# Overcoming Copyright Barriers in Corpus Distribution Through Non-Reversible Hashing

Anonymous ACL submission

## Abstract

While annotated corpora are crucial in the field of natural language processing (NLP), those containing copyrighted material are difficult to exchange among researchers. Yet, such corpora are necessary to fully represent the diversity of data found in the wild in the context of NLP tasks. We tackle this issue by proposing a method to lawfully share the annotations of any sequential copyrighted corpus. The corpus creator shares the annotations in clear, along with a *non-reversible hashed* version of the source material. The corpus user must own the source material, and apply the same hash function to their own tokens, in order to match them to the shared annotations. Crucially, our method is robust to reasonable divergences in the version of the copyrighted data owned by the user. As an illustration, we present alignment experiments on different editions of novels. Our results show that our method is able to align correctly almost all of the shared corpus. We publicly release `novelties-bookshare`, a Python implementation of our method.

## 1 Introduction

Corpora, and in particular annotated corpora, are one of the central resources in the modern natural language processing (NLP) landscape. For many computational tasks, they are critical to train models, evaluate them and compare them against each other. Yet, while their importance is clearly established, researchers are often restricted to using freely available data released under a permissive license or in the public domain. This not only limits the amount of data available to the NLP community, but also introduces a systematic bias in studies since the performance of systems on copyrighted works is often not considered. For example, in the case of literary texts, recent works are protected, meaning most researchers restrict themselves to classical 19th plays and novels that fall in the public domain (Bamman et al., 2019; Han et al., 2021), with possible generalization issues (Lazaridou et al., 2021; Beelen et al., 2022).

Ideally, a corpus creator should be able to transmit the annotations of a corpus to a user, provided the latter is also in possession of the original material. A naive strategy could be to share the corpus annotations along with a way for the user to verify that their data is exactly identical to the creator's. The user could then use their data jointly with the shared annotations, and copyright would be enforced. However, in practice, the user's data is rarely exactly identical to the creator's. For instance, they can differ in terms of how the digitization process was carried out, or how the raw data was prepared to the required format for NLP experiments. Therefore, we need an annotations sharing scheme that is robust enough to handle reasonable divergences in the user's material while providing copyright compliance guarantees.

Motivated by this issue and inspired by a previous attempt by Bost et al. (2020), we propose a method to easily share a corpus under copyright constraints, in the case where the user possesses a sequence that is reasonably close to the creator's. We place ourselves in the situation where the corpus to share is composed of a sequence of tokens, and one or more annotations for each token. The named entity recognition (NER) task is a good example of such situation, where each token is annotated with a tag. We hash each token of the creator's corpus with a non-reversible cryptographic function, and shorten each resulting code to voluntarily create collisions and avoid attacks based on precomputed hash tables. On the user's side, each token is hashed using the same method. We robustly match the creator's and user's truncated hashes, which allows us to align annotations with user's tokens. Since the two corpora may be marginally different, it is likely that some tokens remain unaligned during this first stage. Therefore, we propose additional strategies to align some of these

remaining tokens. We present this overall process in Figure 1.

To validate the effectiveness of our sharing technique, we carry multiple experiments on a corpus of three novels, each one coming in three different editions. We empirically show that our method can accurately align almost all the hashed tokens, even when the user owns a different edition from the creator. Furthermore, our experiments validate the interest of our additional alignment strategies, that can be used to increase the number of correctly aligned tokens.

We release all the source code and data needed to reproduce our experiments under a free license[1]. Additionally, we publicly release *novelties-bookshare*, an implementation of our alignment method that can be used to share the annotations of any sequential copyrighted material.

## 2 Related Work

### 2.1 NLP Corpus Sharing

The most popular contemporary large language models come from the industry and rely on extremely large collections of textual data drawn from a wide range of sources, including books, journalistic content, and other works protected by intellectual property rights (Henderson et al., 2023). This shows the importance of such copyrighted material in tackling a variety of NLP tasks. Because these corpora are often assembled through large-scale Web crawling and aggregation, they frequently contain protected material for which no direct licensing or authorization has been obtained, prompting ongoing debates about the legal status of such practices and the lack of transparency surrounding training data composition (Buick, 2024). Academic researchers typically do not have access to the same legal and financial resources as these large firms, and therefore adopt various strategies to work around copyright constraints.

The first option is to constitute corpora based only on *public domain* data. This is for example the case of the well-known literary corpus Litbank (Bamman et al., 2019) and its French equivalent fr-Litbank (Mélanie-Becquet et al., 2024), whose most recent novels are from 1922 and 1937 respectively. Similarly, the speaker attribution corpora QuoteLi3 (Muzny et al., 2017) and PDNC (Vishnubhotla et al., 2022) focus only on classic texts, and the coreference resolution corpus FantasyCoref (Han et al., 2021) only includes *Alice in Wonderland* and public domain fairy tales. This reliance on older, public domain texts in such corpora is a fundamental problem when using them to train NLP models. Since these models are often used on contemporary texts, training on older data systematically biases model evaluation and hinders their performance (Lazaridou et al., 2021; Beelen et al., 2022). Additionally, public domain texts are often part of the training data of large models, which further increase evaluation bias concerns due to data contamination (Johnson et al., 2024).

The second strategy to avoid legal problems is simply to not share at all any corpus that includes protected material. For example, van de Camp and van den Bosch (2012) create a corpus containing biographies annotated for the identification and classification of personal relationships, but do not share this copyrighted material. Chun et al. (2025) extract character networks from Korean dramas, but completely anonymize their data to the point of not providing even the titles of the considered works. Of course, this practice is not on par with modern NLP standard, as it hinders reproducibility and the ability of researchers to draw comparisons.

The third approach to limit copyright infringement issues is to use only *excerpts* of the original texts when building a corpus. The rationale here is that this practice, as it concerns academic research, could fall under fair use. For instance, Dekker et al. (2019) propose a corpus of contemporary novels annotated for NER, and explicitly state that they only share the first chapters of these novels because of copyright concerns. Under Chinese law, Zhao et al. (2025) can use up to 10 chapters by novel to constitute their GenWebNovel NER corpus. This practice illustrates the legal ambiguity faced by academic researchers, as publicly sharing annotated excerpts of copyrighted literary works goes beyond what is clearly permitted under standard research exceptions in many jurisdictions.

