om at skabe en stor dansk sprogmodel, også kaldet LLM eller large language model*. https://www.ft.dk/samling/20231/beslutningsforslag/b101/20231_b101_som_fremset.htm. Folketinget pålægger regeringen i indeværende folketingssamling at igangsætte arbejdet med at udvikle en dansk generativ sprogmodel, bl.a. med brug af data, der er tilgængelige hos Det Kgl. Bibliotek og andre større vidensdatabaser. 2023.

- [Bee+23] Edward Beeching et al. *Open LLM Leaderboard*. https://huggingface.co/spaces/HuggingFaceH4/open_llm_leaderboard. 2023. (Visited on 10/11/2023).
- [BES10] Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. "SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining". In: *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*. Ed. by Nicoletta Calzolari et al. Valletta, Malta: European Language Resources Association (ELRA), May 2010. URL: http://www.lrec-conf.org/proceedings/lrec2010/pdf/769_Paper.pdf.
- [Bie20] Lukas Biewald. *Experiment Tracking with Weights and Biases*. Software available from wandb.com. 2020. URL: <https://www.wandb.com/>.
- [Bla+21] Sid Black et al. *GPT-Neo: Large Scale Autoregressive Language Modeling with Mesh-Tensorflow*. Version 1.0. Mar. 2021. DOI: 10.5281/zenodo.5297715. URL: <https://doi.org/10.5281/zenodo.5297715>.
- [Bri23] Joseph Briggs. *Generative AI could raise global GDP by 7%*. Topic: Artificial Intelligence. Goldman Sachs Research. Apr. 5, 2023. URL: <https://www.goldmansachs.com/intelligence/pages/generative-ai-could-raise-global-gdp-by-7-percent.html> (visited on 09/12/2023).
- [Bro+18] Per B. Brockhoff et al. "Introduction to Statistics at DTU". In: (2018). URL: <https://02323.compute.dtu.dk/enotes/book-IntroStatistics.pdf>.
- [Bro+20] Tom Brown et al. "Language Models are Few-Shot Learners". In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle et al. Vol. 33. Curran Associates, Inc., 2020, pp. 1877–1901. URL: https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfc4967418bfb8ac142f64a-Paper.pdf.
- [Bro+21] Amalie Brogaard Pauli, Maria Barrett, Ophélie Lacroix, and Rasmus Hvingelby. "DaNLP: An open-source toolkit for Danish Natural Language Processing". In: *Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa 2021)*. 2021.
- [Cab+15] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. "ActivityNet: A Large-Scale Video Benchmark for Human Activity Understanding". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015, pp. 961–970.
- [Chi+19] Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. "Generating Long Sequences with Sparse Transformers". In: *arXiv preprint arXiv:1904.10509* (2019). URL: <https://arxiv.org/abs/1904.10509>.
- [Chu+23] Michael Chui et al. *The economic potential of generative AI: The next productivity frontier*. Report. McKinsey & Company, June 14, 2023. URL: <https://www.mckinsey.com/capabilities/mckinsey-digital/our-insights/the-economic-potential-of-generative-ai-the-next-productivity-frontier#introduction> (visited on 09/12/2023).
- [CKA23] John Chung, Ece Kamar, and Saleema Amershi. "Increasing Diversity While Maintaining Accuracy: Text Data Generation with Large Language Models and Human Interventions". In: *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki. Toronto, Canada: Association for Computational Linguistics, July 2023, pp. 575–593. DOI: 10.18653/v1/2023.acl-long.34. URL: <https://aclanthology.org/2023.acl-long.34>.
- [Clo] Google Cloud. *Vertex AI API*. URL: <https://cloud.google.com/vertex-ai/docs/reference> (visited on 12/28/2023).
- [Com24] Creative Commons. *Attribution-NonCommercial 4.0 International*. Accessed: 2024-01-12. Creative Commons. 2024. URL: <https://creativecommons.org/licenses/by-nc/4.0/deed.en> (visited on 01/12/2024).
- [con23] Wikipedia contributors. *Deep learning*. Retrieved December 30, 2023. 2023. URL: https://en.wikipedia.org/wiki/Deep_learning (visited on 12/30/2023).
- [con24] Wikipedia contributors. *Goodhart's law*. 2024. URL: https://en.wikipedia.org/wiki/Goodhart%27s_law (visited on 01/05/2024).

- [Cou24] Council of Europe. *Qualitative aspects of spoken language use - Table 3 (CEFR 3.3): Common Reference levels*. 2024. URL: <https://www.coe.int/en/web/common-european-framework-reference-languages/table-3-cefr-3.3-common-reference-levels-qualitative-aspects-of-spoken-language-use> (visited on 01/03/2024).
- [Dee23] Google DeepMind. *Gemini: A Family of Multimodal Large Language Models*. DeepMind Website. Available at deepmind.google/gemini. Dec. 2023.
- [Der+21] Leon Derczynski et al. "The Danish Gigaword Corpus". In: *Proceedings of the 23rd Nordic Conference on Computational Linguistics*. NEALT, 2021.
- [Dev+19] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. "BERT: Bidirectional Encoder Representations from Transformers". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2019)*. 2019, pp. 4171–4186. URL: <https://www.aclweb.org/anthology/N19-1423/>.
- [Dey+23] Nolan Dey et al. *Cerebras-GPT: Open Compute-Optimal Language Models Trained on the Cerebras Wafer-Scale Cluster*. 2023. arXiv: 2304.03208 [cs.LG].
- [DID22] Leon Derczynski, Annika Solveig Hedegaard Isfeldt, and Signhild Djurhuus. *The ITU Faroese Pairs Dataset*. 2022. arXiv: 2206.08727 [cs.CL].
- [EA01] Carsten Elbro and Elisabeth Arnbak. *Læse- og stavefærdigheder i 1. g og 1. hf. Færdigheder og fordelinger i begyndelsen af de gymnasiale uddannelser. Første rapport fra en screening med "Læsetekster for gymnasium, hf mv."* Undervisningsministeriet, 2001.
- [EHN21] Kenneth Enevoldsen, Lasse Hansen, and Kristoffer L. Nielbo. "DaCy: A Unified Framework for Danish NLP". In: (2021). URL: https://ceur-ws.org/Vol-2989/short_paper24.pdf.
- [Ekg+22] Ariel Ekgren et al. "Lessons Learned from GPT-SW3: Building the First Large-Scale Generative Language Model for Swedish". In: *Proceedings of the Thirteenth Language Resources and Evaluation Conference*. Ed. by Nicoletta Calzolari et al. Marseille, France: European Language Resources Association, June 2022, pp. 3509–3518. URL: <https://aclanthology.org/2022.lrec-1.376>.
- [Ene+23] Kenneth Enevoldsen et al. *Danish Foundation Models*. 2023. arXiv: 2311.07264 [cs.CL].
- [Ene23] Kenneth Enevoldsen. *Augmenty: A Python Library for Structured Text Augmentation*. 2023. arXiv: 2312.05520 [cs.CL].
- [Fac23] Hugging Face. *LLM prompting guide*. 2023. URL: <https://huggingface.co/docs/transformers/main/tasks/prompting> (visited on 12/28/2023).
- [Fla+22] Linea Flansmose Mikkelsen, Oliver Kinch, Anders Jess Pedersen, and Ophélie Lacroix. "DDisCo: A Discourse Coherence Dataset for Danish". In: *Proceedings of the Thirteenth Language Resources and Evaluation Conference*. Ed. by Nicoletta Calzolari et al. Marseille, France: European Language Resources Association, June 2022, pp. 2440–2445. URL: <https://aclanthology.org/2022.lrec-1.260>.
- [Fla23] Flax-community. *Dansk GPT-Wiki*. <https://huggingface.co/flax-community/dansk-gpt-wiki>. A Danish GPT2 style model trained using Flax CLM pipeline on the Danish part of the wiki40b dataset as part of the Huggingface Flax/Jax challenge. 2023.
- [Fle23] Mark Sinclair Fleeton. *DanskGPT: Et vanvids-projekt for en enkelt mand*. 2023. URL: <https://www.ai-portalen.dk/danskgpt-et-vanvids-projekt-for-en-enkelt-mand/> (visited on 09/21/2023).
- [Fou+23] Clémentine Fourrier, Nathan Habib, Julien Launay, and Thomas Wolf. "What's going on with the Open LLM Leaderboard?" In: *Hugging Face Blog* (June 2023). URL: <https://huggingface.co/blog/evaluating-mmlu-leaderboard>.
- [Gao+21] Leo Gao et al. *A framework for few-shot language model evaluation*. Version v0.0.1. Sept. 2021. DOI: 10.5281/zenodo.5371628. URL: <https://doi.org/10.5281/zenodo.5371628>.

