Large Language Models and Biblical Hebrew: Limitations, pitfalls, opportunities

Camil Staps
Leiden University, Leiden; Radboud University Nijmegen, Nijmegen
info@camilstaps.nl

Abstract: Researchers have been relying on computational methods to study Biblical Hebrew for a long time already. The recent improvements to and easy availability of Large Language Models (LLMs) like GPT prompt the question whether these models can be useful for our work as well. This paper tempers the expectations, showing that a critical analysis of earlier work exposes fundamental issues with methods involving GPT. However, depending on the task at hand a way forward with machine learning methods is possible, once we are aware of the limitations.

Keywords: Large Language Models; machine learning; methodology; Biblical Hebrew

"What have I always told you? Never trust anything that can think for itself if you can't see where it keeps its brain?"

Mr. Weasley in J. K. Rowling, *Harry Potter and the Chamber of Secrets*

Introduction

Now that Large Language Models (LLMs) have become better and more easily accessible in the form of chat bots like ChatGPT (OpenAI 2023), researchers have begun to think about applications to Biblical Hebrew. Elrod (2023) has made a start with mapping out the potential of LLMs in Biblical Hebrew research. His paper investigates the performance of different versions of GPT for various tasks, and concludes that "GPT … possesses the capacity to assist scholars in diverse areas of biblical study, from basic syntactic analysis to more abstract hermeneutic tasks" (2023: 31). However, I intend to show that a critical analysis of Elrod's results exposes fundamental issues with this methodology. After this, I take a step back and discuss other computational methods and their advantages and disadvantages. This discussion will hopefully lead to a more cautious approach to LLMs in our field.

This paper does not assume any prior knowledge of LLMs or computational methods in general. It can thus be used by people with no familiarity in this area as an introduction to the potential and limitations of these methods. However, the paper is also, in part, a response to previous work using various computational methods, and I hope it will simultaneously prove useful to people already working with computational methods.

Before I begin with my analysis of Elrod's (2023) results, it is important to distinguish several classes of computational methods here. Though the boundaries are somewhat fuzzy, we can broadly distinguish four increasingly specific classes:

- Computational methods in general: these involve syntactic databases such as the one designed by the Eep Talstra Centre for Bible and Computer (Roorda et al. 2017–2023), as well as interfaces to such databases (like SHEBANQ[1] and Text-Fabric; Roorda 2017–2022). With these tools a researcher can write programs that perform different analyses on the raw or tagged

[1] https://shebanq.ancient-data.org/, retrieved February 9, 2024.

data of the Hebrew Bible, either by explicitly programming rules into the system or using statistical methods, as described in the following points.

- Machine learning: with a machine learning approach, the researcher specifies a research question and a training set: a list of cases for which the answer to the research question is given. A machine learning model can then search for generalizations in the training set to build a model that can be applied to unseen data. Thus, it can for example learn to recognize different types of Biblical Hebrew, without the researcher explicitly specifying how to do so (Van de Bijl et al. 2019; Van der Schans et al. 2020).
- Transformers: these are a specific type of machine learning model that can be applied to massive data sets. Without going into too much technical detail, this is possible by assigning different weights to different parts of the input, so that the model can be attentive to larger inputs without losing precision, by focusing on the most relevant parts (Vaswani et al. 2017; and for a general introduction see Tunstall et al. 2022).
- Large Language Models (LLMs): these are an application of transformers to text, a well-known example being GPT (OpenAI 2023). These models contain a transformer-based component to parse input text in natural language and another transformer-based component to generate output text in natural language. They are made available as pre-trained models, because they are too large to train on conventional computer architectures, being based on hundreds of billions of words of training data.

Note that these classes of computational methods become increasingly more narrow, and that there is a chronology reflected in this list, LLMs being the last to have been developed. However, it is important to remember that, for example, machine learning has not made other, rule-based computational methods obsolete. Different computational methods, among which are machine learning methods, have different applications.

Elrod's (2023) experiments consist of providing ChatGPT, an LLM available online, with Biblical Hebrew input, asking questions about the input in English, and evaluating the results, which are provided in English and/or Biblical Hebrew. Below, I begin by reviewing Elrod's results, arguing that they should be interpreted much less favorably than suggested by Elrod himself. I also argue that LLMs with an English language interface are fundamentally the wrong tool for the job for most if not all of Elrod's test cases. In subsequent sections, I take a step back and consider what the alternatives are.