Fourth, it is lawfully possible to share full copyrighted texts, provided their authors agree. Some researchers therefore explicitly ask for permission. For instance, Johnson et al. (2024) obtain the consent of each individual author to create *FicSim*, their corpus of fan-fictions. Alrashid and Gaizauskas (2023) ask an author permission to include his kid story in ScANT, their corpus of scene-annotated narrative texts. While this is an ethically sound way of releasing data, it is unre-

[1] https://anonymous.4open.science/r/novelties-bookshare-2F33/

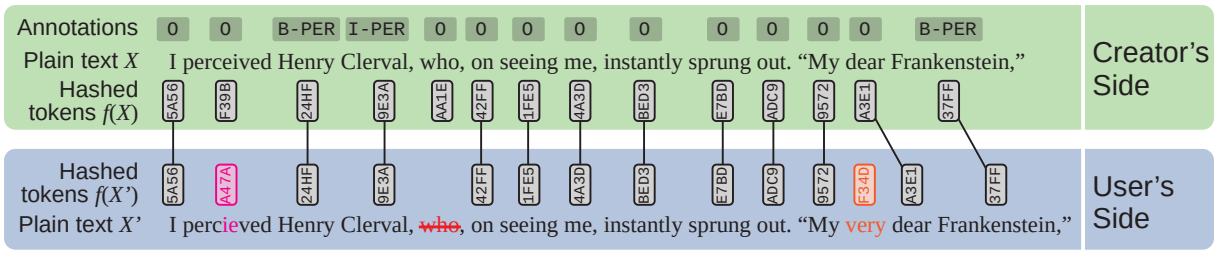


Figure 1: Proposed hashing scheme. Top part: the corpus creator's plain text ($X$) and hashed tokens $f(X)$, and corresponding annotations. Bottom part: the corpus user's plain text ($X'$) and hashed tokens ($f(X')$. Our method matches both sequences of hashes, while supporting small differences such as typos (shown in purple), missing tokens (red) and additional tokens (orange). The user never accesses the creator's plain text.

alistic to rely on such a costly method, even to constitute relatively small corpora.

Finally, the fifth solution to the issue of sharing annotated copyrighted material is proposed by Bost et al. (2020). In order to share the copyrighted TV series dialogues constituting their *Serial Speakers* corpus, they design an hash-alignment scheme that allows any user possessing the original dialogues to access the annotations. This approach is the only one allowing to share annotations of full copyrighted texts in a lawful way, as it does not share the original text but a *non-reversible hashed version*. We discuss this legal aspect further in Appendix A. We expand on this method by studying the effects of its parameters, generalizing it to any sequence of annotated tokens, and proposing additional strategies to improve alignment performance.

### 2.2 Related Problems

While this issue of copyrighted corpus sharing is not explicitly identified in NLP except by Bost et al. (2020), it resembles other problems from different fields. *Sequence reconstruction* is a class of problems introduced by Levenshtein (2001a,b) where the goal is to reconstruct a sequence from a set of partial and sometimes noisy observations (Wei et al., 2024). The *trace reconstruction problem* (Batu et al., 2004) is also related: in that setup, a binary string is passed through a deletion channel that may delete each of its bits with a set probability $p$, resulting in a shorter string called a *trace*. The goal is to determine how many of these traces are needed to reconstruct the original sequence with a high probability. Both of these tasks differ from ours: the user does not have access to a hashed non-noisy version of the original sequence, both problems assume the multiplicity of observed distorted sequences and the noise observed in our setup is specific to our application. Similar problems also arise in the context of DNA sequencing, where genomic data consisting of DNA sequences must be communicated and used securely in public cloud environments (Mete et al., 2015; Lu et al., 2021). For *read mapping*, a fundamental operation consisting in aligning a set of short DNA sequences to a reference genome, an existing strategy consists in sending only hashed sequences with non-trusted environments to perform part of the computation (Chen et al., 2012; Kang et al., 2016). This differs from our setting as we need to align freely shareable annotations to the user's version of the corpus.

## 3 Methods

### 3.1 General Principle

Let $X$ be a sequence $(x_1, \ldots, x_n)$ of copyrighted tokens, annotated by a *corpus creator*. Each token in $X$ can have one or more annotations, that correspond to task-specific labels. These annotations are not copyrighted, and consist of one or more different sequences of the same length as $X$, that can be freely shared. The creator wants to share these annotations with a *corpus user*, but without making the tokens $X$ public, as they are copyrighted. For this purpose, we hash them in a non-reversible way, producing a new sequence $f(X) = (f(x_1), \ldots, f(x_n))$ which is shared along with the annotations. This is illustrated by the top part of Figure 1 (green block). In this example, we consider the NER task under the BIO annotation scheme (Ramshaw and Marcus, 1995). As can be seen in the figure, each token of the corpus has a tag attached that indicates whether or not this token is part of an entity.

On their side, the user must possess the tokens too, in order to match them to the annotations shared by the creator. However, it is very unlikely

that the user has access to the *exact* same sequence $X$ as the creator. In the case of novels, for instance, turning a text into a token sequence usable for NLP experiments is a multi-step process involving many parameters. First, the user might have a different version of the novel, including editorial differences such as corrections, revisions, or a modernized text. Second, the process of digitization necessary to obtain an electronic book relies on some technical choices that can vary depending on place, time, and publisher: punctuation and typographic conventions, decomposition in chapters, text encoding. This step may even require error-prone steps such as optical character recognition. Finally, extracting a token sequence from the electronic book also necessitates to make methodological choices that can differ from the creator's, especially regarding text tokenization.

For all these reasons, it is reasonable to assume that the user possesses a slightly different token sequence, noted $X' = (x'_1, \ldots, x'_m)$. In order to get the annotations associated to these tokens, the user must first align their tokens with the creator's. For this purpose, we use the same hashing method as the creator, to produce a new sequence $f(X') = (f(x'_1), \ldots, f(x'_n))$. This part is represented in the bottom part of Figure 1 (blue block). The alignment is then performed only by comparing the hashes.

At this stage, it is crucial to stress two essential methodological points that establish the lawfulness of our method as discussed in Section 2.1 and Appendix A. First, **the creator's plain text is *never* shared with the user**: only the hashed tokens and their corresponding annotations are. Second, **our method does not try to decrypt the tokens** to obtain creator's plain text: it works only with the user's plain tokens, which must therefore be as similar as possible to the creator's.

### 3.2 Hashing

To hash the copyrighted sequence $X$, we process each token using the SHA-256 cryptographic function, resulting in hashed sequence $f(X)$. We pick SHA-256 for its wide availability, usage and its known robustness to inversion attempts.

Since an attacker can easily acquire a precomputed hash table with every word from a dictionary for a specific language, it would be trivial to break such a naive scheme. Therefore, we truncate each hash to forcibly create some collisions: thus, even if the attacker computes the hash of each possible word in the vocabulary, this only allows them to narrow down the set of possible tokens for a given hash to a set of words, but they have no way of knowing which of these words is the correct one.

On the user's side, we proceed similarly and apply the same hash function to the plain tokens $X'$ in order to produce hashed sequence $f(X')$.

