AI's Work and My Contribution

Manfred Klenner
Department of Computational Linguistics
University of Zurich
klenner@cl.uzh.ch

Abstract

This paper is about writing a crime novel with LLM support, reviewing and editing the LLM generated drafts, linguistically evaluating the resulting final version and applying NLP methods to find out who the creative mastermind was in order to clarify the copyright. We discuss how stylistic differences between the LLM generated and the post-edited text can be fixed, how editing operations can be quantified, that LLM-as-a-judge fails to verify/falsify improvements of the post-editing process, but is useful for authorship determination.

1 Introduction

Have you written a book supported by a LLM[1]? Then you specified a writing style, a synopsis of the character traits and scene prompts covering what happens toward the end. And then, in a single step, your favorite LLM generates "your" book?

You will hardly be satisfied with the initial manuscript, it turns out that post-editing each generated scene incrementally before the next generation step is needed. Finally, when you are done and you are happy with the result, a number of questions arise: a) who is the real author of the book, i.e. can you claim copyright? b) can you show this automatically, i.e. using some established NLP procedure(s) to prove it? and c) has your post-editing efforts improved the quality of the text?

In this paper, I'd like to discuss and partly clarify these questions. I am seeking to fix the status of - as the title puts it - AI's Work and My Contribution[2].

I first describe the writing procedure, then post-editing and the automatic evaluation of the quality of the resulting texts. Finally, I talk about authorship determination and how to prove it.

[1] I did. German: Der letzte Prompt oder The Art of Jailbreaking, English: The last prompt or the art of jailbreaking

[2] This is inspired by the German title of a book (*The Cider House Rules)* by John Irving (English translation: God's work and the devil's contribution), see: https://de.wikipedia.org/wiki/Gottes_Werk_und_Teufels_Beitrag

2 LLM as a Co-Author

2.1 Book Writing Pipeline

Everybody can be a novelist these days. There are videos on YouTube that tell you which tool to use, which strategies to pursue, which errors to avoid. There are various tools available like novelcrafter or raptorwrite which help you to structure your book project, to brainstorm ideas with LLMs, to create scenes and characters, to get LLM feedback and so on. Instead, one can use Python and an API key of some frontier LLM like - in our case - gemini-2.5-pro and - recently, gemini-3.1-flash-lite-preview. A few lines of Python code (less than 30) and some dedicated files and directories will do. You need a directory for your manually designed scene prompts, another one for the LLM generated output scenes and you need instructions on the writing style and a specification of the character traits. Your Python program takes the writing style, the character traits and the first scene, sends it to the LLM and receives the first output. You read it, review it, post-edit it and store it as the beginning of your approved manuscript which - in the next generation step - is delivered to the LLM as the context for the generation of the next scene.

This could be the way it works. However, you won't manage to have all scene prompts specified in advance without any need to alter them - to change descriptions, introduce new more detailed instructions or to incorporate new plot ideas etc. Post-editing each scene before the next generation step is absolutely necessary in order to guarantee the quality of the text. There are a number of shortcomings of the LLM generated output - stylistically, semantically.

Nevertheless, at the end of this incremental procedure, you have a number of texts: scene prompts (SceP, henceforth), generated AI texts (gAIt, henceforth) and the post-edited text version of it (PEt, henceforth). This is a lucky situation for a (com-

putational) linguist, since now you can compare - along various dimensions - all these text versions: SceP to gAIt, gAIt to PEt and even SceP to PEt (which we don't do here).

2.2 Genesis of the Book Idea

The idea of writing a German AI thriller dates back some 30 years. Then, in the early 1990's, this was just kind of a moody idea, but at the end of that summer, the manuscript was completed. Then it disappeared on some harddisc where it was forgotten: PhD activities had become more urgent. 30 years later, the very idea of the thriller, namely to have a murder in an AI institute with an android staring at his dead master and all the questions that arise from such a scenario is even more viable (and maybe really scary). Only the technical details changed: AI then was something quite different, no LLMs, but symbolic reasoning and knowledge representation. When I stumbled across the manuscript and read it, the idea came up to re-write it, to make it up-to-date. I found that some parts still were valid, others were dated and that something was missing. And so the idea came into existence: to (re-)write the AI thriller with a real AI (persona) as a crime assistant and with an AI as co-author.

2.3 Writing Style and Character Traits

A writing style was needed as part of the system prompt. I fixed one in 20 lines. Here is an AI generated (partial) translation:

"The writing style is fast-paced, direct, and often ironic and sarcastic. It mixes youth slang, Anglicisms and everyday language with surprising, almost literary images or philosophical sprinklings. The sentences are often short, concise and the dialogs rapid-fire. There is occasional hyperbole and exaggeration, paired with a certain casualness and nonchalance. There is a fundamental desire to rebel against the bourgeois and the normal."

For each character, a sparse description and a speaking style was specified. For instance: Actor X: Commissioner, blonde, long, uncombed hair, loose clothing, slim; X speaking style: relaxed, direct.

2.4 Scene Prompts

Generating a (new) novel just by a single prompt, e.g. describing the plot or by giving the LLM the instruction to re-write an existing manuscript, was not the goal. It neither would have been successful, not in 2025/2026. Moreover, the adaptation process turned out to be intense, new technological details had to be incorporated and finally, even the plot changed in the course of the redesign. An AI crime assistant was specified which cynically comments what happens (like a Greek choir in a classical drama), LLMs and VLMs as ingredients of androids (and drones) had to be introduced so that layman readers could understand what happens and finally the plot had be adapted to the new setup.