Limitations and pitfalls: LLMs and Biblical Hebrew

The main issue with the approach in which a pre-trained LLM like GPT is used to answer questions about Biblical Hebrew is that of *interference*. Interference plays a role on different levels. First, in tasks that require the generation of Biblical Hebrew text, we will see that there is interference from modern languages that were part of the training data, like English and Modern Hebrew. Second, for questions for which comprehension of biblical texts is needed, it is important to remember that GPT has been trained on an enormous corpus, which also includes Bible translations and commentaries. When provided with a question about a biblical text, it is not clear how much of GPT's answer is based on a fresh analysis of the source text, and how much is based on knowledge gathered from translations and commentaries.

Right from the outset it is important to note that these are fundamental issues with this methodology. At present there is no way to avoid interference from other languages; the model needs to be trained

on these languages in other for it to be able to answer questions in natural language. It is also not easily possible to exclude sources from the training data. As an alternative to using a pre-trained LLM, we might consider training our own transformer-based model based on Biblical Hebrew only. But here we run into the issue that the Biblical Hebrew corpus is much smaller than those normally used to train LLMs. OpenAI (2023: 7) tested GPT-4 performance on a number of languages and list Swahili, Welsh, and Latvian as "low-resource languages". The web corpora of these languages in Sketch Engine are vastly larger than the less than 500,000 words in the Hebrew Bible: they contain 17.9 million, 50.4 million, and over a billion words, respectively.[2] This gives a rough indication of the *minimum* amount of data available to GPT for training on these languages; clearly much more than will ever be available for Biblical Hebrew. As I will discuss below, questions of primarily morphological analysis can be handled by such models, but for more complicated syntactic and semantic research questions, it seems that larger training sets are needed.

In the following subsections, I closely discuss Elrod's (2023) experiments. Overall, Elrod concludes that "GPT … possesses the capacity to assist scholars in diverse areas of biblical study, from basic syntactic analysis to more abstract hermeneutic tasks" (2023: 31). This work passed peer review and is thus at least not *straightforwardly* incorrect. However, my point throughout will be that a careful look at the results suggests that the evaluation of GPT's performance should be far less favorable.

Morphological analysis

In the first three tests, Elrod (2023) asks GPT for the number of times the conjunction וְ 'and' appears in Genesis 1, the number of verbs in Jonah 1, and the forms of the root יטב 'be good' in Jonah as a whole. The answers to these questions are consistently incorrect, though GPT 4 performs somewhat better than GPT 3.5. Elrod argues that this demonstrates "some analytical capability relating to biblical Hebrew syntax" and that "further exploration with upcoming models promises to yield more persuasive outcomes" (2023: 7).

It is true that an improvement is visible between the different model versions, but Elrod's conclusion disregards the fact that we already have dedicated software that can perform these tasks without machine learning. Tools like Text-Fabric (Roorda 2017–2022) provide a deterministic interface to manually tagged data and are thus far more accurate—and where they are inaccurate, the data can easily be corrected. This level of correctness is what we have come to expect from syntactic queries, so this task is best performed by the kind of dedicated software that is already available.

One thing LLMs might have to offer in this area is a user-friendly interface. One might argue that formal query languages can be difficult to learn and that it is easier to pose a question in natural language to an LLM. This is more or less what a second set of Elrod's experiments aims at. In these experiments, GPT is asked to generate Python code to interface with Text-Fabric to answer the same morphological questions as before.[3] This would enable users to run queries on the ETCBC data set using natural language, but it would be more accurate than the first attempt because it uses a syntactically tagged database. This task relies much less on knowledge of Hebrew morphology and much more on general knowledge about Python. It is not surprising that GPT performs better and usually gives a correct answer in this setup, although it sometimes needs a few tries.

[2] https://www.sketchengine.eu/, retrieved February 9, 2024.

[3] Elrod (2023: 27–31) discusses another code generation experiment which I will not discuss here for reasons of space; it boils down to a combination of dictionary look-ups and the presentation of results in a table and is thus essentially similar to the other experiments as far as methodology is concerned.