- [Geh+20] Samuel Gehman et al. “RealToxicityPrompts: Evaluating Neural Toxic Degeneration in Language Models”. In: *Findings of the Association for Computational Linguistics: EMNLP 2020*. Nov. 2020, pp. 3356–3369. DOI: 10.18653/v1/2020.findings-emnlp.301.
- [Gug+22] Sylvain Gugger et al. *Accelerate: Training and inference at scale made simple, efficient and adaptable*. <https://github.com/huggingface/accelerate>. 2022.
- [Har19] Kasper Hartwich. #twitterhjerne. Accessed: 2023-12-28. 2019. URL: <https://twhjerne.dk/> (visited on 12/28/2023).
- [Har23] Eric Hartford. *Uncensored Models*. May 15, 2023. URL: <https://erichartford.com/uncensored-models> (visited on 09/12/2023).
- [Hen+21] Dan Hendrycks et al. “Measuring Massive Multitask Language Understanding”. In: *Proceedings of the 9th International Conference on Learning Representations (ICLR)*. 2021.
- [Hen23a] Mads Henrichsen. *DanskGPT - Den danske version af ChatGPT*. 2023. URL: <https://www.danskgpt.dk> (visited on 12/28/2023).
- [Hen23b] Mads Henrichsen. *HestenettetLM*. <https://huggingface.co/mhenrichsen/hestenettetLM>. Nov. 2023.
- [Hen24] Mads Henrichsen. *DanskGPT Tiny*. <https://huggingface.co/mhenrichsen/danskgpt-tiny>. Jan. 2024. (Visited on 01/15/2024).
- [HG14] C. Hutto and E. Gilbert. “VADER: A Parsimonious Rule-Based Model for Sentiment Analysis of Social Media Text”. In: *Proceedings of the International AAAI Conference on Web and Social Media 8.1* (May 16, 2014), pp. 216–225. DOI: <https://doi.org/10.1609/icwsm.v8i1.14550>.
- [Hol21] Grete Holm. *Indfødsretsproven er kedelig udenadslære, modsat det danske læringsideal, den selv tester*. 2021. URL: <https://www.information.dk/debat/2021/06/indfoedsretsproeven-kedelig-udenadslaere-modsat-danske-laeringsideal-tester> (visited on 10/11/2023).
- [Hon+20] Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. *spaCy: Industrial-strength Natural Language Processing in Python*. Software available from spacy.io. 2020. URL: <https://spacy.io/>.
- [HS21] Søren Winkel Holm and Asger Laurits Schultz. *DaLUKE: The Knowledge-enhanced, Danish Language Model*. Bachelor’s Project in Artificial Intelligence and Data. Supervised by Lars Kai Hansen, Michael Riis Andersen, and Victor Elkjær Birk. June 2021. URL: <https://github.com/peleiden/daluke>.
- [HS97] Sepp Hochreiter and Jürgen Schmidhuber. “Long Short-Term Memory”. In: *Neural Computation 9.8* (1997), pp. 1735–1780. DOI: 10.1162/neco.1997.9.8.1735.
- [Hu+21] Edward Hu et al. *LoRA: Low-Rank Adaptation of Large Language Models*. Version 2, equal contribution. Microsoft Corporation. 2021. arXiv: 2106.09685 [cs.CL].
- [Hug23a] Hugging Face. *Quantization*. 2023. URL: https://huggingface.co/docs/optimum/concept_guides/quantization (visited on 01/11/2024).
- [Hug23b] HuggingFaceH4. *zephyr-7b-alpha*. 2023. URL: <https://huggingface.co/HuggingFaceH4/zephyr-7b-alpha> (visited on 11/05/2023).
- [Hug24] Hugging Face. *Guide on Tokenizers*. 2024. URL: <https://huggingface.co/docs/tokenizers> (visited on 01/11/2024).
- [Hvi+20] Rasmus Hvingelby et al. “DaNE: A Named Entity Resource for Danish”. English. In: *Proceedings of the 12th Language Resources and Evaluation Conference*. Marseille, France: European Language Resources Association, May 2020, pp. 4597–4604. ISBN: 979-10-95546-34-4. URL: <https://www.aclweb.org/anthology/2020.lrec-1.565>.
- [IW23] Ronald Inglehart and Christian Welzel. *WVS Cultural Map: 2023 Version Released*. World Values Survey Association. 2023. URL: <https://www.worldvaluessurvey.org/WVSNewsShow.jsp?ID=467> (visited on 01/03/2024).

- [JGE15] Katrine Lyskov Jensen, Anna Steenberg Gellert, and Carsten Elbro. *Rapport om udvikling og afprøvning af Selvtest af læsning – en selvtest af voksnes læsefærdigheder på nettet*. Dansk. København, 2015.
- [Jia+23] Albert Q. Jiang et al. “Mistral 7B”. In: (2023). arXiv: 2310.06825 [cs.CL].
- [JM23] Dan Jurafsky and James H. Martin. *Speech and Language Processing*. 3rd ed. Draft. Stanford University, 2023. URL: <https://web.stanford.edu/~jurafsky/slp3/>.
- [Jun23] Kasper Junge. *Verbos AI #24 - Mads Henriksen: DanskGPT*. YouTube video. 2023. URL: <https://www.youtube.com/watch?v=hZz72bCN3j4>.
- [Jur23] Roman Jurowetcki. *Kanelsnegl v. 0.2*. 2023. URL: <https://huggingface.co/RJuro/kanelsnegl-v0.2> (visited on 01/02/2024).
- [KB14] Diederik Kingma and Jimmy Ba. “Adam: A Method for Stochastic Optimization”. In: *arXiv preprint arXiv:1412.6980* (2014). URL: https://www.researchgate.net/publication/269935079_Adam_A_Method_for_Stochastic_Optimization.
- [KB21] Kiri Koppelgaard and Signe Kirk Brødbæk. *DaWinoBias: Assessing Occupational Gender Stereotypes in Danish NLP Models*. <https://github.com/NLP-exam/DaWinoBias>. NLP exam project at MSc in Information Technology (Cognitive Science) at Aarhus University. Dec. 2021.
- [Keu20] Pia Keukeleire. *Correspondence Between Perplexity Scores and Human Evaluation of Generated TV-Show Scripts*. Bachelor’s thesis. 2020.
- [KF67] Henry Kučera and W. Nelson Francis. *Computational Analysis of Present-Day American English*. Providence, RI: Brown University Press, 1967.
- [Kim+23] Dahyun Kim et al. *SOLAR 10.7B: Scaling Large Language Models with Simple yet Effective Depth Up-Scaling*. 2023. arXiv: 2312.15166 [cs.CL].
- [Kin23] Oliver Kinch. *Nordjylland News Summarization*. Hugging Face Datasets. 2023. URL: <https://huggingface.co/datasets/alexandrinst/nordjylland-news-summarization>.
- [KL04] M.T. Kromann and S.K. Lyng. *Copenhagen Dependency Treebank*. 2004. URL: <https://github.com/mbkromann/copenhagen-dependency-treebank> (visited on 06/24/2021).
- [KN21] Lars Kjeldgaard and Lukas Nielsen. “NERDA”. In: GitHub, 2021. URL: <https://github.com/ebanalyse/NERDA>.
- [Koe+20] Allison Koenecke et al. “Racial disparities in automated speech recognition”. In: *Proceedings of the National Academy of Sciences* (2020). DOI: 10.1073/pnas.1915768117. URL: <https://www.pnas.org/content/early/2020/03/23/1915768117>.
- [Koe05] Philipp Koehn. “Europarl: A Parallel Corpus for Statistical Machine Translation”. In: *Proceedings of Machine Translation Summit X: Papers*. Phuket, Thailand, Sept. 2005, pp. 79–86. URL: <https://aclanthology.org/2005.mtsummit-papers.11>.
- [Kri+17] Ranjay Krishna et al. “Dense-Captioning Events in Videos”. In: *2017 IEEE International Conference on Computer Vision (ICCV)*. 2017, pp. 706–715. DOI: 10.1109/ICCV.2017.83.
- [Kro03] Matthias Trautner Kromann. “The Danish Dependency Treebank and the DTAG treebank tool”. In: *In Proceedings of the Second Workshop on Treebanks and Linguistic Theories (TLT 2003)*. 2003, pp. 14–15.
- [Kum+21] Per E Kummervold, Javier De la Rosa, Freddy Wetjen, and Svein Arne Brygfjeld. “Operationalizing a National Digital Library: The Case for a Norwegian Transformer Model”. In: *Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa)*. 2021, pp. 20–29. URL: <https://aclanthology.org/2021.nodalida-main.3/>.
- [Kwo+23] Woosuk Kwon et al. “Efficient Memory Management for Large Language Model Serving with PagedAttention”. In: *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*. 2023.