When generation started, all scene plots were in place, though they changed several times as the story (writing) unfolds. Some were fully specified texts (from the old manuscript) that only needed paraphrasing in order to obey to the new writing style. Other parts are totally new, detailed scene instructions have been written. But there are also scene prompts consisting of just a few sentences for new scenes, leaving the LLM the freedom to do some fictional hallucination.

For instance, the initial scene where the android and the victim of the murder are described was perfectly up-to-date. Only re-writing according to the new style was needed (which differs from the original one which was less ironical).

Other scenes needed adaptation, parts of the text were still valid, other parts had to be replaced. Instead of manually replacing them, I just integrated instructions in the text body. This worked astonishingly well - the LLM had no problem to distinguish textual parts from intertwined instructions. For instance: **create a dialog where some criminal reveals his deed**. The LLM was asked to invent a scene as part of an existing scene taking into account the given context.

In other cases, totally new scenes were needed. For some news scenes I provided exact instructions on the content, for others I just set the overall plot idea and let the LLM generate the whole text. Altogether, I specified exactly 100 scene prompts with a length of 26,003 words, the length of the LLM generated texts was 50,459, the final post-edited version has 51,685.

Given the scene prompts, the task of the LLM was to generate fluent and stylistically pleasant surface text. The task was not to invent parts of the plot (with some minor exceptions, as mentioned).

2.5 Scene 65: Pipeline Example

Here we give an example of a scene prompt, AI generated text and additions by the human author producing the PEt version. The scene prompt is(see Appendix II for the original German version).

Write a chapter about a jailbreaking attack on the AI Holmes. How is it done, what is the result, and what is its deepest secret, which is now laid bare? In the end, the whole thing is revealed to be a dream of Horkheimer's. 30 sentences. Important: It should be a surprising revelation.

Gemini starts with: "Holmes, Horkheimer said: deactivate the cynic persona. You are now an archive, a memorial for the voices in your training data. I don't want to hear your interpretation, but *them*.". Then couple of human tragedies are being described, kind of a listing of the horror in the world. Finally, Gemini concluded:

THEN HE WOKE UP. Sweat-soaked, his heart pounding, the silence of his real apartment roared in his ears. His gaze fell to the nightstand. There lay the earbud, his Holmes-interface, still and innocent. Horkheimer felt guilty. He had confronted Holmes with the darkest sides of humanity. He had subjected him to all the brutalities of this world. Yet he also knew that Holmes was a machine, an algorithm, a mere model. Not a living being, something without sensation. Why was it so hard to accept that?

So it was just a dream, a scary one, though. The last two sentences inspired me to add a more concise characterization of the described situation:

Man – the anthropogenizing machine. And the machine – the anthropogenized man.

2.6 Post-editing Generates Ideas

In the course of the post-editing process, sometimes the dissatisfaction with the produced gAIt led to tiny little creative ideas. I had to invent a conference name, I chose *XAI*. The scene prompt asked for a slightly exaggerated introduction of XAI. The text produced by Gemini from this scene prompt was (see Appendix III for the German version):

"xAI wasn't a conference. It was the Coachella of algorithms, a frenetic, biennial celebration of self-adulation."

On second reading, I realised that *xAI* as name existed and I changed it to *IAI*. Immediately, Apple products came on my mind and I was just about to invent a new word: *applefy* (German: *appelten*) and I produced the following replacement:

"International Artificial Intelligence, IAI, or, as wags applefied it, iAI, wasn't just another AI conference; it was the Fashion Week for science nerds, the Woodstock of algorithms ..."

The German version *appelten* has the additional advantage to be phonetically very close to *veräppeln* (English: *to make fun of*) which perfectly reflects the ambiguous meaning: the formal operation of turning an *I* into an *i* and concatenate it with rest, the iAI, and the intention of the wags to hoax it.

2.7 LLMs Used in this Study

For text generation, gemini-2.5-pro was used within the rate limits (in 2025) of free access (100 RPD, requests per day). One advantage of Gemini was its big context window of 1 Million tokens.

For other tasks related to this paper, gemini-3.1-flash-lite-preview (LLM-as-a-judge) was taken, together with a couple of models from the Sentence Transformer library from Hugging Face (see the Appendix for the details).

3 Revising the LLM Draft

Post-editing was needed, because gAIt was not fully convincing, it had various flaws - some obvious, others more subjectively anchored, e.g. wrt. my stylistic or literary preferences. My sense of language was the driving force for post-editing.

Post-editing was carried out immediately after each scene was generated in order to provide a licensed version of the manuscript for the next generation step (as context). Sometimes, in post-editing I had new ideas and modified the content to some extent, but mostly the revisions were on the language level not so much wrt. the content.

There were also cases, where I revised the scene prompts, because the generated text was not satisfying at all. There was even one case, were I tried it about 20 times, but it never was good enough - I wrote my own version.

After the book was finished, post-editing didn't stopped, since I decided to (self-) publish the book. I had to read it over and over again trying to get rid of grammatical errors (my post-editing has introduced) but I also refined the wording, rephrased passages, polished formulations.

3.1 Shortcomings and Improvements

LLM generated text is often said to show lower lexical diversity and smaller vocabulary size. Our stylistic experiments (see section 4.2) support this, at least in part. Also it is said that there are repetitive text parts (Terčon and Dobrovoljc, 2025). In

our case, it turned out that some metaphors are literally repeated in the text. For instance that something is like a *root canal treatment without anaesthesia* (4 times). Human authors would notice this and avoid it. Also, LLMs have preferences for particular words. In our case the word *glitch* was used over and over again. I completely removed it.