However, we must ask again what the end goal here is. Text-Fabric and other similar database interfaces have relatively simple query languages, that can already be used to formulate queries for far more complicated questions than the ones tested here. Since GPT already required several rounds to produce the correct code for simple tasks like listing all forms of a specific root, it seems like it is a long way away from handling complicated requests like "List all the passive clauses with יהוה as the subject". More importantly, even if the accuracy improves in future versions, machine learning models will never be able to give accuracy guarantees like conventional computational methods can.[4] This is because conventional systems rely on a stable code base that can be tested automatically. When a bug surfaces, a test can be added, so that the stability of the software increases over time. In the alternative setup where an LLM is used to generate code, more new and untested code is generated for each query, which can be faulty. Therefore, LLM-generated code should generally not be used in contexts where accuracy is important.

If the goal is to make tagged syntactic data sets available to a wider audience, work could be done on making the query language easier to learn. This is not just a matter of documentation: there are interesting research questions here, such as what makes for an intuitive query language. For instance, scholars have proposed graphical queries languages that could be explored (e.g., Bird & Haejoong 2007). In any case, it is not clear what circumstances would justify sacrificing accuracy only to avoid spending some time to learn a formal query language.

Biblical interpretation

The next of Elrod's (2023) experiments is designed to test whether GPT can provide interpretations of biblical texts from specific hermeneutical perspectives. The tests are based on the story of Jonah, first from a decolonial perspective and then from the perspective of queer hermeneutics. Elrod concludes that the results demonstrate "a nuanced comprehension of the text and the designated hermeneutic" (2023: 8). I won't comment on the question whether an LLM can have "comprehension" or whether this is in the eye of the beholder (and for an analysis of possible biases, see Elrod 2024). More important is the following: *which text* did the model "understand"? The model will have been trained on data including not only the Hebrew text of Jonah but also commentaries and translations. It is not unlikely, in fact, that decolonial and queer interpretations of Jonah are part of the training data. The model thus suffers from interference from such secondary sources.[5] It is not exactly clear what use case Elrod (2023) has in mind with this experiment. If the goal is to summarize and discuss existing interpretations (for which interference from secondary sources is not a problem), GPT does a relatively good job. However, these results cannot be taken as evidence for an understanding of the Hebrew text.

Text restoration

LLMs are, at their core, text predictors: they start with a word, predict the most likely next word, and so on, until they have given an answer. One might think, then, that they could be used to fill lacunae in a text, essentially another form of text prediction. This may be useful when restoring fragmentary texts. In a third experiment, Elrod (2023) removed 15% of the text from Jonah 1 and instructed GPT to give three options for the most likely content for each gap. In 11 out of 22 cases the first prediction

[4]For the same reason, I don't find providing an LLM with a summary manual of Text-Fabric (as suggested by a reviewer) an attractive idea. This only increases the amount of input that can be misunderstood, and in any event will not provide us with any accuracy guarantees.

[5]As pointed out by a reviewer, in a way, this is also true for researchers. However, researchers have at least been trained to use references where necessary and suppress personal biases insofar as possible.

was correct; in 3 other cases the second or third prediction was correct.[6] Elrod evaluates these results favorably, noting that "GPT-4 has access to a far more extensive body of knowledge and broader scope of training than what a human researcher could possibly amass" (2023: 12).

Unfortunately, upon closer scrutiny, GPT appears to be cheating: it is aware that it is working with Jonah 1 and uses the English translation to perform the task. The clearest evidence of this comes from two cases where GPT gives a partially correct response:

- ויהי סער [גדול בים] והאניה חשבה להשבר 'and there was a [great] storm [on the sea]'
- ויפלו גורלות ויפל [הגורל על יונה] ויאמרו אליו 'and they threw lots, and [the lot] fell [on Jonah], so they said to him …'

In both cases, GPT predicted only the first word (גדול 'great' and הגורל 'the lot'). Crucially, in most English translations, the translation of the first Hebrew word is separated from the translation of the remainder of the lacuna due to different word order (this is also the case in the English translation that GPT 3.5 responds with given the prompt "Give me the text of Jonah 1"). What I suspect that is happening, roughly, is that GPT recognizes that the given Hebrew text is that of Jonah 1, and uses the English to fill gaps in the Hebrew. However, it only recognizes the first part of the lacuna, because the second part is separated from it in the English translation.