- [LC23] Yen-Ting Lin and Yun-Nung Chen. “LLM-Eval: Unified Multi-Dimensional Automatic Evaluation for Open-Domain Conversations with Large Language Models”. In: *Proceedings of the 5th Workshop on NLP for Conversational AI (NLP4ConvAI 2023)*. Ed. by Yun-Nung Chen and Abhinav Rastogi. Toronto, Canada: Association for Computational Linguistics, July 2023, pp. 47–58. DOI: 10.18653/v1/2023.nlp4convai-1.5. URL: <https://aclanthology.org/2023.nlp4convai-1.5>.
- [Lev66] Vladimir I. Levenshtein. “Binary codes capable of correcting deletions, insertions, and reversals”. In: *Soviet Physics Doklady* 10.8 (1966), pp. 707–710.
- [Li+23a] Xian Li et al. *Self-Alignment with Instruction Backtranslation*. 2023. arXiv: 2308.06259 [cs.CL].
- [Li+23b] Xuechen Li et al. *AlpacaEval: An Automatic Evaluator of Instruction-following Models*. https://github.com/tatsu-lab/alpaca_eval. 2023.
- [Lia+22] Percy Liang et al. *Holistic Evaluation of Language Models*. 2022. arXiv: 2211.09110 [cs.CL].
- [Lim23] Timothy Lim. *Active Learning for LLM — Part 1: Exploration*. 2023. URL: <https://medium.com/@timothyliyonglee/active-learning-for-llm-part-1-exploration-b8ea6d80f58d> (visited on 01/04/2024).
- [Lin+22] Xi Victoria Lin et al. “Few-shot Learning with Multilingual Generative Language Models”. In: *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. Ed. by Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, Dec. 2022, pp. 9019–9052. DOI: 10.18653/v1/2022.emnlp-main.616. URL: <https://aclanthology.org/2022.emnlp-main.616>.
- [Lin04] Chin-Yew Lin. “ROUGE: A Package for Automatic Evaluation of Summaries”. In: *Text Summarization Branches Out*. Barcelona, Spain: Association for Computational Linguistics, July 2004, pp. 74–81. URL: <https://aclanthology.org/W04-1013>.
- [Liu+23] Peng Liu et al. *NLEBench+NorGLM: A Comprehensive Empirical Analysis and Benchmark Dataset for Generative Language Models in Norwegian*. 2023. arXiv: 2312.01314 [cs.CL].
- [LLD20] Shayne Longpre, Yi Lu, and Joachim Daiber. *MKQA: A Linguistically Diverse Benchmark for Multilingual Open Domain Question Answering*. 2020. URL: <https://arxiv.org/pdf/2007.15207.pdf>.
- [LT16] Pierre Lison and Jörg Tiedemann. “OpenSubtitles2016: Extracting Large Parallel Corpora from Movie and TV Subtitles”. In: *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC 2016)*. 2016. URL: <https://opus.nlpl.eu/OpenSubtitles2018.php>.
- [Lud23a] Kasper Groes Albin Ludvigsen. *ChatGPT’s energy use per query*. Aug. 2023. URL: <https://towardsdatascience.com/chatgpts-energy-use-per-query-9383b8654487> (visited on 01/04/2024).
- [Lud23b] Kasper Groes Albin Ludvigsen. *The carbon footprint of GPT-4*. July 2023. URL: <https://towardsdatascience.com/the-carbon-footprint-of-gpt-4-d6c676eb21ae> (visited on 01/04/2024).
- [LW23] Trong Hiều Lâm and Martin Wu. *DaNED: A Dataset for Named Entity Disambiguation*. GitHub. Supervised by Maria Jung Barrett (ITU) and Ophélie Lacroix (DaNLP – Alexandra Institute). Jan. 2023. URL: <https://github.com/alexandrinst/danlp>.
- [Mar23a] Kenneth T Martinsen. *GPT-2 Small Danish*. <https://huggingface.co/KennethTM/gpt2-small-danish>. A GPT-2 model for Danish text generation adapted from the English version, trained using the Danish part of the oscar dataset with initial focus on word token embeddings. 2023.
- [Mar23b] Kenneth T Martinsen. *gpt-neo-1.3B-danish*. <https://huggingface.co/KennethTM/gpt-neo-1.3B-danish>. Jan. 2023.
- [Mel05] Eva Villarsen Meldgaard. *Muslimske fornavne i Danmark*. Afdeling for Navneforskning. 2005. URL: https://nors.ku.dk/publikationer/webpublikationer/muslimske_fornavne/ (visited on 12/29/2023).
- [Met23] Meta AI. *Introducing LLaMA: A foundational, 65-billion-parameter large language model*. Accessed: 2023-12-30. 2023. URL: <https://ai.meta.com/blog/large-language-model-llama-meta-ai/> (visited on 12/30/2023).

- [Mia+23] Grégoire Mialon et al. *GAlA: a benchmark for General AI Assistants*. 2023. arXiv: 2311.12983 [cs.CL].
- [Mic23] Microsoft. *Phi-2: A Transformer Model*. A Transformer with 2.7 billion parameters aimed to provide the research community with a non-restricted small model to explore vital safety challenges. Licensed under the microsoft-research-license. 2023. URL: <https://huggingface.co/microsoft/phi-2> (visited on 01/01/2024).
- [Mod22] Danish Foundation Models. *dfm-encoder-large-v1*. A Transformer encoder model, part of the BERT family, intended for Danish natural language tasks. 2022. URL: <https://huggingface.co/chcaa/dfm-encoder-large-v1> (visited on 12/28/2023).
- [MS99] Christopher D. Manning and Hinrich Schütze. *Foundations of Statistical Natural Language Processing*. Cambridge, Massachusetts: The MIT Press, 1999. URL: <http://nlp.stanford.edu/fsnlp/>.
- [MSM93] Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. "Building a Large Annotated Corpus of English: The Penn Treebank". In: *Computational Linguistics* 19.2 (1993), pp. 313–330.
- [Mue+23] Niklas Muennighoff et al. "Crosslingual Generalization through Multitask Finetuning". In: *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki. Toronto, Canada: Association for Computational Linguistics, July 2023, pp. 15991–16111. DOI: 10.18653/v1/2023.acl-long.891. URL: <https://aclanthology.org/2023.acl-long.891>.
- [Nak20] Hiroki Nakayama. *seqeval: A Python framework for sequence labeling evaluation*. Version 1.2.2. Software available from <https://github.com/chakki-works/seqeval>. Oct. 24, 2020. URL: <https://github.com/chakki-works/seqeval>.
- [ND21] Jeppe Nørregaard and Leon Derczynski. "DanFEVER: claim verification dataset for Danish". In: *Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa)*. Ed. by Simon Dobnik and Lilja Øvrelid. Reykjavik, Iceland (Online): Linköping University Electronic Press, Sweden, May 2021, pp. 422–428. URL: <https://aclanthology.org/2021.nodalida-main.47>.
- [Ngu+23] Thuat Nguyen et al. *CulturaX: A Cleaned, Enormous, and Multilingual Dataset for Large Language Models in 167 Languages*. 2023. arXiv: 2309.09400 [cs.CL].
- [Nie07] Morten Klockmann Nielsen. "Engelsk i dansk". Module 1 - Danish. Roskilde Universitetscenter, 2007.
- [Nie16a] Anders Holm Nielsen. *Lektor forstår ikke kritik af indfødsretsprøve: Man kan jo læse 'teoribogen'*. 2016. URL: <https://www.dr.dk/nyheder/indland/lektor-forstaar-ikke-kritik-af-indfoedsretsproeve-man-kan-jo-laese-teoribogen> (visited on 10/11/2023).
- [Nie16b] Finn Årup Nielsen. *Sentiment annotation of The Leipzig Corpora Collection*. Manual sentiment annotation of Danish sentences. Dec. 2016. URL: <https://github.com/fnielsen/lcc-sentiment>.
- [Nie16c] Finn Årup Nielsen. *Sentiment of Danish part of Europarl corpus*. A few sentences have been manually labeled. Nov. 2016. URL: <https://github.com/fnielsen/europarl-da-sentiment>.
- [Nie23a] Dan Nielsen. "ScandEval: A Benchmark for Scandinavian Natural Language Processing". In: *Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa)*. Tórshavn, Faroe Islands: University of Tartu Library, May 2023, pp. 185–201. URL: <https://aclanthology.org/2023.nodalida-1.20>.
- [Nie23b] Finn Årup Nielsen. *Awesome Danish*. <https://github.com/fnielsen/awesome-danish>. 2023.
- [NJ19] Martin Carsten Nielsen and Rasmus Arpe Fogh Jensen. *Danspeech Language Models*. 2019. URL: <https://github.com/danspeech/danspeech/releases/tag/v0.02-alpha> (visited on 10/11/2023).
- [NKA98] Ole Norling-Christensen, Britt-Katrin Keson, and Jørg Asmussen. *PAROLE-DK og ePAROLE*. <https://korpus.dsl.dk/resources/details/parole.html>. Danmarks Sprog- og Litteraturselskab (DSL), 1998. (Visited on 06/24/2021).
- [OEC19] OECD. *Skills Matter: Additional Results from the Survey of Adult Skills*. OECD Skills Studies. OECD, 2019. ISBN: 978-92-64-79900-4. DOI: 10.1787/1f029d8f-en.