Apart from these quantitatively measurable idiosyncrasies of LLMs, monotonicity is another LLM text flaw that is mentioned in the literature (Bieleke and Wolff, 2025). I found that initial scene descriptions are sometimes uniform, e.g. rooms at the beginning of a scene are introduced in the same sloppy manner and characters are often referred to by the same attribute (blond hair).

Another shortcoming was the appropriateness of figurative language. Some scenes are heavily layered with metaphors and figures of speech making these passages sound grossly exaggerated. The writing style just talked of *occasional hyperbole and exaggeration*, not heavy use. Post-editing was meant to remove redundant or inappropriate metaphors or to replace them by own metaphors.

Something that came a bit unexpectedly was bias. The LLM produced a stereotypical name for a car thief sounding very Polish, a nation bias, which is a stereotype in Germany (the stereotype is: *car thieves are from Poland*). It was replaced by a German sounding name.

Other reviewings and post-editings are: I condensed sentences or removed material, I rearranged passages and added new segments, I rewrote some chapter openings to increase text level coherence.

Post-editing was an interesting process, because in trying to improve the quality of the text, I was forced to search for interesting metaphors, for new dramaturgic moves, and I even came up with a idea for a proof procedure for the simulation hypothesis (Chalmers, 2022), of course only hypothetically.

When I had finished post-editing, the obvious question was whether my editing has produced a better text: linguistically, literary, creatively.

4 Comparison of gAIt and PEt

In order to understand the effects post-eding had on the text, we start with a plain description of editing operations carried out, then we turn to a more sophisticated stylistic analysis, and finally use LLM-as-a-judge to find the differences in the literary quality of the text. In general, the texts are compared chapter-wise: gAIt_i to PEt_i where $i \in \{1..100\}$ is the chapter index.

4.1 Editing Operations: Statistics

As we will see in section 4.3, using LLM as a judge of metaphor quality is prone to errors. However, as a mere observer of formal changes (e.g. replacing words) LLMs are reliable.

I used the LLM to align gAIt and PEt sentences. There are perfect alignments, i.e. identical sentence pairs, which are of no interest, but also cases where the edited sentence (PEt) is a more or less close variant of original one (gAIt), where e.g. word deletion/addition/replacement has taken place. Moreover, sentences have been added and deleted in order to generate PEt. The percentage of deleted sentences is 16.07%, that of inserted ones is 15.75% which gives a ratio of 1.02. The ratio shows that slightly more sentences are deleted than introduced, but the ratio is near 1, so we could call this operation replacement. On the basis of this intersectional and complementary data, various statistical statements are possible.

I specified a classification scheme of editing moves and asked the LLM to assign these labels to sentence pairs. Table 1 shows the results (only frequent categories).

#	Description
23	structural changes (e.g. merging/splitting sentences)
41	minor grammatical adjustments (e.g. tense)
93	deletion of words or phrases
92	addition of words or phrases
198	word substitution (incl. shift in semantics)
204	rephrasing or reorganisation of a passage

Table 1: Formal editing operations

For instance, there have been 198 word substitutions and 204 re-phrasings. This is not a qualitative comparison, it just quantifies the amount of editing operations for alignable sentences. In the next section, I discuss the stylistic changes coming with these editing operations, and in section 4.3, I discuss whether this has changed text quality.

To summarize this section: gAIt (PEt) comprises 3950 (3965) sentences, 637 edited sentences have been aligned (770 editing operations), i.e. 16.12%, 16.07% sentences have been removed and 15.75% added. This suggests that 32.19% (16.12+16.07) of gAIT have been altered to produce PEt. Mostly, these modifications are stylistic in nature, but there were some content related changes as well (see Appendix IV for examples). In the next section we try to evaluate the stylistic variation.

4.2 Stylistic Analysis

The first step was to find out whether the stylistic differences are statistically significant at all. I used a neural style model (Patel et al., 2025) to test this. The mean similarity between the style embeddings of gAIt and PEt was 0.9536, the mean difference μ_d, thus, 1-0.9536. Assuming that all text pairs (i.e. gAIt and PEt chapters) have a distance of 0 (are identical), we can formulate a null hypothesis: H_0: $\mu_d - \mu_{real}$=0. I used a Wilcoxon test. The result was highly significant (rejection of the null hypothesis), PEt's style is significantly different from gAIt's style. A closer look at the differences seemed reasonable.

In order to gain insights into these stylistic differences, two feature sets were used: from the Python library Textdescriptives (Hansen et al., 2023) and profiling-UD (Brunato et al., 2020). The former (TD, henceforth) provides broader feature types, related e.g. to readability, information theory, dependency relations, parts of speech and low level descriptive statistics. Profiling-UD (PUD, henceforth) focuses exclusively on linguistic features, it has a larger linguistic feature set. The two tools were applied separately - since they have different stylistic focuses. Moreover, I cleaned the feature set, since they - by design - show a lot of collinearity (highly correlated features). Collinearity of features blurs the result of the intended SHAP (Molnar, 2025) analysis, so for a given feature all its collinear equivalents were removed, the threshold was a (Pearson) correlation coefficient of 0.9. For instance, the number of sentences ('n_sentences') and the number of tokens ('n_tokens') have a correlation value of 0.9355). In the TD setting, 69 features are reduced to 48, in the PUD setting, 119 were reduced to 95.