More evidence comes from the fact that the model correctly predicts "קום לך אל [נינוה] העיר הגדולה" 'stand up, go to [Nineveh], the big city', even though Nineveh is not mentioned elsewhere in the text. This can again easily be explained if GPT suffers from interference from an English translation.[7]

Elrod (2023) performed a second test using a text with actual lacunae, using a fragment from the Dead Sea Scrolls. This is a much better test case, as the text and its translation are less likely to have been part of the model's training data. Predictably, the suggestions are much less useful. Elrod (2023: 17) claims that the results should be better if content of the Dead Sea Scrolls were included in the training data. If, however, the difference in accuracy is primarily due to the availability of an English translation, this would not help much.

To be sure, the use of machine learning for text restoration is an interesting avenue for exploration, and below I discuss some work that has been done in this direction. However, using pre-trained LLMs of the scale of GPT is an unsound methodology, as most of the training data of these models comes from languages other than Biblical Hebrew.

Text generation

Given that GPT can generate natural language text, we could consider using it to generate new texts in Biblical Hebrew. Elrod (2023) does not make explicit what use case he has in mind, but one could perhaps imagine something like this being used in teaching. The experiment consisted of asking GPT to generate a new text in Biblical Hebrew using only words from (excerpts of) Deuteronomy 1:1–13 and Ruth 1:1–8. The result is as follows:

[6]It is not clear that the input to the model was constructed correctly. Elrod (2023) replaced some words in the consonantal text by numbers in square brackets, as in "[העיר הגדולה2אל]" ('to [2=Nineveh] the big city'). However, in the input text given in the paper, gaps [13] through [22] are followed by the text that should have been omitted, e.g. "[הגורל על יונה13ויפל]" ('and [13=Ø] the lot fell on Jonah'). This does not affect my evaluation of the experiment, however.

[7]Elrod (2023: 12) suggests that the mention of "Jonah" may hint at "Nineveh". This is a possibility as well, but also assumes interference.

האלהים אשר דבר אל משה, דיבר אל כל ישראל, בארץ האמרי והכנעני. איש מבית לחם יהודה הלך לגור בארץ אחרת. יהוה, אלהינו, דבר אלינו בחרב, ראה נתתי לפניכם את הארץ. באו ורשו את הארץ הכנעני והלבנון, יהוה נשבע לאברהם, ליצחק וליעקב לתת להם ולזרעם אחריהם. ויהיו שם בימי שפט השופטים, עם עלה מארץ מואב, כי שמעה שם יהוה פקד את עמו לתת להם לחם. ויברך אתכם כאשר דבר לכם. אלה הדברים אשר צוה יהוה אתו להם.

The God who spoke to Moses spoke to all Israel in the land of the Amorites and the Canaanites. A man from Bethlehem of Judah went to live in another land. YHWH our God spoke to us at Horeb, saying, "I have set the land before you. Come and possess the land of the Canaanites and Lebanon, which YHWH swore to Abraham, Isaac, and Jacob to give to them and to their descendants after them." They were there in the days of the judges, with people who had ascended from Moab, for they had heard that YHWH had remembered His people to give them bread. He will bless you as He has spoken to you. These are the words that YHWH commanded for them.

(GPT-4 output from Elrod 2023: 19)

Elrod finds that "GPT-4 successfully generated a unique and largely coherent piece of text" (2023: 19). In my opinion, the coherence of this output can be debated. Some longer stretches of text are internally coherent ("They were there … to give them bread"), but these sentences together do not form a coherent whole. Furthermore, the longer stretches of internally coherent text seem to be taken more or less verbatim from the input text. In any case, these qualifications are subjective and should be replaced by proper benchmarks.

More importantly, the generated Hebrew contains several unexpected and possibly ungrammatical features. It is not clear why 'for they had heard' is translated "כי שמעה" rather than "כי שמעו".[8] The pronominal suffix on אתו in "הדברים אשר צוה יהוה אתו להם" 'the words which Yahweh commanded [it/him?] for them' is not ungrammatical if it refers back to Moses, though this would be a dependency over a rather long distance.[9] Note also that the Hebrew for 'with people who had ascended from Moab' is "עם עלה מארץ מואב" and that either 'with' or 'people', but not both, can be the translation of עם.