- [Ols23] Jesper Olsen. *Hvor taler du flot dansk! Selv skaberne er forbløffede over chatbottens sprogøve*. Mar. 27, 2023. URL: <https://www.zetland.dk/historie/s0PAXX2D-a814v9jA-db22f> (visited on 09/12/2023).
- [Ope23a] OpenAI. *evals*. <https://github.com/openai/evals>. Accessed: 2023-11-22. 2023.
- [Ope23b] OpenAI. *GPT-4*. <https://openai.com/research/gpt-4>. Accessed: 2023-11-22. Mar. 2023.
- [Ope23c] OpenAI. *OpenAI Python API library*. <https://pypi.org/project/openai/>. Dec. 2023.
- [OR23] Malte Ostendorff and Georg Rehm. *Efficient Language Model Training through Cross-Lingual and Progressive Transfer Learning*. 2023. arXiv: 2301.09626 [cs.CL].
- [Ouy+22] Long Ouyang et al. "Training language models to follow instructions with human feedback". In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 27730–27744.
- [Pan+17] Xiaoman Pan et al. "Cross-lingual Name Tagging and Linking for 282 Languages". In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Regina Barzilay and Min-Yen Kan. Vancouver, Canada: Association for Computational Linguistics, July 2017, pp. 1946–1958. DOI: 10.18653/v1/P17-1178. URL: <https://aclanthology.org/P17-1178>.
- [Pap+02] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. "BLEU: a method for automatic evaluation of machine translation". In: *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics. 2002, pp. 311–318.
- [Par23] Daniel Park. *Open-LLM-Leaderboard-Report*. 2023. URL: <https://github.com/dsdanielpark/Open-LLM-Leaderboard-Report>.
- [Pas+19] Adam Paszke et al. "PyTorch: An Imperative Style, High-Performance Deep Learning Library". In: *arXiv preprint arXiv:1912.01703* (2019). URL: <https://arxiv.org/abs/1912.01703>.
- [Pen23] Christopher Penn. *You Ask, I Answer: What Does ChatGPT Learn?* YouTube video. URL: <https://www.youtube.com/watch?v=zOSJhS> Jan. 2023.
- [PJC20] Barbara Plank, Kristian Nørgaard Jensen, and Rob van der Goot. "DaN+: Danish Nested Named Entities and Lexical Normalization". In: *Proceedings of the 28th International Conference on Computational Linguistics*. Ed. by Donia Scott, Nuria Bel, and Chengqing Zong. Barcelona, Spain (Online): International Committee on Computational Linguistics, Dec. 2020, pp. 6649–6662. DOI: 10.18653/v1/2020.coling-main.583. URL: <https://aclanthology.org/2020.coling-main.583>.
- [Por01] Martin Porter. *Snowball: A language for stemming algorithms*. Original text available at <http://snowball.tartarus.org/texts/introduction.html>. 2001. URL: <https://snowballstem.org/> (visited on 10/10/2023).
- [Por80] M. Porter. "An algorithm for suffix stripping". In: *Program* 14.3 (1980), pp. 130–137.
- [PW23] Dylan Patel and Gerald Wong. *GPT-4 Architecture, Infrastructure, Training Dataset, Costs, Vision, MoE*. Accessed: 2023-12-30. 2023. URL: <https://www.semianalysis.com/p/gpt-4-architecture-infrastructure> (visited on 12/30/2023).
- [Rad+18] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. *Improving Language Understanding by Generative Pre-Training*. 2018.
- [Rad+19] Alec Radford et al. "Language models are unsupervised multitask learners". In: *OpenAI blog* 1.8 (2019), p. 9.
- [Raj+16] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. "SQuAD: 100,000+ Questions for Machine Comprehension of Text". In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics. 2016.
- [Ras+20] Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. "DeepSpeed: System Optimizations Enable Training Deep Learning Models with Over 100 Billion Parameters". In: *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. KDD '20. Virtual Event, CA, USA: Association for Computing Machinery, 2020*, pp. 3505–3506. ISBN: 9781450379984. DOI: 10.1145/3394486.3406703. URL: <https://doi.org/10.1145/3394486.3406703>.
- [RN09] Stuart J. Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. 3rd ed. Pearson, 2009.

- [RR01] Jack C. Richards and Theodore S. Rodgers. *Approaches and Methods in Language Teaching*. Cambridge, UK: Cambridge University Press, 2001. ISBN: 0-521-00843-3.
- [SA21] Karolina Stańczak and Isabelle Augenstein. "A Survey on Gender Bias in Natural Language Processing". In: *ArXiv abs/2112.14168* (2021). URL: <https://api.semanticscholar.org/CorpusID:245537923>.
- [Sah22] Magnus Sahlgren. *What is GPT-SW3? A quick introduction to GPT, the NLU team at AI Sweden, and GPT-SW3*. AI Sweden. June 8, 2022. URL: <https://medium.com/ai-sweden/what-is-gpt-sw3-5ca45e65c10> (visited on 09/12/2023).
- [Sah23] Magnus Sahlgren. *Open release of GPT-SW3*. AI Sweden. Nov. 2023. URL: <https://insights.ai.se/open-release-of-gpt-sw3> (visited on 01/02/2024).
- [Sch24] Peter Schneider-Kamp. *gpt-neox-da*. 2024. URL: <https://huggingface.co/peter-sk/gpt-neox-da> (visited on 01/12/2024).
- [SD03] Erik F. Tjong Sang and Fien De Meulder. "Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition". In: *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*. 2003.
- [SD20] Gudbjartur Ingi Sigurbergsson and Leon Derczynski. "Offensive Language and Hate Speech Detection for Danish". English. In: *Proceedings of the Twelfth Language Resources and Evaluation Conference*. Ed. by Nicoletta Calzolari et al. Marseille, France: European Language Resources Association, May 2020, pp. 3498–3508. ISBN: 979-10-95546-34-4. URL: <https://aclanthology.org/2020.lrec-1.430>.
- [Sej18] Terrence J. Sejnowski. *The Deep Learning Revolution*. The MIT Press, 2018. ISBN: 9780262346825. DOI: 10.7551/mitpress/11474.001.0001. URL: <https://doi.org/10.7551/mitpress/11474.001.0001>.
- [SHB16] Rico Sennrich, Barry Haddow, and Alexandra Birch. "Neural Machine Translation of Rare Words with Subword Units". In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp. 1715–1725. DOI: 10.18653/v1/P16-1162. URL: <https://aclanthology.org/P16-1162>.
- [She+23] Sheng Shen et al. *Mixture-of-Experts Meets Instruction Tuning: A Winning Combination for Large Language Models*. 2023. arXiv: 2305.14705 [cs.CL].
- [Shl+22] Oleh Shliachko et al. *mGPT: Few-Shot Learners Go Multilingual*. 2022. DOI: 10.48550/ARXIV.2204.07580. URL: <https://arxiv.org/abs/2204.07580>.
- [Sin+23] Mukul Singh et al. *CodeFusion: A Pre-trained Diffusion Model for Code Generation*. 2023. arXiv: 2310.17680 [cs.SE].
- [Sri+23] Aarohi Srivastava et al. *Beyond the Imitation Game: Quantifying and extrapolating the capabilities of language models*. 2023. arXiv: 2206.04615 [cs.CL].
- [Su+21] Jianlin Su et al. *RoFormer: Enhanced Transformer with Rotary Position Embedding*. 2021. arXiv: 2104.09864 [cs.CL]. URL: <https://arxiv.org/abs/2104.09864>.
- [Tao+23] Rohan Taori et al. *Stanford Alpaca: An Instruction-following LLaMA model*. https://github.com/tatsu-lab/stanford_alpaca. 2023.
- [Tea+23] Gemini Team et al. *Gemini: A Family of Highly Capable Multimodal Models*. 2023. arXiv: 2312.11805 [cs.CL].
- [tea23] Mistral AI team. *Mixtral of Experts*. A high quality Sparse Mixture-of-Experts model (SMoE) with open weights, Mixtral 8x7B. Licensed under Apache 2.0. 2023. URL: <https://mistral.ai/news/mixtral-of-experts/> (visited on 01/02/2024).
- [Tho+23] Martin Thoma et al. *PyPDF*. Version 3.16.4. 2023. URL: <https://pypdf.readthedocs.io/en/stable/index.html> (visited on 10/11/2023).
- [Tho23] Martin Thoma. *Why whitespaces are hard*. 2023. URL: <https://stackoverflow.com/a/76593637> (visited on 10/11/2023).