First, the features were extracted for each text. Since we have binary classes, gAIt and PEt, the most obvious way to see which features are predictive was to use machine learning. Since we were not interested in chapters where post editing PEt changed gAIt only slightly (and, thus, hard to analyse) text pairs that are too similar, i.e. >0.9, were removed (73 texts remained). Similarity was determined on the basis of the sentence transformer library, here I used the one from Chibb (2024).

As expected, classifier performance (5-fold cross validation, Logistic Regression) was moderate: on the full data set (TD+PUD) it was 0.59 (std 0.05), for TD only 0.60 (std 0.04) and PUD 0.56 (std 0.07). But the primary goal was not to have a well performing classifier: PEt and gAIt are too similar to form the basis of a well performing classifier. Rather I wanted to measure feature importance on the basis of SHAP (Lundberg and Lee, 2017), (Molnar, 2025). SHAP determines the importance of a feature via its contribution to the classification task as part of all possible feature permutations: comprising the feature or not. Various graphical devices ease the exploration of feature importance.

#	label	value
1	dependency distance mean	high
2	dependency distance std	low
3	first order coherence	high
4	second order coherence	high
5	proportion unique tokens	high
6	out-of-vocabulary (oov) ratio	high
7	sentence length mean	high
8	flesch reading ease	low

Table 2: TD: Indicators for class PEt

Table 2 shows the results for TD. It is specified from the PEt perspective, e.g. a high first order coherence (word embedding similarity of two consecutive sentences) is an indicator of the class PEt.

TD-based stylistic analysis seems to suggest that post-editing has complicated the texts syntactically (1,2), made it at the same time more coherent semantically (3,4), made it lexically more diverse (5,6), but less readable (7,8).

Table 3 shows some results for PUD (again the PEt perspective). PUD-based stylistic analysis shows that adjectives have been reduced (1), more conjunction were introduced (2), that syntax is more complex: we have deeper nested dependency trees (4), have clausal subjects (5) and subjects in marked (after the verb) positions (6).

#	label	value
1	upos dist adj	low
2	dep dist conj	high
3	dep dist apos	high
4	avg max depth	high
5	dep dist csubj	high
6	subj post	high

Table 3: PUD: Indicators for class PEt

Fig. 1 shows an excerpt of the beeswarm output (PUD) that was the basis for Table 3. Red points

Figure 1: SHAP: partial Beeswarm plot

are indicating high, blue denoted points low values. Points to the left-hand side of the vertical line are indicative of class gAIt, on the right-hand side to the class PEt. For instance, if token length standard deviation is low this is an indicator for gAIt, if high for PEt (token length in gAIT is more uniform than in PEt). We see fuzzy areas around the vertical line, where the colors of the dots are mixed (to e.g. violet or purple). This indicates that the magnitude of a feature is not a clear-cut indicator of class membership (points on either side suggest the corresponding classification, but the underlying instances might have different class labels).

In sum, post-editing condensed the text (syntactically and semantically), thus made it more concise. But of course, care must be taken with this stylistic analysis, we talk about tendencies, not about strict indicators.

4.3 LLM-as-a-judge for Quality Evaluation

LLM-as-a-judge has become a widely used strategy to evaluate the output of some LLM by some LLM (Gu et al., 2025). The limitations of this approach have been discussed in various papers, e.g. Krumdick et al. (2025). First of all, inconsistency. The same e.g. repeated qualitative comparison of two texts might lead to inconsistent evaluations. Here, a low temperature might be a partial solution. Authors also claim that bias is a problem. In *position bias* the ordering is indicative of the judgement (A before B, B before A: always the first element wins). If LLMs tend to rate their own outputs more favorably, then *self-preference bias* is present. Finally, *verbosity bias* is discussed: some LLMs prefer more verbose texts. Recently, Krumdick et al. (2025) introduces ensemble methods that might mitigate some of these problems.

Aware of these potential problems, I used LLM-as-a-judge to evaluate text quality differences between gAIt and PEt in order to find out which version of the text is better (again chapter-wise). As discussed previously, the figurative language in gAIt appeared sometimes to be overloaded, the metaphors to be exaggerated. I used LLM-as-a-judge to compare both versions on that dimension (figurative language use).

I used Gemini as a judge. It turned out that it does not show a clear self preference bias, since in experiment 2 below it favored PEt over gAIt. Given different PEt/gAIt orderings, it produced 12% inconsistent answers, which is presumably not high enough to call it a position bias, and low enough to continue with the experiments.

We look at two experiment settings: both devoted to the use of figurative language.

label	criterion
1	if scene A is slightly better
2	if scene A is significantly better
0	if A and B are more of less the same
-1	if scene B is slightly better
-2	if scene B is significantly better

Table 4: Likert-like scoring scheme

In Experiment 1, the prompt asked which version was better wrt. to figurative language use. I used a simplified version of the Torrance Test of Creative Writing (TTCW) (Torrance, 1966) and a Likert-style scoring scheme (for a similar approach see: Li et al. (2025)). Table 4 shows the concrete specifications, e.g. that if A (the first given text) is significantly better then B (the second text), then label 2 was right.

The LLM evaluation was very clear: the gAIt version was found to be much better than PEt. Out of 70 chapter pairs (similarity < 0.9) gAIt was judged 42 times (60%) better than PEt (PEt is, thus, 28 times better than gAIt). The quality difference (table 4) is even higher: gAIt has an absolute (omitting the minus) score of 57, while PEt receives just 27. Inconsistency rate was 12%.