Some of these unexpected features can again be explained by interference from other, modern languages. For example, the generated Hebrew for "The God … spoke to all Israel" is " האלהים … דיבר אל כל ישראל", with the Modern Hebrew form דיבר for *dibbēr* rather than דבר (or וידבר). Interference from other languages may also explain the odd subject-initial asyndetic relative clause "... יהוה נשבע" '[which] Yahweh swore …', and the lack of complementizer כי in "שמעה שם יהוה פקד" 'they heard there [that] Yahweh had visited'. The interference from Modern Hebrew yields the model unusable for this task. There is no obvious path to improvement, because training on high-resource languages will be necessary for the complex task of producing coherent texts.

Moving forward

[8]A reviewer suggests that שְׁמֻעָה 'report' might be intended rather than שָׁמְעָה 'she(!?) heard': 'there was a report'. I find this unlikely because (a) the model was instructed to use only words from the two input texts; (b) the noun is usually spelled plene, which would therefore also be the form expected here; and (c) שְׁמֻעָה always appears with a form of the verb שׁמע or a verb of motion in narrative texts, so 'there was a report' is unexpected.

[9]I am grateful to a reviewer for this suggestion. In the comparable case in Deut 1:3 the referent of the suffix is considerably less far away. In Num 8:20 the object is repeated to avoid confusion.

Given the significant limitations of pre-trained LLMs for Biblical Hebrew, as well as the potential pitfalls in interpreting their results, is a way forward possible? It depends on the task. Some of the use cases considered above are essentially solved problems. We have a tagged syntactic database for the Hebrew Bible with a deterministic computational interface, so it is not clear what the benefits of an LLM-based search engine or database interface would be. But other tasks, such as text restoration and generation, are not solved yet and would clearly be useful to be able to automate.

For these other tasks, it appears to be crucial to avoid interference from English translations as well as Modern Hebrew. This is currently not possible with large-scale pre-trained LLMs, and there is no reason to assume that this will change in the near future, as large tech companies have no incentive to work on this. The alternative is to use other methods, such as models trained specifically on Biblical Hebrew (transformer-based or otherwise).

Some work on this has been done, but for other tasks than those considered by Elrod (2023). For instance, Naaijer et al. (2023) present a transformer-based parser for Syriac morphology and Wilson-Wright (2023) uses a transformer-based model to predict whether a text is written in Archaic, Early, Transitional, or Late Biblical Hebrew. It is important to note that both models are mainly doing morphological analysis. For syntactic or semantic analysis of the type needed for text restoration and generation, it is likely that vastly larger training sets are needed. For comparison, Assael et al. (2022) present a transformer-based text restoration model for Ancient Greek which reaches 61.8% accuracy with a training set of 63,014 inscriptions and over three million words.[10]

As always, it is also important to be clear about the goal of the model and the context in which it will be used. The goal of Naaijer et al.'s (2023) Syriac parser is to reduce the amount of work needed to parse Syriac texts. If the results are to be used in a database, possibly after manual correction by a human annotator, the most important characteristic of any tool is its accuracy. If a transformer-based model is currently the best in class, that is good reason to use it. As suggested by a reviewer, such applications are of course not limited to the realm of morphology; noun phrases could also be automatically tagged for animacy, for instance.

There are also contexts in which accuracy is not the most important. For example, in the context of chronological attribution of biblical texts (as in Wilson-Wright 2023), we are presumably not only interested in *whether* a text is early or late, but also *why the model thinks so*: we want to understand how the model thinks that the language has changed. Here it becomes important that transformers are fundamentally black-box models: although work has been done on explaining black-box models (Amgoud 2023), it remains difficult to explain why a transformer-based model gave a certain output given the input.[11] Other machine learning methods are white-box and therefore more easily interpretable, but often have lower accuracy. One cannot say in general that one type of model is preferable over the other; rather, one needs to be aware of the differences and choose the right kind of model depending on the overall goal and the context in which the model is applied.

Finally, it is important to understand that the use of any model can still involve certain assumptions. Consider again Wilson-Wright's (2023) use case of distinguishing Archaic, Early, Transitional, and Late Biblical Hebrew. It is possible to use machine learning methods to distinguish these corpora, but this does not necessarily constitute evidence that the corpora are distinguished *by date of writing* (to

[10]I am grateful to Aren Wilson-Wright for suggesting this reference.

[11]In the case of ChatGPT, one might think to simply ask the model in natural language why it gave a certain answer. However, in practice LLMs often come up with a new explanation that fits the earlier answer, instead of actually explaining how they arrived at the answer in the first place (cf. Elrod 2023: 12, note to table).

be sure, Wilson-Wright does not claim that it does). For all their benefits, we must be sure to remember that these methods do not address fundamental concerns such as those raised by Young & Rezetko (2008) for chronological attribution, and similar concerns in other contexts.