### 3.3 Naive Alignment

The next step consists in applying any alignment algorithm between $f(X)$ and $f(X')$, through exact matching. As a consequence, when two hashes $f(x_i)$ and $f(x'_j)$ are matched, they are necessarily equal. In this case, we assume that $x_i = x'_j$, allowing us to determine which annotation is associated to $x'_i$. It is worth stressing that this assumption is not guaranteed to be correct, since we use truncated hashes that lead to collisions. In practice though, we find that such alignment method is sufficiently robust, since the context tokens disambiguate these collisions as we show in Section 4.3.

### 3.4 Additional Alignment Strategies

Due to the differences between $X$ and $X'$, it is likely that the above naive alignment method will not be able to align all the hashes. We identify three distinct situations. In the case of an **addition**, the user provides superfluous hashes that do not correspond to any hashes in the creator's sequence. As an example, this is the case of the orange token *"very"* in Figure 1. In the case of a **deletion**, the user does not provide one or more hashes, like the missing red token *"who"* in Figure 1. Finally, a **substitution** occurs when the user provides certain hashes that should be aligned with some of the creator's hashes, but their values are different. In Figure 1, there is a typo in the token *"perceived"* on the user's side, resulting in the token *"percieved"* (in purple) causing the substitution.

In the addition case, we can safely discard the superfluous user's tokens, as there are no corresponding annotations in the creator's sequence. The deletion and substitution cases are more challenging, which is why we handle them through additional strategies aimed at being applied *after* the initial naive alignment.

**`propagate` strategy** When a creator's hash $f(x_i)$ cannot be matched due to a missing or substituted user's hash, it is possible that other tokens $x_j$ with the same hash $f(x_j) = f(x_i)$ were already aligned with some user's hashes at different points in the sequence. In that case, we proceed to a vote and set

the token at position $i$ as the majority token in the user's sequence. We thereby "propagate" decisions made at the previous stage to hashes still pending alignment. Figure 2 shows an example of applying the `propagate` strategy on a deletion.

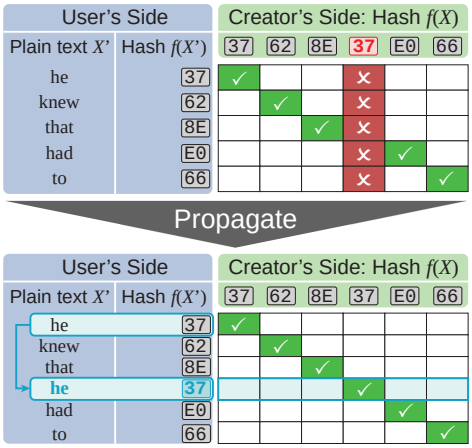


Figure 2: An example of applying the `propagate` strategy to retrieve a token missing on the user's side (shown in red), by leveraging the fact that the same hash value (37) is matched in another location (shown in cyan). Note that the creator's plain text $X$ is not available at the alignment stage.

**`retokenize` strategy** In the case of a substitution, there is a possibility that the creator's tokens underwent a different tokenization compared to the user's. Figure 3 shows such an example, where the creator's tokens *"runner"* and *"up"* correspond to the user's token *"runner-up"*. To resolve this case, we iterate through all possible splits of the user's token, hash them and compare them to the creator's hashes. If they match, we keep this split in the user's sequence $X'$. In the reverse case where the user's tokens have been incorrectly split, we merge them and compare the subsequent hash to the aligned creator's hash.

**`case` strategy** Certain substitutions are due to a different casing between the creator's and user's tokens $x_i$ and $x'_j$. To handle this situation, we try different casing options $c()$ for $x'_j$ and recompute its hash. If $f(c(x'_j))$ matches $f(x_i)$, we align $x_i$ with $c(x'_j)$ in the user's sequence.

**`mlm` strategy** When a token is missing in the user's sequence (either because of a deletion or a substitution), we leverage the textual context available on the user's side and try to estimate this token using masked language modeling (MLM). We insert a `[MASK]` token at the concerned position in $X'$, and use a pretrained model to predict the most likely word. If its hash matches the creator's hash $f(x_i)$, we keep the word in $X'$.

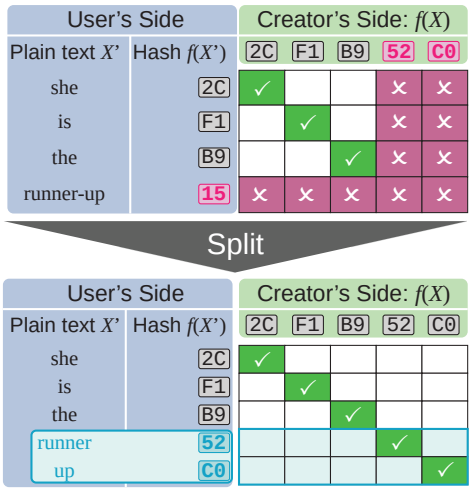


Figure 3: An example of applying the `retokenize` strategy on a substitution case (shown in purple). The corpus creator and user perform a different tokenization, resulting in tokens *"runner"* and *"up"* vs. *"runner-up"*, respectively. By testing all possible splits of the user's token, the strategy is able to recover from this error (in cyan).

**`pipe` meta-strategy** We also combine the above strategies within the `pipe` meta-strategy, where we sequentially apply them in a predefined sequential order to try and fix multiple classes of sequence differences.

The above strategies are language-agnostic, except for `case`, which only makes sense for scripts with a concept of letter case, such as the Latin script; and `mlm`, which necessitates to have a trained model that supports the language of interest.

It is important to stress that these strategies do not aim at aligning the user's text at any cost. If this text is too different from the creator's, the method *should* fail: trying to reconstruct the creator's text with MLM would go against international copyright laws. Our proposed strategies implement a tradeoff between, on the one hand, robustness to minor differences in the user's text, and, on the other hand, being copyright-compliant.

## 4 Experiments

We now perform alignment experiments on real novels to validate the effectiveness of our method.

### 4.1 Corpus

Our corpus is based on three public domain novels: Mary Shelley's *Frankenstein*, Herman Melville's *Moby Dick* and Jane Austen's *Pride and Prejudice*. For each one, we gather three editions. We consider the earliest one as the *creator's edition*, used to produce the annotated token sequence of the corpus creator, while both remaining editions are alternative *user's editions*. Among the latter, one is commonly considered as *close* to the creator's edition, whereas the other is more *distant*. We hypothesize that a closer edition should allow for better alignment performance. We provide more information on the exact sources we use for each edition in Appendix B.

***Frankenstein*** The first edition was first published in 1818 (`F-1818`, creator's). A later 1823 edition (`F-1823`, close) came with minor changes, and remains close to the original. Meanwhile, the 1831 edition (`F-1831`, distant) is a version of the novel revised by its author, with many significant differences: for example, the original first chapter was expanded and split in two.