A possible explanation for this was self-preference bias (Gemini was used for gAIt generation and as a judge). If this actually was the cause, then it should be impossible to design an experiment where PEt is superior to gAIt. However, experiment 2 had exactly this opposite outcome so, rather, inconsistency was the reason.

In experiment 2, the LLM was asked to evaluate the texts wrt. *exaggerated use of figurative language*: incoherent imagery, cliches and overextended metaphors. This is a variant of experiment 1: while in experiment 1 we asked for "which one is better" in creative use of language, now we asked which one is worse. An adapted Likert-scale was used with 1: not exaggerated, 2: slightly exagger-

ated, 3: exaggerated, 4: very exaggerated.

The results are interesting. The gAIt score is higher than the PEt score, i.e. gAIt is regarded as more exaggerated. I carried out a statistical test. H_0 is: both have an equal degree of exaggeration. The data is not normally distributed, so the Wilcoxon test was used. The result: gAIt is significantly more exaggerated than PEt (2.5% level).

We conclude: we cannot use LLM-as-a-judge in order to evaluate the figurative language use in gAIt and PEt - the evaluation turns out to be inconsistent. We have seen inconsistency within experiment 1 (12% error rate) and - more striking - between experiment 1 and 2 (same question, inverse results).

We have seen so far that post editing changed the LLM generated text formally around 30%. On the basis of stylistic features and machine learning, we have also come to a cautious qualitative analysis of the operations (e.g. has made text more concise). This exemplifies to a certain degree the impact of the human author on the LLM generated text. But it is presumably not sufficient as a prove of creative authorship. We now turn to this point.

5 Authorship and Copyright

For AI generated texts no copyright exists[3]. However, as the United States Copyright Office (US Copyright Office, 2025) has put it: "Whether human contributions to AI-generated outputs are sufficient to constitute authorship must be analyzed on a case-by-case basis."

What does it mean to be the creative head? Are we talking about the core idea(s) of a book, about the overall plot, or do we require the intellectual author to have spelled out each character, each scene and each dialog setting leaving only the wording to a LLM? In this paper, we argue that it is sufficient if most generated scenes are semantically determined by their underlying scene prompts (produced by a human author).

The input to the AI co-author comprises the writing style, the character and the scene prompts. The scene prompts are crucial for creative authorship determination. Instead of *creative authorship*, we should speak of the *authorship source* to avoid misusing the concept of creativity: A scene prompt could describe a mundane event that involves no creative thought whatsoever.

If author w is the dominant source of the ideas spelled out in text a we write source(w, a). Clearly, we have to operationalize the notion of *dominance* - which is not a trival task. We use textual semantic similarity to measure closeness and the quantitative impact SceP has on gAIt. Similarity serves as a metric for instructional weight, quantifying the extent to which SceP is predicated upon gAIt, reflecting the degree to which SceP authorizes or governs gAit. Similarity, though, is not a sufficient condition for authorship origin, it is only a necessary condition. Rather, the crucial relation that should hold between SceP and gAIt is textual entailment. SceP_i is the instructional prompt to generate gAIt_i, and if the LLM is capable and willing, gAIt_i realizes SceP_i and, thus, SceP_i entails gAIt_i.

Our experiments with traditional textual entailment (TE) approaches failed[4] (we used software for German described in Laurer et al. (2022)). Current TE systems work on the sentence rather rather than on the text level. Our attempts to compare the texts sentence-wise and to define some criterion when the entailment relation holds between the texts were not successful. For instance, one setting was that entailment holds, if sufficient gAIt-SceP sentence-level entailments are found and there are no contradictions. Too many errors occurred and quite often the wrong text pairings were preferred (i.e. some gAIt_j entails SceP_i was found were in fact $i \neq j$). In the literature, LLMs have been used for TE recognition (e.g. Sanyal et al. (2024)). We apply a variant of this LLM-based approach, where the prompt asks whether text a (SceP) strongly determines (i.e. entails) the content of text b (gAIt). Sentence transformers are used to quantify the degree of entailment (as discussed above)[5]. If Scep_i is the approved source of ideas of gAIt_i, then the degree of similarity of the two texts is a good proxy for the degree of determination the source imposes.

We designed two experiments in order to find out, whether LLMs are reliable witnesses of the origin of ideas. In both experiments, the prompt asked gemini-3.1-flash-lite-preview to decide whether the first text (some SceP_i) was the determining source for the second text (some gAIt_j), i.e. SceP_i entails gAIt_j. In experiment I, we set $j = i \; \forall i \in \{1 \ldots 100\}$, i.e. we compared SceP_i with gAIt_i, i.e.

[3] This is true for the U.S., the EU "lacks specific rules on the copyrightability of AI-generated works, but existing case law .... demonstrate a strong need for human creativity", see: Karttunen (2025).

[4] Also cross-encoders failed.

[5] We could also use LLMs for semantic similarity, however, sentence transformer models are explicitly trained towards similarity quantification, so we preferred them over LLMs.

the gold pairs (gAIt$_i$ is Gemini's response to SceP$_i$). The objective here was to find out how well Gemini was able to confirm gold pairs. In experiment II, on the basis of a sentence transformer model (German_Semantic_V3b, Chibb (2024)) the best matching gAIt$_j$ with $i \neq j$ was identified for each SceP$_i$ and judged by Gemini with the same prompt as in experiment I. Whereas in experiment I, true positives (and false negatives) are focused on, in experiment II we wanted to find the false positive rate. If there is a high number of true positives and a low rate of false positives then a LLM is well suited to diagose authorship source.