Discussion

I have tried to show here that extreme care must be taken when interpreting the output of large-scale pre-trained LLMs like GPT for research purposes. Overall, the results of Elrod's (2023) experiments seem impressive at first: often, they are almost correct. One might think that future iterations of GPT could easily resolve the remaining issues. However, I have shown that the interpretation of these results requires much more care. We must be hesitant with using these tools as morphological or syntactic search engines, for which tooling is already available that approaches 100% accuracy. We have come to expect (near-)complete correctness of these tools, but GPT is currently unable to answer even simple questions of this type correctly.[12] Even if future iterations can improve on the accuracy, it will never be able to give accuracy *guarantees* of the type that dedicated software without machine learning can give. Using GPT to generate code for queries posed in natural language is a potentially interesting idea, but once again the lack of accuracy guarantees is reason for skepticism—especially since there are still ways to make tagged data sets accessible to a wider audience that have not been explored for Biblical Hebrew, such as graphical query languages.

LLMs would seem to be well-suited for tasks requiring natural language comprehension and generation (exegesis, text restoration, and the generation of new texts). However, in the experiments for these use cases, it seems that GPT's answers suffered from interference from both modern languages and translations and commentaries of the texts being studied. To avoid such interference, dedicated models for Biblical Hebrew are needed. These are fundamental issues that cannot easily be resolved in future versions of pre-trained models like GPT.

Ways forward are possible, using models specifically trained on Biblical Hebrew. It is important to be aware of the properties of such models before choosing which one to apply. In particular, as long as explaining results from black-box models is difficult, there may be reason to stick to white-box models in contexts where interpretability is important. If interpretability is less important, a black-box model (possibly based on transformers) may be a good option. However, given the small corpus size we must be aware that accuracy will be worse than on high-resource languages, especially for complicated tasks involving syntactic and semantic analysis.

Finally, a critical discussion of the use of large-scale LLMs like GPT cannot avoid mentioning certain ethical issues. It is well-known that artificial intelligence in general and LLMs in particular use enormous amounts of energy, the majority of which does not come from renewable resources (Strubell et al. 2019; Bender et al. 2021). Energy consumption is high for both training and application, so using pre-trained models do not resolve this issue. Of course, without any regulation in place, everyone can decide for their personal use case whether this is acceptable; I merely point out that there are ethical questions to think about here. Another important issue is that most advanced pre-trained LLMs, especially in terms of market share, are proprietary and behind a paywall. It is necessary to open source these tools and lower barriers to use them to create a level playing field for researchers from different

[12]A reviewer points out that GPT 4 "generally says that it cannot answer such questions". This was not my experience with GPT 3.5 (answers are non-deterministic and may differ from user to user), but also does not address the more fundamental question whether relying on LLMs for these tasks would ever be a good idea.

parts of the world and to improve reproducibility.[13] While the paywall is currently relatively low for most of these tools, it will likely rise as society becomes more accustomed to their use (cf. recent hikes in prices of streaming services like Netflix). Relying on proprietary pre-trained LLMs for research thus means setting up an undesirable flow of public money to large private corporations. Finally, the use of online LLMs like ChatGPT provides the companies behind them with important feedback to improve their systems, and thereby supports the status quo. The business model of these companies involves the application of LLMs in other domains, including high stake contexts where black-box models are discouraged by experts, such as the criminal justice system (Rudin 2019), and contexts in which biases may be amplified (e.g., on "environmental racism, …, biodiversity loss, or pollution"; Rillig et al. 2023: 3464). Given these ethical issues, researchers should have strong reasons for resorting to these large-scale LLMs. Alternatives should be considered and used whenever possible. As long as regulation (whether by governments or in another form, e.g. through journal policies) is lacking, researchers have to carefully consider whether the benefits of these tools outweigh their costs and risks.

Acknowledgements

I am grateful to Aren Wilson-Wright and Ellen van Wolde for commenting on earlier versions of this paper. I alone am responsible for the opinions expressed here and any possible mistakes.