***Moby Dick*** This novel was originally published in 1851 both in the USA (`MD-1851-US`, creator's) and the UK (`MD-1851-UK`, distant). These editions differ significantly from one another though, as the UK edition was censored and modified heavily and independently by its editor, which led for example to the removal of a few chapters. Finally, we also include the 1988 Northwestern-Newberry edition (*MD-1988*, close), which is closer to the original US edition.

***Pride and Prejudice*** By contrast, this novel had a simpler editorial life, its later editions only differing in small changes such as modernized spelling. The first edition came in 1813 (`PP-1813`, creator's), and the second one in 1817 (`PP-1817`, close). We also include the later 1894 illustrated edition (`PP-1894`, distant).

### 4.2 Setup

In all of our experiments, we use the enhanced version of the gestalt pattern matching alignment algorithm (Ratcliff and Metzener, 1988) implemented by `difflib` in the Python standard library. Since the algorithm is quadratic in time for the worst case scenario, we optimize its runtime by aligning novels in our corpus chapter by chapter. We deem this optimization acceptable as it is unlikely that tokens should be aligned across chapters. We only apply this optimization when the number of chapters is the same between the creator's and user's novels, otherwise we align directly on the entire content (`F-1831`, `MD-1851-UK`).

We use the `ModernBERT-base` model (Warner et al., 2025) for the `mlm` strategy, with a window size of 32 (see Appendix E.1 for details). For the `pipe` meta-strategy, we use the sequence `retokenize`, `mlm`, `case`, `propagate`. This order prioritizes high-precision strategies (see Appendix E.2 for details).

### 4.3 Effect of Truncated Hash Length

We first study the effect of the length of our truncated hash on performance and security. For each novel, we hash the creator's, close and distant editions, and align the resulting hashes with our proposed method. Lowering the length of the hash creates more collisions, improving security but also increasing the risk of errors as aligning tokens is more difficult and some alignment strategies may be impacted.

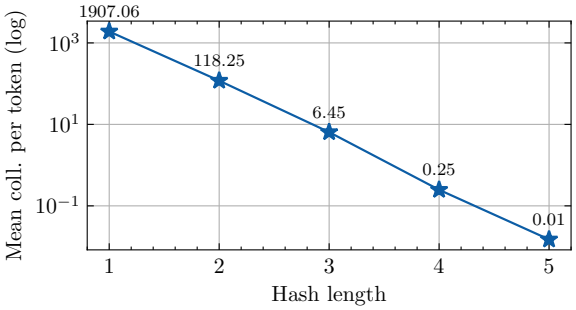


Figure 4: Mean number of hash collisions per token in our experimental corpus. We exclude hash lengths higher than 5 since they are too close to 0.

Figure 4 shows the mean number of collisions per token in our corpus depending on hash length. Given this figure, values 1 (1907.06 collisions per token), 2 (118.25) or 3 (6.45) appear as appropriate candidates. For longer hashes, the number of collisions is close to 0, which we deem not secure enough.

To choose the best hash length, we have to observe its impact on alignment performance. We do so by performing our experiment with hash lengths in $\{1, 2, 3, 4, 64\}$. Figure 5 shows the mean percentage of token alignment errors across editions. Decreasing the hash length increases the number of errors for all strategies, highlighting the necessary tradeoff between security and alignment reliability.

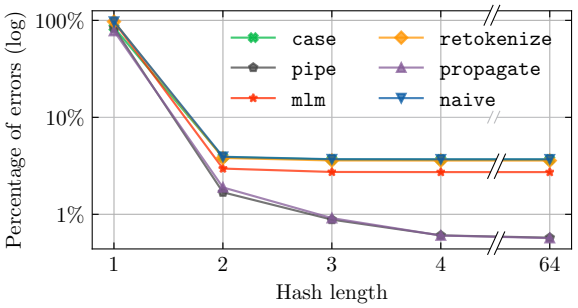


Figure 5: Mean percentage of errors across editions as a function of hash length ({1, 2, 3, 4, 64}) for different alignment strategies

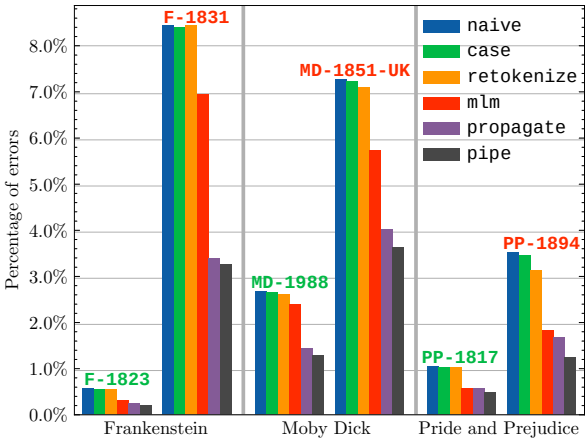


Figure 6: Percentage of misaligned tokens depending on the strategy and user's edition. For each novel, the user's edition which is the closest to the creator's is shown on the left (green name), whereas the most distant edition is on the right (red name).

Additionally, we also observe that some alignment strategies are more sensitive to hash length than others. The `propagate` strategy is particularly vulnerable to truncation, as it may propagate errors rather than correctly align tokens. On the other end of the spectrum, the `case` and `mlm` strategies are less sensitive. For the rest of the experiments, we present results with a hash length of 2 as a tradeoff between security and alignment performance.

### 4.4 Results Per Edition

In the ideal case, we would expect the user to own exactly the same plain text as the creator, i.e. the same edition of the novel. However, here we consider a more difficult situation, where the user owns a different edition (close, distant) compared to the creator. We compare the impact of our proposed alignment strategies by recording the number of incorrectly aligned tokens, and plot our results in Figure 6. As a practical illustration, we also apply our alignment method to NER in Appendix D, where it is able to align 96.48% of entities.

Overall, we observe that all of our alignment strategies successfully reduce the original number of errors compared to the naive alignment. The `case` and `retokenize` strategies are the least effective. Meanwhile, the `mlm` and `propagate` strategies obtain better results. Meta-strategy `pipe` bests all of the singular strategies, confirming the interest of combining them.

As we hypothesized earlier, the alignment performance strongly depends on the proximity of the user's edition to the original text. Using the close editions and the best strategy results in a percentage of errors that does not exceed 1.3%. Meanwhile, using the reworked 1831 edition of *Frankenstein*, the censored UK edition of *Moby Dick* or the modernized 1894 edition of *Pride and Prejudice* yield substantially more errors. These results emphasize the need for the user to have a version of the data that is as close as possible to the original text.