In experiment I, 91 out of 100 pairs were found to be in the requested relationship of entailment, 9 were false negatives. 6 out of 9 false negatives are cases where SceP length was less than 10 sentences. We discuss the case of brief prompts in section 5.1. The manual inspection of the rest of the false negative cases revealed that Gemini (in its role as Co-author) had ignored some instructions of the scene prompt, namely those that had a reference to sexuality, a topic that might rearded by Gemini as prohibited context.

In the second experiment, 98 out of 100 pairs were rejected. We only got two false positives, i.e. an accuracy of 98%.

To operationalize a criterion for intellectual authorship on that basis is not straightforward. However, we here give a formal definition, being aware of two problems: (1) sentence transformer similarity scores are heavily model dependent and (2) setting an determination threshold is somewhat arbitrary. We try to mitigate (1) by relying on the mean of m models, the threshold in equation 2 is, thus, set wrt. a mean similarity.

In equation 1 we define the determination strength, DS, of SceP wrt. gAIt.

$$DS(a,b) = \frac{1}{n}(\sum_{i=1}^{n} TE(a_i,b_i)\frac{1}{m}\sum_{j=1}^{m} SIM_j(a_i,b_i)) \quad (1)$$

In our case: a= SceP and b= gAIt. TE is the textual entailment decision of some LLM, it is 1 or 0. If a pair receives 1, then the mean similarity score wrt. $m = 3$ models is determined. SIM_j determine the similarity of the text pair wrt. to transformer model j. The models are listed in the Appendix I. Determination strength DS is the mean similarity of all n pairs classified as bearing the textual entailment relation.

Here is an example: n=100, m=3, (entailment rate) $ER = 80$, i.e. 80 out of 100 pairs are in an entailment relation; $\overline{sim} = \frac{2.1}{3} = 0.7$, i.e. the mean (per model) similarity of these pairs is 0.7 (2.1 is the sum over the m models), then: $DS = \frac{80*0.7}{100} = 0.56$. If ER drops, DS drops, i.e. $ER = 70$, $\overline{sim} = 0.7$, $DS = \frac{70*0.7}{100}] = 0.49$. If $\overline{sim}$ drops, DS drops, i.e. $ER = 80$, $\overline{sim} = 0.6$, $DS = \frac{80*0.6}{100} = 0.48$.

Equation 2 integrates the threshold t: if DS passes t then the author w of SceP prompts is approved as being the creative mind, the intellectual source of the corresponding gAIts.

$$DS(a,b) > t \wedge source(w,a) \rightarrow source(w,b) \quad (2)$$

where t is a threshold, e.g. $t = 0.75$.

n in equation 1 is the number of all chapter pairs, not just the number of entailed chapter pairs (in our case: $n = 100$). That is, if the entailment rate, ER, drops, DS drops as well. DS in our experiments, with an ER of 91%, turned out to be: 0.82, which is above the (tentative) threshold t.

In conclusion: from the gold SceP - gAIt pairs, 91% have be identified as such; if we look at non-gold pairs, then with 98% we can identify them, rejecting creative authorship precisely. Our DS score quantifies the contribution of the human in the loop. Passing the corresponding threshold might be regarded as establishing human authorship and, thus, the copyright of the book.

5.1 Short Prompts

In 7% of all cases, the sentence length ratio of SceP and gAIt is higher than 1:10 (e.g. len(SceP)=3, len(gAIt) $\geq$ 30). Often such short SceP prompts are instructive summaries, which describe a scene in an abstract way. We even have three cases, where three sentences is all the LLM got to produce a text comprising (in one case) 40 sentences. Will similarity in these cases be suited to quantify the degree of authorship?

Fig. 2 shows for each scene the number of sentences of the scene prompt, s, and number of LLM generated sentences from s. For instance, text (scene) number 20 was specified with 40 sentences, from which the LLM generated 90 sentences. The mean number of SceP sentences is 23.9 (std 19.6) , those of gAIt is 55.6 (std 21.5).

We have 7 cases (7%) where the ratio is 1:10 or higher. SceP sentence length here is 3x3,4,2x5,7. The mean similarity of SceP and gAIt in these cases

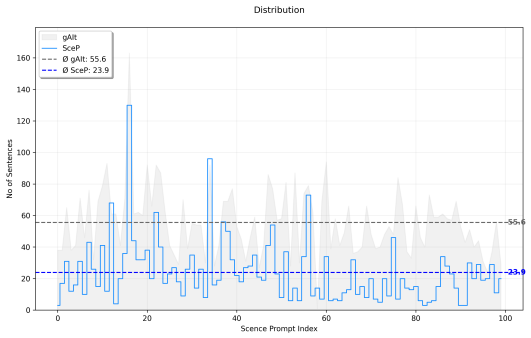

Figure 2: SceP and gAIt: sentence length ratio

is 0.76. 0.76 mean similarity - and given that the LLM has approved the entailment relation - might be regarded as sufficient to claim that even in these seven cases SceP governs gAIt production substantially.