References

Amgoud, Leila. 2023. 'Explaining black-box classifiers: Properties and functions'. *International Journal of Approximate Reasoning* 155:40–65. https://doi.org/10.1016/j.ijar.2023.01.004

Assael, Yannis, Thea Sommerschield, Brendan Shillingford, Mahyar Bordbar, John Pavlopoulos, Marita Chatzipanagiotou, Ion Androutsopoulos, Jonathan Prag & Nando de Freitas. 2022. 'Restoring and attributing ancient texts using deep neural networks'. *Nature* 603(7900):280–283. https://doi.org/10.1038/s41586-022-04448-z

Bender, Emily M., Timnit Gebru, Angelina McMillan-Major & Shmargaret Shmitchell. 2021. 'On the dangers of stochastic parrots: Can language models be too big? 🦜'. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, 610–623. ACM. https://doi.org/10.1145/3442188.3445922

Bird, Steven & Haejoong Lee. 2007. 'Graphical query for linguistic treebanks'. In *Proceedings, PACLING 2007 – 10th Conference of the Pacific Association for Computational Linguistics*. Melbourne. http://hdl.handle.net/11343/34835

Elrod, A. G. 2023. 'Nothing new under the sun? The study of Biblical Hebrew in the era of generative pre-trained AI'. *HIPHIL Novum* 8(2):1–32. https://doi.org/10.7146/hn.v8i2.143114

Elrod, A. G. 2024. 'Uncovering theological and ethical biases in LLMs: An integrated hermeneutical approach employing texts from the Hebrew Bible'. *HIPHIL Novum* 9(1):2–45. https://doi.org/10.7146/hn.v9i1.143407

Naaijer, Martijn, Constantijn Sikkel, Mathias Coeckelbergs, Jisk Attema & Willem Th. van Peursen. 2023. 'A Transformer-based parser for Syriac morphology'. In *Proceedings of the Ancient Language Processing Workshop associated with RANLP-2023*, 23–29. https://aclanthology.org/2023.alp-1.3

OpenAI. 2023. 'GPT-4 technical report'. https://doi.org/10.48550/arXiv.2303.08774

Rillig, Matthias C., Marlene Ågerstrand, Mohan Bi, Kenneth A. Gould & Uli Sauerland. 'Risks and benefits of Large Language Models for the environment'. *Environmental Science & Technology* 57(9):3464–3466. https://doi.org/10.1021/acs.est.3c01106

[13]This issue is not specific to LLMs, of course. For example, the Tiberias stylistic classifier for the Hebrew Bible is likewise closed source (https://tiberias.dicta.org.il/, retrieved February 9, 2024).

Roorda, Dirk. 2017–2022. *Annotation/text-fabric*. Zenodo. https://doi.org/10.5281/zenodo.592193

Roorda, Dirk, Christiaan Erwich, Cody Kingham & SeHoon Park. 2017–2023. *ETCBC/bhsa*. Zenodo. https://doi.org/10.5281/zenodo.10049740

Rudin, Cynthia. 2019. 'Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead'. *Nature Machine Intelligence* 1(5):206–215. https://doi.org/10.1038/s42256-019-0048-x

Strubell, Emma, Ananya Ganesh & Andrew McCallum. 2019. 'Energy and policy considerations for deep learning in NLP'. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 3645–3650. Florence: Association for Computational Linguistics. https://doi.org/10.18653/v1/P19-1355

Tunstall, Lewis, Leandro von Werra & Thomas Wolf. 2022. *Natural Language Processing with Transformers: Building language applications with Hugging Face*. Beijing: O'Reilly.

Van de Bijl, Etienne P., Cody Kingham, Wido van Peursen & Sandjai Bhulai. 2019. 'A probabilistic approach to syntactic variation in Biblical Hebrew'. https://doi.org/10.5281/zenodo.2546802

Van der Schans, Yanniek, David Ruhe, Wido van Peursen & Sandjai Bhulai. 2020. 'Clustering biblical texts using recurrent neural networks'. https://doi.org/10.5281/zenodo.4003509

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser & Illia Polosukhin. 2017. 'Attention is all you need'. *Advances in Neural Information Processing Systems 30 (NIPS 2017)*. https://doi.org/10.48550/arXiv.1706.03762

Wilson-Wright, Aren. 2023. 'COHeN'. https://huggingface.co/gngpostalsrvc/COHeN, version 82ff154, retrieved March 18, 2024.

Young, Ian & Robert Rezetko. 2008. *Linguistic dating of biblical texts*. London: Equinox.