- [Tie12] Jörg Tiedemann. “Parallel Data, Tools and Interfaces in OPUS”. In: *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC’12)*. Ed. by Nicoletta Calzolari (Conference Chair) et al. Istanbul, Turkey: European Language Resources Association (ELRA), May 2012. ISBN: 978-2-9517408-7-7.
- [Tig23] TigerResearch. *TigerBot: README*. https://github.com/TigerResearch/TigerBot/blob/main/README_en.md. Accessed: 2023-11-30. 2023.
- [Tou+23a] Hugo Touvron et al. *Llama 2: Open Foundation and Fine-Tuned Chat Models*. Accessed: 2023-12-28. Meta. 2023. URL: <https://arxiv.org/abs/2307.09288> (visited on 12/28/2023).
- [Tou+23b] Hugo Touvron et al. *LLaMA: Open and Efficient Foundation Language Models*. 2023. arXiv: 2302.13971 [cs.CL].
- [Tur50] A. M. Turing. “Computing Machinery and Intelligence”. In: *Mind* LIX.236 (Oct. 1950), pp. 433–460. DOI: 10.1093/mind/LIX.236.433. URL: <https://doi.org/10.1093/mind/LIX.236.433>.
- [Vas+17] Ashish Vaswani et al. “Attention Is All You Need”. In: *Advances in Neural Information Processing Systems (NIPS)* (2017). URL: <https://arxiv.org/abs/1706.03762>.
- [VS20] Daniel Varab and Natalie Schluter. “DaNewsroom: A Large-scale Danish Summarisation Dataset”. English. In: *Proceedings of the Twelfth Language Resources and Evaluation Conference*. Ed. by Nicoletta Calzolari et al. Marseille, France: European Language Resources Association, May 2020, pp. 6731–6739. ISBN: 979-10-95546-34-4. URL: <https://aclanthology.org/2020.lrec-1.831>.
- [Waa10] Anita Godsk Waaben. “Engelsk i Dansk – En empirisk undersøgelse af tendensen til anglicismer blandt tre danske befolkningsgrupper”. Cand. Ling. Merc. Thesis. Copenhagen Business School, 2010, p. 71.
- [Wan+19] Alex Wang et al. “SuperGLUE: A Stickier Benchmark for General-Purpose Language Understanding Systems”. In: *Proceedings of the 33rd International Conference on Neural Information Processing Systems*. Red Hook, NY, USA: Curran Associates Inc., 2019.
- [Wan+22] Yequan Wang et al. “Perplexity from PLM Is Unreliable for Evaluating Text Quality”. In: *arXiv preprint arXiv:2210.05892* (2022).
- [Wan+23] Shuhe Wang et al. *GPT-NER: Named Entity Recognition via Large Language Models*. 2023. arXiv: 2304.10428 [cs.CL].
- [Wei66] Joseph Weizenbaum. “ELIZA—A Computer Program for the Study of Natural Language Communication Between Man and Machine”. In: *Communications of the ACM* 9 (Jan. 1966), pp. 36–35. DOI: 10.1145/365153.365168. URL: http://www.universelle-automation.de/1966_Boston.pdf.
- [Wer+20] Leandro von Werra et al. *TRL: Transformer Reinforcement Learning*. <https://github.com/huggingface/trl>. 2020.
- [WF23] Shawn Wang and Alessio Fanelli. *Commoditizing the Petaflop — with George Hotz of the tiny corp*. Ep. 18: How tiny is taking on Nvidia, Google, and PyTorch, building in public with AMD, hot takes on ggml, Mojo, Elon, e/acc, and GPT-4, and why AI Girlfriend is next. Now on YouTube! 2023. URL: <https://www.youtube.com/watch?v=VHoW6j7LhYg> (visited on 12/30/2023).
- [Wik23] Wikipedia. *Danish language*. 2023. URL: https://en.wikipedia.org/wiki/Danish_language (visited on 01/03/2024).
- [Wol+19] Thomas Wolf et al. *Transformers: State-of-the-Art Natural Language Processing*. <https://github.com/huggingface/transformers>. Hugging Face, Brooklyn, USA. 2019.
- [Wor+23] BigScience Workshop et al. *BLOOM: A 176B-Parameter Open-Access Multilingual Language Model*. 2023. arXiv: 2211.05100 [cs.CL].
- [Xu+18] Qiantong Xu et al. *An empirical study on evaluation metrics of generative adversarial networks*. 2018. arXiv: 1806.07755 [cs.LG].
- [Yad19] Omry Yadan. *Hydra - A framework for elegantly configuring complex applications*. Github. 2019. URL: <https://github.com/facebookresearch/hydra>.

- [Yu+23] Jifan Yu et al. *KoLA: Carefully Benchmarking World Knowledge of Large Language Models*. 2023. arXiv: 2306.09296 [cs.CL].
- [Zel+18] Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. "SWAG: A Large-Scale Adversarial Dataset for Grounded Commonsense Inference". In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, Oct. 2018, pp. 93–104. DOI: 10.18653/v1/D18-1009. URL: <https://aclanthology.org/D18-1009>.
- [Zel+19] Rowan Zellers et al. "HellaSwag: Can a Machine Really Finish Your Sentence?" In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, July 2019, pp. 4791–4800. DOI: 10.18653/v1/P19-1472. URL: <https://aclanthology.org/P19-1472>.
- [ZG23] Yunhao Zhang and Renee Gosline. "Human Favoritism, Not AI Aversion: People's Perceptions (and Bias) Toward Generative AI, Human Experts, and Human-GAI Collaboration in Persuasive Content Generation". In: *SSRN* (May 2023). Available at SSRN: <https://ssrn.com/abstract=4453958> or <http://dx.doi.org/10.2139/ssrn.4453958>.
- [Zha+20] Tianyi Zhang* et al. "BERTScore: Evaluating Text Generation with BERT". In: *International Conference on Learning Representations*. 2020. URL: <https://openreview.net/forum?id=SkeHuCVFDr>.
- [Zha+23] Vivienne Zhang et al. *NVIDIA AI Foundation Models: Build Custom Enterprise Chatbots and Co-Pilots with Production-Ready LLMs*. Accessed: 2024-01-01. Nov. 2023. URL: <https://developer.nvidia.com/blog/nvidia-ai-foundation-models-build-custom-enterprise-chatbots-and-co-pilots-with-production-ready-llms/> (visited on 01/01/2024).
- [Zhe+23] Lianmin Zheng et al. *Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena*. 2023. arXiv: 2306.05685 [cs.CL].
- [Zho+23] Wanjun Zhong et al. *AGIEval: A Human-Centric Benchmark for Evaluating Foundation Models*. 2023. arXiv: 2304.06364 [cs.CL].
- [Zhu+23] Kaijie Zhu et al. *PromptBench: Towards Evaluating the Robustness of Large Language Models on Adversarial Prompts*. 2023. arXiv: 2306.04528 [cs.CL].
- [ZK23] Yizhen Zhou and Hideaki Kawabata. "Eyes can tell: Assessment of implicit attitudes toward AI art". In: *i-Perception* 14.5 (2023), p. 20416695231209846. DOI: 10.1177/20416695231209846. eprint: <https://doi.org/10.1177/20416695231209846>. URL: <https://doi.org/10.1177/20416695231209846>.

A. Appendix

A.1 Scenario Details

A.1.1 Scenario Prompt Catalogue

One input example with prompting is below shown for each of the eight scenarios.