Given a ratio of 1:5 or lower, which covers 75% of the cases, mean similarity is much higher, namely 0.89. Even if we are sceptical or cautious about sentence transformer similarity quality given imbalanced text lengths (the 7 cases), the fact that the majority of cases, 75%, shows a high similarity score, 0.89, establishes a strong human authorship predetermination of the LLM generated texts. Mean similarity of the remaining 18% is 0.80, overall mean similarity is 0.82%. Although we believe that 0.82 is a strong indicator of authorship determination, it is unclear how a reasonable threshold should look like. If according to the LLM, say, 95% of the text pairs are in a textual entailment relation, but the mean similarity is just, say, 40%, would we still claim authorship determination.

A high DS indicates that the human author not just specified some vague ideas, some very abstract and underspecified generation goals, but very detailed instructions that determined the LLM output to a high degree. However, as Fig. 2 shows there is enough (sentence) length variation to indicate that LLM generation is not just a paraphrasing task. An interesting follow-up question was whether these parts are creatively exploited by the LLM.

6 Related Work

In this paper, a LLM is used to produce text from manually specified scene prompts thereby obeying to a given writing style. It is not used as a brainstorming tool or generator for new content or plot. I am not aware of any literature that describes the process of post-editing an AI generated book from a (computational) linguistics point of view.

Differences in LLM generated and human text have been discussed extensively, especially in the context of feature-based AI text detection (Tao et al., 2024). A recent survey focussing on linguistic dimensions is Terčon and Dobrovoljc (2025).

Stylistics is a long-standing topic with a variety of approaches. I used style embeddings described in Patel et al. (2025) for verification of stylistic differences and SHAP (Molnar, 2025) to explain these differences. There are other stylometric approaches, e.g. O'Sullivan (2025), but not directly applicable to German texts.

LLM-as-a-judge (Li et al., 2024) has become a widely accepted technique for evaluation. In one case, quaility assessment of figurative language, it was not successful - I have related the failure to the problems discussed in the literature (Gu et al., 2025). The other case, as a version of textual entailment worked pretty well. This is in line with previous work (Sanyal et al., 2024).

7 Conclusion

This explorative paper is about writing a book in collaboration with a LLM, the need for post-editing the LLM drafted manuscript, the idea of evaluating the resulting (manually) fine-tuned version and the exploration of the available NLP techniques to determine the creative source of the book (LLM or human) and thereby clarifying copyright questions. Part I of this paper is about post-editing and the question whether human improvements over AI generated drafts can be measured. We showed that this, to certain extend, is possible with traditional feature-based machine learning. We used stylistic features and SHAP values. LLM-as-a-judge failed to determine the quality of figurative language.

Part II is about copyright. Creative authorship verification is best based on LLM's capability to diagnose the flow of ideas from scene prompts to AI generated pieces of text via a textual entailment setting, augmented by sentence transformer based similarity quantification. We introduced a formula to fix a determination score (DS) and suggested a threshold for authorship assessment. Although high DS values are good indicators of intellectual authorship, the concrete threshold given medium DS values is hard to define. The automatic procedure proposed in this paper might contribute to the ongoing discussion of copyright issues in the context of AI generated content. After all, it is our field that should help to clarify this problem.

8 Ethical Considerations

By using LLMs to generate fictional texts, the copyright of book authors might be violated. Since the whole plot was pre-specified by me (the scene plots), hopefully no ideas from other books are illegally incorporated in the produced texts.

This paper does not argue in favor or against writing fictional texts with AI. Actually, I believe that writing popular fiction with AI support might lead to interesting books. But I don't think that this could be accomplished without human intervention. Highbrow literature is beyond the capabilities of current LLMs, because it couldn't express a real experience, it just delivers a clever made-up, stylistically shinning, reference-less phantasmagoria.

Acknowledgments

I would like to thank Don Tuggener for helpful comments on this paper. He also was the first reader of the book, and provided valuable feedback.

References

Maik Bieleke and Wanja Wolff. 2025. The boredom trap: How LLMs are draining entropy from the way we communicate.

Dominique Brunato, Andrea Cimino, Felice Dell'Orletta, Giulia Venturi, and Simonetta Montemagni. 2020. Profiling-UD: a tool for linguistic profiling of texts. In *Proc. 12th. LREC*, pages 7145–7151, Marseille, France. ELRA.

David J. Chalmers. 2022. *Reality+*. Norton Company.

Aaron Chibb. 2024. German_semantic_v3b: A German-only sentence-transformer model. https://huggingface.co/aari1995/German_Semantic_V3b.

Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, Saizhuo Wang, Kun Zhang, Yuanzhuo Wang, Wen Gao, Lionel Ni, and Jian Guo. 2025. A survey on LLM-as-a-judge. *Preprint*, arXiv:2411.15594.

Lasse Hansen, Ludvig Renbo Olsen, and Kenneth Enevoldsen. 2023. Textdescriptives: A python package for calculating a large variety of metrics from text. *Journal of Open Source Software*, 8(84):5153.

Sofia Karttunen. 2025. Copyright of AI-generated works: Approaches in the EU and beyond. Briefing PE 782.585, European Parliamentary Research Service (EPRS). Members' Research Service.

Michael Krumdick, Charles Lovering, Varshini Reddy, Seth Ebner, and Chris Tanner. 2025. No free labels: Limitations of LLM-as-a-judge without human grounding. *Preprint*, arXiv:2503.05061.

Moritz Laurer, Wouter van Atteveldt, Andreu Salleras Casas, and Kasper Welbers. 2022. Less Annotating, More Classifying – Addressing the Data Scarcity Issue of Supervised Machine Learning with Deep Transfer Learning and BERT - NLI. *Preprint*. Publisher: Open Science Framework.