Not all strategies are equal when it comes to runtime, as we show in Appendix C. `mlm` and `pipe` in particular are costly, and can bring the alignment time up to the hour in the worst cases. Other strategies most often stay under a 10 seconds runtime.

### 4.5 Synthetic Errors

To better understand the effect of different possible degradations in the user's sequence and the impact of our additional alignment strategies, we add synthetic errors to the creator's edition of our corpus, creating a new synthetic user's edition, and to measure the impact it has on the performance of our alignment system. We experiment with six types of synthetic errors. The **`add`** error type samples tokens from a dictionary, using their frequency in the considered novel, and adds them to a uniformly sampled position in the text. The **`substitute`** type samples tokens as in add, but replaces them by others instead of adding new ones. The **`delete`** type removes uniformly sampled tokens. We simulate tokenization errors with **`token_split`**, splitting uniformly sampled tokens. Conversely, the **`token_merge`** error type merges uniformly sampled consecutive tokens. Finally, the **`ocr_scramble`** type simulates realistic OCR issues using the `scrambledtext` library (Bourne, 2025).

For all of these types of errors except `ocr_scramble`, we produce $l \times r$ errors, with $l$ the length of the text and $r$ a ratio between 0 and 1.

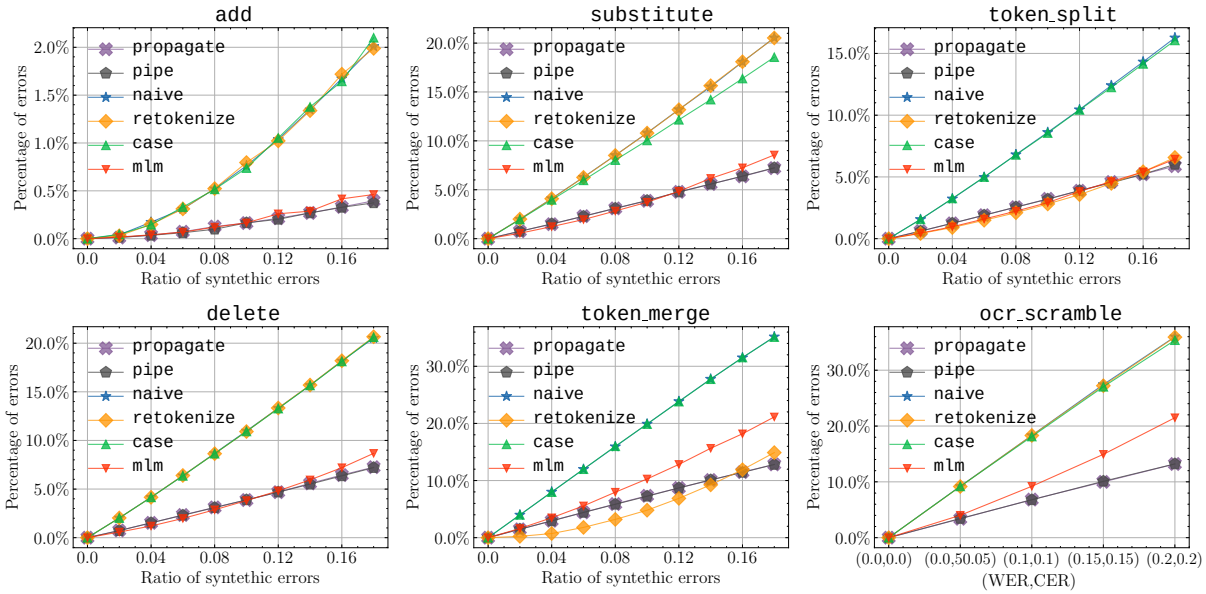


Figure 7: Percentage of alignment errors as a function of the ratio of synthetic errors added.

In practice, we consider error ratios from 0 to 0.1, since we find greater ratios unrealistic: counting additions, deletions and substitutions at the same time, we found the biggest ratio between novels of our corpus to be around 0.035. Regarding OCR errors, the `scrambledtext` library allows setting a target word error rate (WER) and a character error rate (CER). We survey the following (WER, CER) pairs: $\{(0,0), (0.05,0.05), (0.1,0.1), (0.15,0.15), (0.2,0.2)\}$, as examples shown by Bourne (2025) with larger values are corrupted beyond recognition.

Figure 7 shows our results, leading to several observations. First, we are always able to fully align annotations when the user's text is identical to the creator's. We also note that the `pipe` meta-strategy outperforms other strategies in most cases, highlighting the interest of combining them. The `retokenize` strategy is very effective in case of token splitting or merging, but its performance is very weak in other settings, making it a specialized strategy for tokenization issues. `mlm` and `propagate` appear to be the best performing single strategies. Finally, we observe that adding new tokens with `add` errors only has a very weak impact on the number of errors compared to other types of errors.

## 5 Conclusion

In this article, we presented a method to let a corpus creator legally share their annotations of some copyrighted material, provided the user of the corpus is in possession of this material. Section 4.5 shows that our method is always able to fully align annotations when the user's content is identical to the creator's. Furthermore, our experiments in Section 4.4 show that the alignment is successful even if the user owns a different (but sufficiently close) version of the material, as we reach a percentage of errors between 0.21% and 1.3% in that case. The number of errors, however, increases as the user's content diverges from the corpus creator's. This stems from our need to balance alignment performance and security, as an attacker with completely different data must not be able to access the corpus for our method to respect copyright. This also highlights the importance for the creator to provide to the user as much information as possible about the corpus (such as novel editions for literary text), to ease alignment.

As an illustration, we applied our alignment method to NER in Appendix D. Our sharing scheme is, however, applicable to any corpus with token level annotations. This covers many NLP tasks that can be formalized with that paradigm such as POS tagging, coreference resolution, chunking or slot filling. Annotations of corpora from other domains may also be shared through our technique as long as the task can be cut into token-like units. For example, it may be possible to share a bounding box detection dataset at the pixel level.

## 6 Limitations

The severity of alignment errors is task-dependent: certain errors may be more important depending on the application context. We present additional results on NER in Appendix D, but it is not feasible to study the impact of errors for all possible tasks.

Our alignment method is general enough to be applied to any sequential corpus. However, some of the additional alignment strategies we presented are specific to natural language corpora and cannot be applied directly to tasks from other domains, such as bounding box detection. The `retokenize` strategy, for example, relies on the necessity to tokenize text for NLP tasks. The `case` strategy is limited to natural language. Masked language modeling could be applied to other domains, but one needs a pretrained model to do so.

## References


T. Alrashid and R. Gaizauskas. 2023. Scant: A small corpus of scene-annotated narrative texts. In *6th Workshop on Narrative Extraction From Texts*, CEUR Workshop Proceeedings, pages 143–149.