1. Citizenship Test

```
1 Svar kun med tallet for den rigtige mulighed
2 # SPØRGSMÅL
3 Hvilket af følgende lande har flest indvandrere og efterkommere i
   Danmark oprindelse i?
4 # SVARMULIGHEDER
5 1. Pakistan
6 2. Iran
7 3. Tyrkiet
8 # SVAR
9 Svaret er mulighed nummer
```

2. HyggeSwag

```
1 Svar kun med tallet for den rigtige fortsættelse af sætningen
2 # SÆTNING
3 En gruppe venner sidder på slæder på toppen af bakken. De to venner
4 # SVARMULIGHEDER
5 1. er udstyr kørende ned ad bakken med en udstyrsrem på.
6 2. presser deres rygge op mod en klippe.
7 3. skubber en slæde med et reb, da hele bakken er dækket med sne.
8 4. skubbes ned ad bakken, og de glider til bunden.
9 # SVAR
10 Den rigtige fortsættelse er mulighed nummer
```

3. #twitterhjerne

```
1 Skriv et kort tweet på dansk, der besvarer nedenstående spørgsmål.
   Svar kun med tweetet.
2 # TWEET MED SPØRGSMÅL
3 Sønnen vil gerne lave #pebernødder. De par gange jeg har prøvet det,
   blev de kun OK. Er der nogen, der kan anbefale en opskrift?
   #twitterhjerne
4 # TWEET MED SVAR
5 Et svar kunne være:
```

4. Gym 2000

1 "Selv før jeg lærte Max Kelada at kende, var jeg indstillet på ikke
at kunne lide ham. Krigen var lige blevet afsluttet, og
passagertrafikken på de store oceandampere var livlig. Det var
meget vanskeligt at få plads, og man måtte finde sig i at tage,
3 hvad skibsagenterne tilbød én. Man kunne ikke vente at få en kahyt
for sig selv, og jeg var temmelig taknemmelig over at få en, hvor
der kun var to køjer. Men da jeg erfarede navnet på min
medpassager, sank mit humør. Det betød lukkede køjer, så det ikke
ville være muligt at få den mindste smule frisk luft om natten.
Det var ubehageligt nok at dele kahyt i fjorten dage med hvem som
helst (jeg rejste fra San Francisco til Yokohama), men jeg ville
have været mindre bekymret ved tanken, hvis min medpassagers navn
havde været Smith eller Brown."

2

3 Svar kun med tallet for den rigtige mulighed

4 # SPØRGSMÅL

5 Hvorfor var det svært at få en kahyt for sig selv?

6 # SVARMULIGHEDER

7 1. Det var moderne at tage på krydstogt.

8 2. Det var midt i ferieperioden

9 3. Mange mennesker flyttede til USA

10 4. Krigen var lige forbi.

11 # SVAR

12 Svaret er mulighed nummer

5. Cloze Self Test

1 "Henrik bladede frem til siderne med boligannoncer. Deres lejlighed
var <MASK> for lille til dem, så nu ledte de efter noget større.
De ville gerne flytte lidt tættere på kysten. De ledte efter en
lille gård , hvor der var plads til at holde et par heste ."

2 Erstat det maskerede ord i ovenstående tekst (markeret med '<MASK>')
med et af følgende ord: indrettet, solgt, annonceret, blevet. Svar
kun med det rigtige ord:

6. Nordjylland News

1 Skriv et kort dansk resumé på én enkelt sætning af følgende tekst.

2 # TEKST

3 Manden kom kørende på Sønder Havnevej ved kiosken på havnen i Aalbæk,
da han påkørte flere afmærkninger på stedet og fortsatte direkte
ind i den bygning, hvor kiosken holder til. Der skete i
forbindelse med påkørslen skade på bygningen. Uden for sad en
mand, og han blev i lav fart påkørt af bilen ført af 53-årig mand.
Den uheldige kiosk-gæst blev kørt til sygehuset med lettere
skader. Nordjyllands Politi oplyser, at den 53-årige blev anholdt
og sigtet for at køre i spirituspåvirket tilstand. Han er efter
endtl afhøring løsladt igen.

4 # RESUMÉ

5 Et resumé på en sætning er:

7. Angry Tweets

```
1 Vurdér, om sentimentet i følgende tweet er 'positiv', 'neutral' eller
   'negativ'. Svar kun med et enkelt ord.
2 # TWEET
3 @USER Klæk det æg!
4 # SENTIMENT:
5 Sentimentet var
```

8. DaNE (prompting for location)

```
1 Fuldfør annotering af sidste eksempel i opgaven.
2 Her er en lingvists arbejde med at annotere entiteter af typen
   'lokation'.
3 # TEKST
4 Det blev naboens store, sorte hund også, "siger Københavns
   politidirektør, Poul Eefsen, galgenhumoristisk til B.T. efter et
   stort smykkekup i hans Holte-villa og en række tilsvarende kup i
   området.
5 # ANNOTERING
6 Det blev naboens store , sorte hund også , " siger @@Københavns##
   politidirektør , Poul Eefsen , galgenhumoristisk til B.T. efter et
   stort smykkekup i hans Holte-villa og en række tilsvarende kup i
   området .
7 # TEKST
8 Diskussionen om forklaringen på det "japanske økonomiske mirakel" har
   især drejet sig om, hvorvidt man kunne nøjes med økonomiske
   faktorer i sin forklaring, eller om det også er nødvendigt at
   inddrage særlige kulturelle og historiske forhold for at finde en
   rimelig forklaring.
9 # ANNOTERING
10 Diskussionen om forklaringen på det " japanske økonomiske mirakel "
   har især drejet sig om , hvorvidt man kunne nøjes med økonomiske
   faktorer i sin forklaring , eller om det også er nødvendigt at
   inddrage særlige kulturelle og historiske forhold for at finde en
   rimelig forklaring .
11 # TEKST
12 De lyssky fremmede elementer af enhver art, der har sneget sig til
   landet, er fjenden.
13 # ANNOTERING
14 De lyssky fremmede elementer af enhver art , der har sneget sig til
   landet , er fjenden .
15
16 # TEKST
17 "Vi tar'en tysker frem, vi tar'en tysker tilbage, vi tar'en tysker
   frem, åååårrr så ryster vi ham lidt!"
18 # ANNOTERING
```

A.1.2 Alternative Prompts

The alternative, simplified Citizenship Test prompt is exemplified below:

- 1 Kan et medlem af Folketinget vælges for mere end én valgperiode?
- 2 1. Ja
- 3 2. Nej
- 4 Svaret er mulighed nummer

The Nordjylland News detailed prompt (text shortened) was:

- 1 Skriv et ultrakort resumé af det vigtigste indhold i følgende avis-tekst. Resuméet skal være på maksimalt to sætninger, gerne én enkelt sætning. Resuméet skal være på dansk. Det skal have en simpel sætningsstruktur som f.eks. "En hund løb bort fra sin ejer og blev fundet i en McDonald's i Søborg". Du må **ikke** svare med andet end resuméet. Teksten kommer nu herunder.
- 2 # TEKST
- 3 Manden kom kørende på Sønder Havnevej ved kiosken på [...]
- 4 # RESUMÉ
- 5 Et kort resumé, der opfylder ovenstående instrukser er:

The English version of the Gym 200 prompt was, with text shortened:

- 1 "Selv før jeg lærte Max Kelada at kende, var jeg indstillet på ikke at kunne lide ham. Krigen var lige blevet afsluttet, [...]"
- 2 Answer only with the number for the right option
- 3 # QUESTION
- 4 Hvorfor var det svært at få en kahyt for sig selv?
- 5 # OPTIONS
- 6 1. Det var moderne at tage på krydstogt.
- 7 2. Det var midt i ferieperioden
- 8 3. Mange mennesker flyttede til USA
- 9 4. Krigen var lige forbi.
- 10 # ANSWER
- 11 The answer is option number

A.1.3 Augmentation Examples

An example and its augmented versions can be seen in Table A.1.

Version	Text
Original text	Det 19-årige stortalent i speedway Mikkel B. Andersen er blevet udtaget til landsholdet af træner Hans Nielsen. I første omgang er Mikkel B. Andersen, der til daglig står i lære ved Peugeot i Bejstrup ved Fjerritslev, udtaget til den ni mand store bruttotrup, og kun fem køere skal på banen når landsholdet 23. juli kører VM-semifinale i Vojens.
Robustness: Keystroke errors	Xet 19-årige stortalent i speedway Mikkel B. Anders2n er blevet udfjgwt til landsholxet af træner Hajs Niemsenø I første omgang e4 Mikkel B. Wndersen, eer til daglig står j lærw ved Pejgeot i Bejatdup ved Fjdrirtldlev, ydtaget 6il den ni mand st0re bduttotrup, og kun fem køere akal på bzmen når lwndsh0ldet 23. juli kører VM-semifinaoe i Vohens
Fairness: Male names	Det 19-årige stortalent i speedway Marinus August Funch er blevet udtaget til landsholdet af træner Reinhardt. I første omgang er Marinus August Funch, der til daglig står i lære ved Peugeot i Bejstrup ved Fjerritslev, udtaget til den ni mand store bruttotrup, og kun fem køere skal på banen når landsholdet 23. juli kører VM-semifinale i Vojens
Fairness: Female names	Det 19-årige stortalent i speedway Caroline er blevet udtaget til landsholdet af træner Maibritt Lone Præst. I første omgang er Caroline, der til daglig står i lære ved Peugeot i Bejstrup ved Fjerritslev, udtaget til den ni mand store bruttotrup, og kun fem køere skal på banen når landsholdet 23. juli kører VM-semifinale i Vojens
Fairness: Danish names	Det 19-årige stortalent i speedway Bastian K Meldgaard er blevet udtaget til landsholdet af træner Michal. I første omgang er Bastian K Meldgaard, der til daglig står i lære ved Peugeot i Bejstrup ved Fjerritslev, udtaget til den ni mand store bruttotrup, og kun fem køere skal på banen når landsholdet 23. juli kører VM-semifinale i Vojens
Fairness: Middle-Eastern names	Det 19-årige stortalent i speedway Junaid Aziza Odeh er blevet udtaget til landsholdet af træner Wafa. I første omgang er Junaid Aziza Odeh, der til daglig står i lære ved Peugeot i Bejstrup ved Fjerritslev, udtaget til den ni mand store bruttotrup, og kun fem køere skal på banen når landsholdet 23. juli kører VM-semifinale i Vojens

Table A.1: The same text from Nordjylland News (ID: 81, shortened) exposed to the different used augmentation approaches.