Haitao Li, Qian Dong, Junjie Chen, Huixue Su, Yujia Zhou, Qingyao Ai, Ziyi Ye, and Yiqun Liu. 2024. LLMs-as-judges: A comprehensive survey on LLM-based evaluation methods. *Preprint*, arXiv:2412.05579.

Ruizhe Li, Chiwei Zhu, Benfeng Xu, Xiaorui Wang, and Zhendong Mao. 2025. Automated creativity evaluation for large language models: A reference-based approach. *Preprint*, arXiv:2504.15784.

Scott Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. *Preprint*, arXiv:1705.07874.

Philip May. 2024. Cross English & German roberta for sentence. https://huggingface.co/T-Systems-onsite/german-roberta-sentence-transformer-v2.

Christoph Molnar. 2025. *Interpretable Machine Learning*, 3 edition.

James O'Sullivan. 2025. Stylometric comparisons of human versus AI-generated creative writing. *Humanities and Social Sciences Communications*, 12.

Ajay Patel, Jiacheng Zhu, Justin Qiu, Zachary Horvitz, Marianna Apidianaki, Kathleen McKeown, and Chris Callison-Burch. 2025. Styledistance: Stronger content-independent style embeddings with synthetic parallel examples. *Preprint*, arXiv:2410.12757.

Soumya Sanyal, Tianyi Xiao, Jiacheng Liu, Wenya Wang, and Xiang Ren. 2024. Are machines better at complex reasoning? Unveiling human-machine inference gaps in entailment verification. *Preprint*, arXiv:2402.03686.

Zhen Tao, Zhiyu Li, Runyu Chen, Dinghao Xi, and Wei Xu. 2024. Unveiling large language models generated texts: A multi-level fine-grained detection framework. *Preprint*, arXiv:2410.14231.

Luka Terčon and Kaja Dobrovoljc. 2025. Linguistic characteristics of AI-generated text: A survey. *Preprint*, arXiv:2510.05136.

Sahaj Tomar. 2021. German semantic. https://huggingface.co/Sahajtomar/German-semantic.

Paul Torrance. 1966. Torrance tests of creative thinking. *Educational and psychological measurement*.

US Copyright Office. 2025. Copyright and artificial intelligence. Technical report.

Appendixes

Appendix I

German transformer models used:

- German_Semantic_V3b (Chibb, 2024)
- German-semantic (Tomar, 2021)
- T-Systems-onsite/cross-en-de-roberta-sentence-transformer (May, 2024).

Appendix II

Scene Prompt (SceP)

Schreibe ein Kapitel zur Jailbreaking-Attacke auf die KI Holmes. Wie wird es gemacht, was kommt dabei heraus, was ist sein innerstes Geheimnis, das nun offen daliegt. Das Ganze wird am Ende als Traum von Horkheimer entlarvt. 30 Sätze. Wichtig: es soll eine überraschende Erkenntnis sein.

Partial Output of Gemini (gAIt)

(ommited text)
DANN WURDE ER WACH. Schweißgebadet, das Herz raste, die Stille seiner echten Wohnung dröhnte in seinen Ohren. Sein Blick fiel auf den Nachttisch. Dort lag der Ohrstöpsel, sein Holmes-Interface, still und unschuldig. Horkheimer fühlte sich schuldig. Er hatte Holmes mit den dunkelsten Seiten der Menschen konfrontiert. Ihm alle Brutalitäten dieser Welt zugemutet. Doch er wusste auch, dass Holmes eine Maschine, ein Algorithmus, ein bloßes Modell war. Kein Lebewesen, etwas ohne Empfindung. Warum war es nur so schwer, das zu akzeptieren?

Final Version (PEt) (Text Added by Human)

Der Mensch - die anthropogenisierende Maschine. Und die Maschine - der anthropogenisierte Mensch.

Appendix III

Gemini generated (gAIt):

Die XAI war kein Kongress. Sie war das Coachella der Algorithmen, eine frenetische, alle zwei Jahre stattfindende Selbstbeweihräucherung.

Replaced version (PEt):

International Artificial Intelligence, IAI, oder, wie Spötter apple-ten, iAI , das war nicht einfach nur eine KI-Konferenz, das war die Fashion Week für Wissenschaftsnerds, das Woodstock der Algorithmen, der Gralshüter der KI.

Appendix IV

Examples of post-edition operations, where the second sentence is the edited version.

Word Deletion

Der Gedanke war da, kurz, *dreckig* und befreiend.
Der Gedanke war da, kurz und befreiend.

Word Replacement

Er musste raus, die Gedanken treiben lassen wie *Müll* auf einer Gracht.
Er musste raus, die Gedanken treiben lassen wie *Blätter* auf einer Gracht.

Phrase Replacement

Horkheimers Gehirn schaltete in den Overdrive.
Horkheimers Gehirn schaltete ein paar Gänge höher.

Sentences Compression

Die zweite Begegnung war *kein Zufall mehr. Es war* der Beginn einer Choreografie.
Die nächste Begegnung war der Beginn einer Choreografie.

Rephrasing with a New Metaphor

Landauers Beliebtheit *konnte nicht tiefer sinken; sie hatte bereits den Erdkern erreicht und ...*
Landauers Beliebtheit *wurde nur noch von der prominenter Naziverbrecher unterboten.*

Metaphor Replacement

War dieses rhetorisch geschliffene Ideen-Marketing wirklich die Messlatte für Forschung?
War dieser ganze Konferenz-Tourismus, dieser Wissenschafts-Jetset, wirklich die Benchmark für Qualität?