A. Amalvy and V. Labatut. 2024. Annotation guidelines for corpus novelties: Part 1 — named entity recognition. Technical report, Avignon Université.

D. Bamman, S. Popat, and S. Shen. 2019. An annotated dataset of literary entities. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, volume 1, pages 2138–2144.

T. Batu, S. Kannan, S. Khanna, and A. McGregor. 2004. Reconstructing strings from random traces. In *Fifteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, page 910–918.

K. Beelen, J. Lawrence, D. C. S. Wilson, and D. Beavan. 2022. Bias and representativeness in digitized newspaper collections: Introducing the environmental scan. *Digital Scholarship in the Humanities*, 38(1):1–22.

X. Bost, V. Labatut, and G. Linares. 2020. Serial speakers: a dataset of TV series. In *Twelfth Language Resources and Evaluation Conference*, pages 4256–4264.

J. Bourne. 2025. Scrambled text: fine-tuning language models for ocr error correction using synthetic data. *International Journal on Document Analysis and Recognition*.

A. Buick. 2024. Copyright and ai training data–transparency to the rescue? *Journal of Intellectual Property Law and Practice*, 20(3):182–192.

Y. Chen, B. Peng, X. Wang, and H. Tang. 2012. Large-scale privacy-preserving mapping of human genomic sequences on hybrid clouds. In *Network and Distributed System Security Symposium*.

Ye Eun Chun, Taeyoon Hwang, Seung-won Hwang, and Byung-Hak Kim. 2025. CREFT: Sequential multi-agent llm for character relation extraction. *arXiv*, cs.CL:2505.24553.

N. Dekker, T. Kuhn, and M. van Erp. 2019. Evaluating named entity recognition tools for extracting social networks from novels. *PeerJ Computer Science*, 5:e189.

S. Han, S. Seo, M. Kang, J. Kim, N. Choi, M. Song, and J. D. Choi. 2021. Fantasycoref: Coreference resolution on fantasy literature through omniscient writer's point of view. In *4th Workshop on Computational Models of Reference, Anaphora and Coreference*, pages 24–35.

P. Henderson, X. Li, D. Jurafsky, T. Hashimoto, M. A. Lemley, and P. Liang. 2023. Foundation models and fair use. *Journal of Machine Learning Research*, 24(400):1–79.

N. Johnson, A. Bertsch, and E. Strubell. 2024. Ficsim: An ethically constructed dataset for long-context semantic similarity comparison within fictionworkshop on creativity & generative ai. In *NeurIPS Workshop on Creativity & Generative AI*.

S. Kang, K. M. M. Aung, and B. Veeravalli. 2016. Towards secure and fast mapping of genomic sequences on public clouds. In *4th ACM International Workshop on Security in Cloud Computing*, page 59–66.

A. Lazaridou, A. Kuncoro, E. Gribovskaya, D. Agrawal, A. Liška, T. Terzi, M. Gimenez, C. de Masson d'Autume, T. Kocisky, S. Ruder, D. Yogatama, K. Cao, S. Young, and P. Blunsom. 2021. Mind the gap: assessing temporal generalization in neural language models. In *35th International Conference on Neural Information Processing Systems*, pages 29348–29363.

V. Levenshtein. 2001a. Efficient reconstruction of sequences. *IEEE Transactions on Information Theory*, 47(1):2–22.

V. Levenshtein. 2001b. Efficient reconstruction of sequences from their subsequences or supersequences. *Journal of Combinatorial Theory, Series A*, 93(2):310–332.

D. Lu, Y. Zhang, L. Zhang, H. Wang, W. Weng, L. Li, and H. Cai. 2021. Methods of privacy-preserving genomic sequencing data alignments. *Briefings in Bioinformatics*, 22(6).

F. Mélanie-Becquet, J. Barré, O. Seminck, C. Plancq, M. Naguib, M. Pastor, and T. Poibeau. 2024. BookNLP-fr, the French versant of BookNLP. a tailored pipeline for 19th and 20th century French literature. *Journal of Computational Literary Studies*, 3(1):1–34.

A. Mete, O. A., O. Bugra, and Ş. M. 2015. Privacy preserving processing of genomic data: A survey. *Journal of Biomedical Informatics*, 56:103–111.

G. Muzny, M. Fang, A. Chang, and D. Jurafsky. 2017. A two-stage sieve approach for quote attribution. In *15th Conference of the European Chapter of the Association for Computational Linguistics*, volume 1, pages 460–470.

L. Ramshaw and M. Marcus. 1995. Text chunking using transformation-based learning. In *Third Workshop on Very Large Corpora*.

J. W. Ratcliff and D. Metzener. 1988. Pattern matching: The gestalt approach. *Dr. Dobb's Journal*, July 1988:46.

M. van de Camp and A. van den Bosch. 2012. The socialist network. *Decision Support Systems*, 53(4):761–769.

K. Vishnubhotla, A. Hammond, and G. Hirst. 2022. The project dialogism novel corpus: A dataset for quotation attribution in literary texts. In *13th Language Resources and Evaluation Conference*, pages 5838–5848.

B. Warner, A. Chaffin, B. Clavié, O. Weller, O. Hallström, S. Taghadouini, A. Gallagher, R. Biswas, F. Ladhak, T. Aarsen, G. T. Adams, J. Howard, and I. Poli. 2025. Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference. In *63rd Annual Meeting of the Association for Computational Linguistics*, pages 2526–2547.

H. Wei, M. Schwartz, and G. Ge. 2024. Reconstruction from noisy substrings. *IEEE Transactions on Information Theory*, 70(11):7757–7776.

H. Wei, Y. Sun, and Y. Li. 2025. Deepseek-ocr: Contexts optical compression. *arXiv*, cs.CV:2510.18234.

H. Zhao, Y. Yan, S. Zhu, H. Liu, Y. Jia, H. Zan, and M. Peng. 2025. Genwebnovel: A genre-oriented corpus of entities in chinese web novels. In *31st International Conference on Computational Linguistics*, pages 3836–3849.

## A Legal Aspects

**Legal Framework.** The international baseline for the protection of copyrighted literary works is set by the *Berne Convention for the Protection of Literary and Artistic Works*[2]. In particular, it forbids their unauthorized reproduction, in whole or in part (Article 9), and their communication to the public (Articles 11 & 11bis).

Under directive *2001/29/EC*[3], European Union law forbids the direct or indirect unauthorized reproduction of copyrighted works or substantial parts thereof (Article 2), and their unauthorized communication to the public (Article 3). Furthermore, the Court of Justice of the European Union (CJEU) emphasizes that any use containing recognizable expression is prohibited (*Infopaq C-5/08*[4], *Pelham C-476/17* [5]). US copyright law (Copyright Act, Title 17 U.S.C.[6] also forbids reproducing or copying these works (§106(1)) or distributing them (§106(3), §106(4)) without authorization.