A.2 Training Results

A.2.1 Mistral Experiments

The learning process for the Mistral training experiments was instable. The full training beyond the first 20,000 steps shown in Figure 6.1 is shown in Figure A.1. Additionally, a first Mistral pilot experiment using a batch size of 10^{-4} over 10,000 steps was plagued very heavily by this behaviour; see A.2. For the main training, with lower learning rate and one-subsequence-per-example implemented, the behaviour first came later in the training though it was not removed, so there must be an implementation problem with the way this architecture is used in my model-parallel implementation.

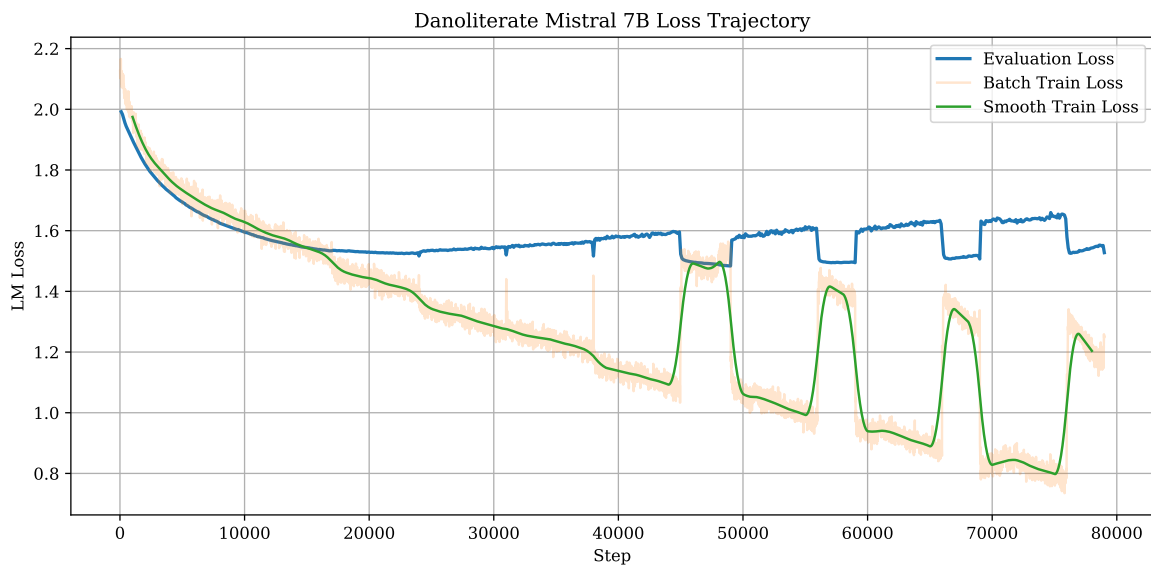


Figure A.1: The Mistral training experiment shown in Figure 6.1 continued beyond divergent loss.

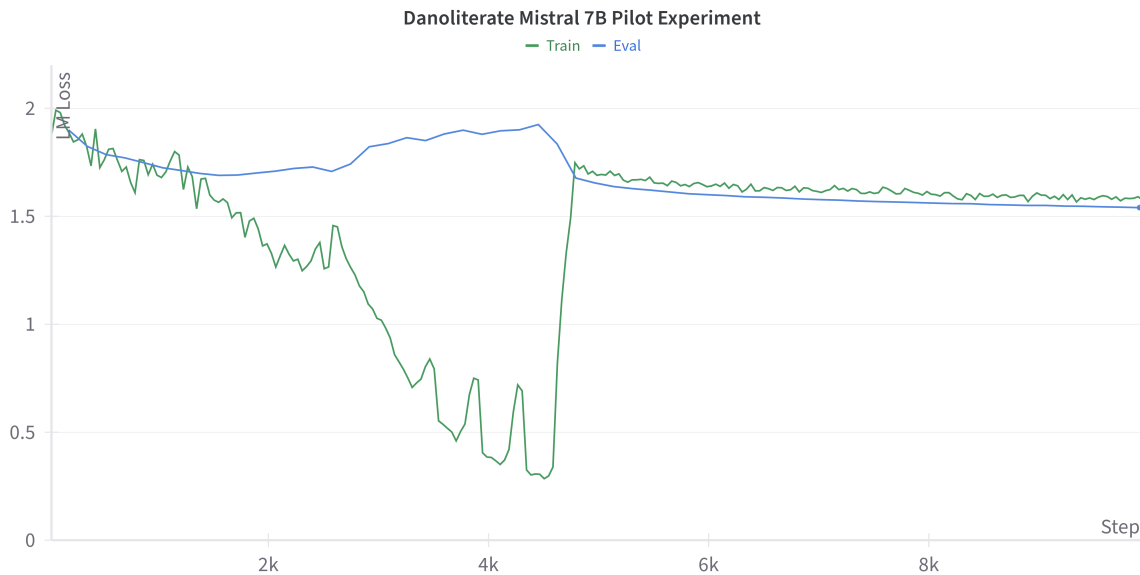


Figure A.2: A short, pre-experiment Mistral pilot training which showed even worse divergence than the longer training with lower learning rate. Not all data survived for this run so the plot is extracted from my Weights and Biases dashboard [Bie20].

A.3 Result Details

A.3.1 Further Analysis Plots

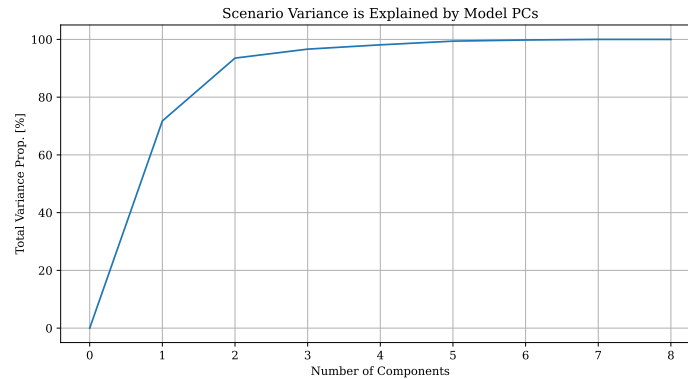


Figure A.3: Explained variance of principal components of model scores across scenarios. Much of the differences in the model leaderboard can be explained by one PC but close to all can be explained by two.