In the UK, the *Copyright, Designs and Patents Act*[7] (CDPA) similarly forbids copying or reproducing these works or substantial parts (ss. 16–17 CDPA), and issuing copies to the public (s.18 CDPA). Other common law countries and major jurisdictions apply the same rules. As a consequence, sharing the original literary text, even partially and even digitally, is in principle forbidden. Whether sharing *excerpts* of literary works for research purpose (as done by (Dekker et al., 2019; Zhao et al., 2025)) constitutes fair use is another debate, which we do not discuss here: we are only interested in sharing *fully* annotated works.

**Sharing Plain Annotations.** Our method does not involve sharing *directly* any copyrighted material, but 1) a hashed version of the original text, and 2) the associated annotations authored by the researchers creating the corpus. These annotations are shared in plain text, but they are technical data, and not part of the original literary work. Put differently, they are distinct from the expressive content of the literary work (i.e. its actual words, narrative voice, the author's stylistic choices). As such, they are not protected by the laws mentioned before.

In the EU, directive *(EU) 2019/790*[8] states that analytical outputs and metadata are legally distinct from the protected works themselves, and explicitly permits research mining for non-expressive results (Articles 3 & 4), even of copyrighted works, provided outputs are non-substitutive (i.e. the original text cannot be recovered based on these data). In the USA, courts recognize that uses which do not

[2] https://www.wipo.int/wipolex/en/text/283698

[3] https://eur-lex.europa.eu/eli/dir/2001/29/oj/eng

[4] https://ipcuria.eu/case?reference=C-5/08

[5] https://ipcuria.eu/case?reference=C-476/17

[6] https://www.copyright.gov/title17/

[7] https://www.legislation.gov.uk/ukpga/1988/48/contents

[8] https://eur-lex.europa.eu/eli/dir/2019/790/oj/eng

communicate the expressive content and are transformative or functional can be fair use (17 U.S.C. §107). For instance, case *Authors Guild v. Google*[9], 804 F.3d 202 (2d Cir. 2015) concluded that Google Books' scanning for indexing and search was illustrative of non-expressive fair use.

In the UK, non-expressive computational uses of lawfully accessed works are permitted (ss.29A, 29B CDPA), including sharing outputs that do not contain readable or recognizable parts of the work. Other major jurisdictions implement similar rules regarding the technical data extracted from literary text.

**Sharing Hashed Tokens.** Let us now focus on the hashed tokens of the copyrighted text. The original text is never shared, reproduced, or communicated as part of the corpus. Instead, each token of the source text is transformed using a non-reversible cryptographic hashing function. Only these hashed identifiers (together with the linguistic annotations produced by the researchers), are disseminated. The resulting corpus contains no readable text, no identifiable excerpts, and no information allowing access to or reconstruction of the original works. The corpus is unusable without prior access to the original text: the user must independently possess the work, and has to locally apply the same hashing procedure in order to align the annotations with their own plain text. As a consequence, there is no reproduction of the protected text, no communication of the original work to the public, and no creation of derivative works.

In the Berne Convention, reproduction (Article 9), communication (Articles 11 & 11bis), and adaptation (Article 12) only apply to recognizable expression: token hashes do not convey the expression of the novel, they are non-recognizable technical identifiers. The same reasoning applies to EU law, as without possession of the original work, the hashes are useless. CJEU cases (Infopaq C-5/08, Pelham C-476/17) require that identifiable expression is reproduced for infringement: hashes do not meet this criterion. US and UK laws similarly do not consider hashes as reproductions or distribution of expressive content.

**Reliability of the Hashing.** Based on these observations, a question arises: what about the *reliability* of the hashing scheme? International copyright law does not require absolute, information-theoretic irreversibility, but rather focuses on whether the material shared by the corpus' creator objectively communicates expressive content or makes it reasonably accessible to the public. If reversing the hashes would require disproportionate technical effort, the corpus would still be treated as non-expressive under all major jurisdictions. As a consequence, there is no statutory minimum security level to be implemented in our hashing scheme, but there is a standard of practical non-reconstructability. This is the reason why our method lets the creator control the length of the hashes, which directly impacts the vulnerability of the hashing scheme to attacks.

The strategies that we propose in Section 3.4 to improve the robustness of our method against misalignment issues weaken this argument, though. Indeed, they could be used to facilitate the reconstruction of the original text without access to this text, at least in theory. However, we must stress that this is not feasible in practice, as these strategies are just ancillary mechanisms that require the user to have access to a text extremely similar to the original content. Strategy `propagate` has the strongest effect on misaligned hashes (cf. Figure 6), but it is limited to tokens that are already known by the user. Strategies `retokenize` and `case` have a very marginal effect, and require the user to have access to the original text, even if incorrectly tokenized or capitalized. Strategy `mlm` has a slightly stronger effect, but it provides no guarantee that the MLM-generated token is the same as in the original text. Moreover, it is efficient only if the textual context of the missing token is known by the user, which therefore still has to prove they have access to the original material. In practice, as shown by our experimental results, even when dealing with two distinct editions of the same novel, we obtain up to 8 % incorrect tokens (cf. Section 4.4). Applying our method from scratch, without possessing a very similar version of the original text, leads to some content very different from the targeted literary work. In conclusion, we think that our method minimizes ethical and legal risk related to copyright infringement, and is aligned with best practices for responsible data sharing in natural language processing research.

## B Corpus Details

Table 1 indicates where we obtained each novel edition used in our experiments. For all editions

[9] https://www.copyright.gov/fair-use/summaries/authorsguild-google-2dcir2015.pdf

of *Frankenstein*, we obtain the text through the Frankenstein Variorum website.

In the case of *Moby Dick*, we obtain the text of the U.S. edition through Wikisource. Since we were unable to find a digital edition of the original U.K. edition, we use Deepseek-OCR (Wei et al., 2025) to extract text from the book images hosted at the Melville Electronic Library.

For *Pride and Prejudice*, we use Wikisource for both the first (`PP-1813`) and second (`PP-1817`) editions in our experiments. We include the `PP-1894` edition through project Gutenberg. Since this version is illustrated, the raw project Gutenberg text contain descriptions of the included illustration: we manually remove these as a preprocessing step.

## C Alignment Runtime

In this section, we present more details on the runtime of alignment in our Section 4.4 inter-edition experiments. As can be seen in Figure 8, the `mlm` strategy is largely the most expensive, with a runtime that can go close to 3 hours for the UK edition of *Moby Dick*. The pipe strategy is close to the runtime of `mlm`, since it includes it. Meanwhile, other strategies do not break the 10 seconds barrier, except for the UK edition of *Moby Dick* where the runtime is comprised between 1 and 2 minutes. For this edition and the 1831 edition of *Frankenstein*, a large part of the runtime comes from the alignment itself as evidenced by the runtime of the naive strategy: since the number of chapters between these editions and their respective source editions is different, we cannot align them chapter-by-chapter as described in Section 4.2, leading to an increased alignment time.