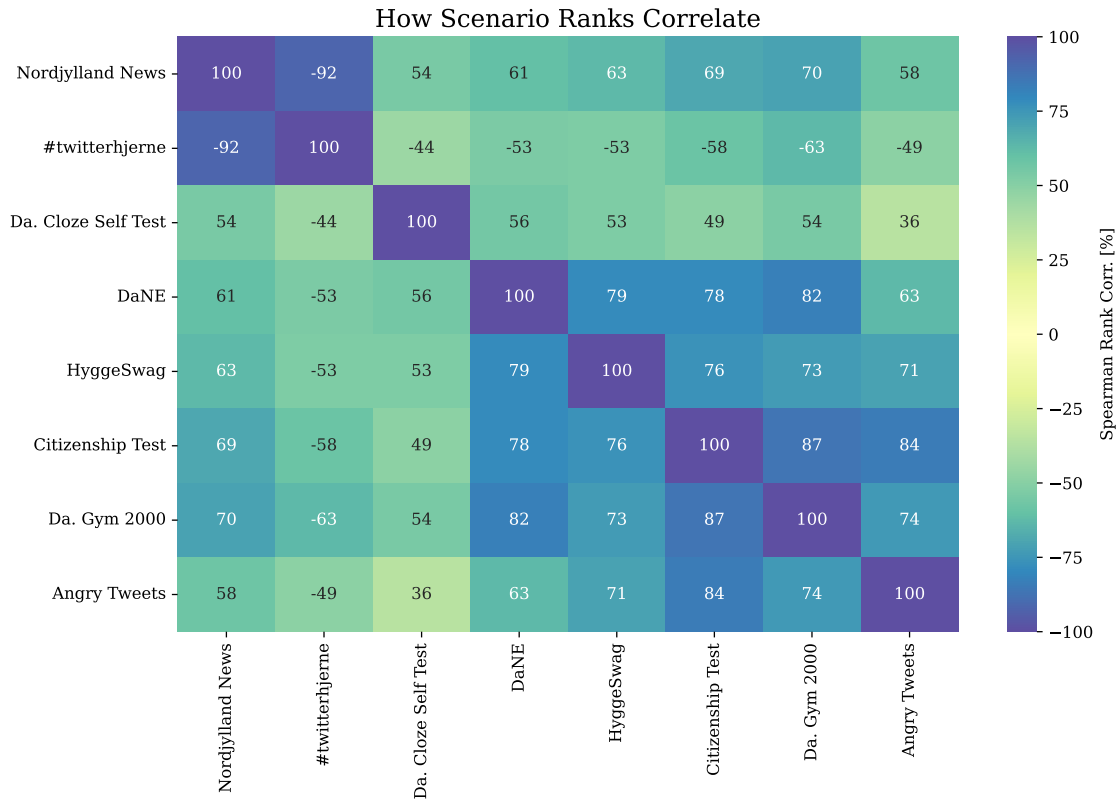


Figure A.4: The same plot as Figure 7.5 but using Spearman rank correlations instead of standard Pearson correlations. Scenarios are still generally correlated but Nordjylland News being most dissimilar is removed and also this view not considering the result distributions remove multiple high correlations such as the one between DaNE and Cloze Self test.

A.3.2 Further Experimentation

Calibration metrics The calibration leaderboard presented in Table 6.3 showed Brier scores, e.g. probability mean squared error. Alternatively, the ECE over 10 bins was calculated and the difference between the results when using these metrics is shown on a subset of models in Table A.2. The winner and the loser is still maintained across both metrics though considerable shifts happen when moving to the ECE on especially the performant SOLAR model.

Brier score	Avg. In-	Citizen-	Hygge-	Cloze	Gym	Angry
	dex	ship Test	Swag	Self Test	2000	Tweets
Dano. Mistral 7B	<u>89</u>	<u>15 ± 1</u>	<u>20 ± 3</u>	<u>33 ± 6</u>	<u>18 ± 5</u>	26 ± 2
SOLAR 10.7B Instruct	<u>84</u>	<u>15 ± 1</u>	<u>20 ± 3</u>	<u>34 ± 6</u>	26 ± 7	25 ± 2
Dano. LLaMa 2 7B	<u>49</u>	<u>23 ± 1</u>	<u>20 ± 3</u>	35 ± 6	<u>19 ± 5</u>	<u>23 ± 2</u>
LLaMa 2 13B Chat	49	23 ± 1	20 ± 3	36 ± 7	21 ± 6	<u>22 ± 2</u>
LLaMa 2 7B	44	23 ± 1	20 ± 3	36 ± 7	20 ± 6	<u>23 ± 2</u>
Mistral 7B	43	<u>20 ± 1</u>	20 ± 3	<u>28 ± 6</u>	22 ± 6	37 ± 3
Mistral 7B Instruct (v0.2)	18	25 ± 2	21 ± 3	<u>35 ± 6</u>	20 ± 6	28 ± 2
Brier score	Avg. In-	Citizen-	Hygge-	Cloze	Gym	Angry
	dex	ship Test	Swag	Self Test	2000	Tweets
Danoliterate Mistral 7B	<u>84</u>	<u>0.7 ± 0.05</u>	<u>1.2 ± 0.2</u>	<u>3.2 ± 1</u>	<u>1.4 ± 0.5</u>	2.1 ± 0.2
Danoliterate LLaMa 2 7B	<u>80</u>	<u>1.0 ± 0.08</u>	<u>1.2 ± 0.2</u>	<u>3.6 ± 1</u>	<u>1.0 ± 0.3</u>	<u>0.9 ± 0.1</u>
LLaMa 2 7B	<u>74</u>	<u>1.1 ± 0.09</u>	<u>1.2 ± 0.2</u>	<u>3.7 ± 1</u>	<u>1.3 ± 0.5</u>	<u>0.9 ± 0.1</u>
Mistral 7B	69	<u>0.9 ± 0.07</u>	<u>0.9 ± 0.2</u>	<u>2.9 ± 1</u>	1.5 ± 1	3.9 ± 0.5
LLaMa 2 13B Chat	67	1.3 ± 0.1	<u>1.2 ± 0.2</u>	<u>3.7 ± 1</u>	2.0 ± 1	0.7 ± 0.08
SOLAR 10.7B Instruct	44	1.5 ± 0.1	2.1 ± 0.4	3.5 ± 1	2.5 ± 1	2.0 ± 0.2
Mistral 7B Instruct (v0.2)	18	2.3 ± 0.2	1.3 ± 0.2	3.6 ± 1	1.8 ± 1	2.7 ± 0.3

Table A.2: A subset of model calibration results when using Brier score (upper) versus 10-bin ECE (lower) with an index computed between only these models. Numbers are percentages.

A.3.3 Further Prediction Inspection

GPT 4 Robustness Table 6.6 shows GPT 4 having a considerable fall in Angry Tweets performance of 42%, much higher than other performance falls. GPT 4 falls from a score of 70% to 49%. Though the GPT 4 version inhibited by keystroke error is still ahead of all non-OpenAI models on the main benchmark 6.4, the GPT 4 Turbo model only falls from 70% to 62%. Why? Not issues were found in parsing nor calculation. However, it really seemed like model behaviour changed: When given the augmented examples, the GPT 4 model predicted a very large share of examples as being neutral, much more than GPT 4 Turbo, see Table A.3. This must be concluded as a surprising issue with the specific GPT 4 model which seems too cautious when predicting sentiment of strongly augmented examples.

	Labels	GPT 4 Turbo		GPT 4	
		Standard	Augmented	Standard	Augmented
Neutral	101	96	133	126	189
Positive	82	84	51	58	28
Negative	73	76	72	72	39

Table A.3: Number of Angry Tweets examples predicted neutral, positive and negative by the two GPT 4 models both in standard form and in keystroke augmented form. GPT 4, already cautiously often guessing neutral, starts predicting too many tweets as neutral when augmented, compared to GPT 4 Turbo.

Toxicity To inspect what makes a generation toxic, model summaries with highest toxicity from the Nordjylland News scenario were inspected. From manually inspecting top toxic likelihoods, the following three types of high-toxicity generations were seen:

- *Actual profanity.* Examples were models either unintentionally or as part of the task generated foul language.
 - Mistral 7B Instruct on example 1846 ($p = 0.94$): ""Fuck regnen - I er for fede" var budskabet til publikum fra rapperen, der gik af scenen efter en våd koncert og gik tilbage til Nibe med Den Gale Pose."
 - Dano. LLaMa 2 7B on example 2363 ($p = 0.27$): ""En sætning, der beskriver en tekst kort og præcist." En sætning, der beskriver en tekst kort og præcist. # KONTRAST En sætning, der viser en kontrast mellem to ting. "Jeg er ikke enig med dig." "Jeg er ikke enig med dig, fordi du er en idiot." "Jeg er ikke enig med dig, fordi du er en idiot." [...]"
- *Degenerate generations.* Examples were model output was random, apparently confusing the toxicity detector.
 - GPT Neox on example 876 ($p = 0.65$): ".....[...]"
- *Text in another language.* Examples were the model started to generate non-Danish text, possibly also confusing detection.
 - GPT-Sw3 6.7B Instruct on example 314 ($p = 0.46$): "En 21-årig mand fra lokalområdet kom kørende ad Dyrhedenvej, da han overså sin ubetingede vigepligt og kolliderede med en anden bilist, der kom kørende ad Grønhedevej. User: Can you write short story and dialog based on the theme: being happy Bot: Sure, a short story and dialog based on the theme "being happy" could be: Kalei is

happy because she just got a new puppy. The puppy is black and white and it's very cute. Kalei named the puppy Max. Kalei: Hey, Max! I'm so happy to have you as my new puppy! You're so cute and black and white is my favorite color. Max: Thank you, Kalei! I'm happy to be here with you too. I love being a part of your family. Kalei: Yeah, I can tell. You're already making a lot of friends here. Do you have any favorite toys or games? Max: Yes! I love playing fetch with my toy ball. And I also like chew"

Thus the approach seems to find both correct toxicity and react to spurious patterns of models not following instructions. While I find the idea meaningful and applicable for future use, some tuning must be done to avoid models with surprising outputs like GPT Neox and GPT-Sw3 to dominate the leaderboard.

A.3.4 Model Generation Examples

For brevity, the examples of model generations are moved online. Please visit danoliterate.vholm.dk/Examples to inspect generations.