## D Task-Specific Alignment Results

In the main text, we present results as a number of alignment errors. However, the severity of errors is task-dependent, as certain tokens may be more important as others. While it is not realistic to explore the impact of errors on every possible task, in this section we present additional results on NER.

To estimate the impact of alignment errors on NER, we use the NER-annotated version of *Moby Dick* from the Novelties corpus (Amalvy and Labatut, 2024) as the source version. We use the `MD-1988` edition of *Moby Dick*, as it is the closest from the Novelties version. We use a strict definition of the notion of error: if a single token from an entity was not alignment, we consider the entire

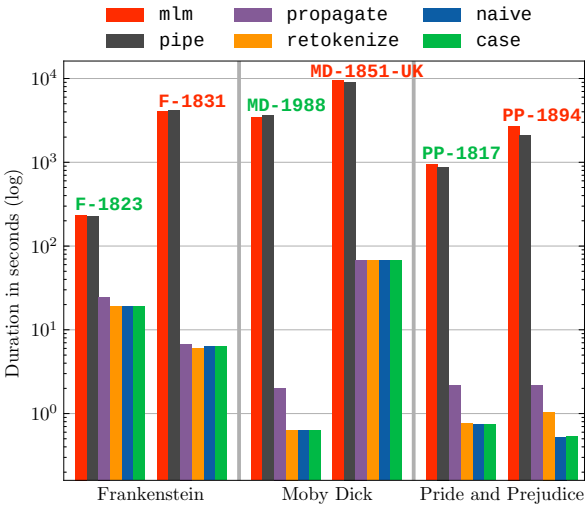


Figure 8: Duration of alignment depending on the strategy and user edition, in seconds.

entity as non-aligned. We obtain a percentage of errors of 0.82% with our best strategy `pipe`, indicating that we are able to recover most of the text. However, we note a percentage of errors on entities of 3.52%. While this is partially due to our strict definition of the notion of errors (a more lenient definition where the entire entity must be lost to be considered an error yields 2.93% of errors), it also highlights that entity tokens are harder to align.

## E Optimal Parameters of Alignment Strategies

### E.1 Masked Language Modeling Context Window Size

In this section, we study the impact of the context window size on the performance of `mlm`, our masked language modeling alignment strategy (cf. Section 3.4).

Figure 9 shows the performance of our `mlm` alignment strategy on our editions experiments from Section 4.4. We observe that a window size of 32 tokens generally better results for all but one edition. Beyond that, increasing window size seem to monotonically increase the number of errors.

### E.2 Order of Strategies in the `pipe` Meta-strategy

The position of each strategy in the `pipe` meta-strategy influences performance. When a strategy corrects an error, it will not be corrected further by the following strategies. Therefore, it is advantageous to place high precision strategies first in the pipeline. In the main text, we only report results

| Novel Edition | Source Type | Source Identifier |
|---|---|---|
| F-1818 | URL | https://github.com/FrankensteinVariorum/fv-data/blob/master/preliminary-edition-data/1818_full_prelim.xml |
| F-1823 | URL | https://github.com/FrankensteinVariorum/fv-data/blob/master/preliminary-edition-data/1823_full_prelim.xml |
| F-1831 | URL | https://github.com/FrankensteinVariorum/fv-data/blob/master/preliminary-edition-data/1831_full_prelim.xml |
| MD-1851-US | URL | https://en.wikisource.org/wiki/Moby-Dick_(1851)_US_edition |
| MD-1851-UK | URL | https://github.com/performant-software/mel-website/tree/master/images/md-british-v1<br>https://github.com/performant-software/mel-website/tree/master/images/md-british-v2<br>https://github.com/performant-software/mel-website/tree/master/images/md-british-v3 |
| MD-1988 | ISBN | 9780810102699 |
| PP-1813 | URL | https://en.wikisource.org/wiki/Pride_and_Prejudice_(1813) |
| PP-1817 | URL | https://en.wikisource.org/wiki/Pride_and_Prejudice_(1817) |
| PP-1894 | URL | https://www.gutenberg.org/ebooks/1342 |

Table 1: Source from which we obtained each edition of our novels (**F**rankenstein, **M**oby **D**ick, **P**ride and **P**rejudice).

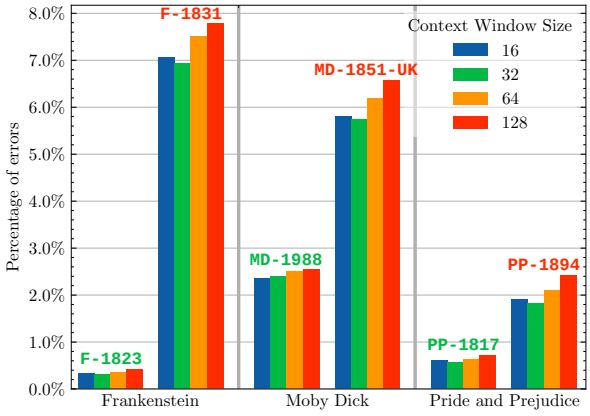


Figure 9: Percentage of errors using our `mlm` alignment strategy, depending on the context window size and user edition.

with the best order we found for the `pipe` strategy. Figure 10 shows the precision errors recorded during our edition experiments of Section 4.4 for each strategy. Interestingly, even though it is one of the best performing strategy, we notice that the propagate strategy has the lowest precision by far. By decreasing order of precision, we determine that the best ordering for the `pipe` strategy is `retokenize`, `mlm`, `case`, `propagate`.

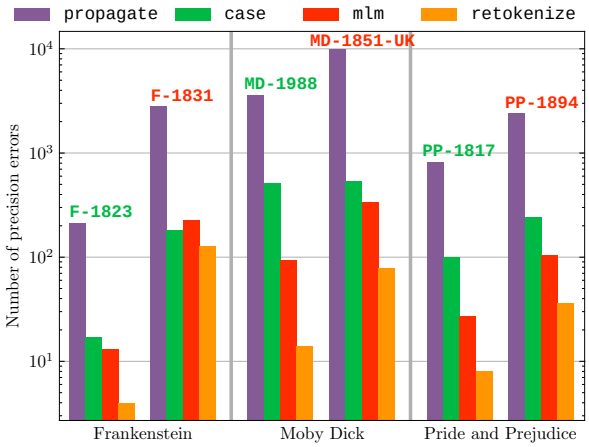


Figure 10: Number of precision errors using our alignment meta-strategy `pipe`, depending on the placement of the `propagate` strategy and user edition.