ABLIT: A Resource for Analyzing and Generating Abridged Versions of English Literature

Anonymous ACL submission

Abstract

Creating an abridged version of a text involves shortening it while maintaining its linguistic qualities. In this paper, we examine this task from an NLP perspective for the first time. We present a new resource, ABLIT, which is derived from abridged versions of English literature books. The dataset captures passage-level alignments between the original and abridged texts. We characterize the linguistic relations of these alignments, and create automated models to predict these relations as well as to generate abridgements for new texts. Our findings establish abridgement as a challenging task, motivating future resources and research.

1 Introduction

An abridgement is a shortened form of a text that maintains the linguistic qualities of that text. It is intended to make the original text faster and easier to read. In this paper, we propose abridgement as an NLP problem and describe its connection to existing inference and generation tasks. We present a novel dataset for this task, focused on abridged versions of English literature books, which we refer to as the ABLIT dataset. We demonstrate the characteristics of ABLIT in terms of the relations between original and abridged texts as well as the challenges of automatically modeling these relations. The dataset and all associated code are available at: github.com/withheld/during/blind/review.

2 The task of abridgement

We define abridgement as the task of making a text easier to understand while preserving as much of its content as possible. As such, abridgement intersects with tasks that fuse natural language inference (NLI) and natural language generation (NLG), in particular summarization and simplification.

Summarization condenses the main content of a text into a shorter version, with the purpose of making its main content easier to understand. Existing research has used the categories of *extractive* and *abstractive* to describe summaries. In the former, the summary 'extracts' sequences from the text, whereas in the latter the summary 'abstracts' out the meaning of the text and rewrites it. The degree of abstractiveness of a summary is indicated by the amount of novel text it contains that is not directly contained in the original. Like a summary, an abridgement is shorter than its original text, but it preserves more of its language and can be seen as an alternative version rather than a meta-description. According to how summaries are characterized, abridgements are highly extractive, even if some abstraction is needed to connect the extracted components. Some research has examined summarization of narratives, including literary text (Kazantseva, 2006; Mihalcea and Ceylan, 2007; Zhang et al., 2019). Of particular relevance to our work are datasets recently released by Chaudhury et al. (2019), Kryściński et al. (2021), and Ladhak et al. (2020), all of which consist of summaries of fiction books. The summaries in these datasets are significantly different from abridgements in that they are highly abstractive; they convey the book's narrative without preserving the linguistic properties of the text itself. In Kryściński et al., summaries are provided at different levels of granularity (book, chapter, and paragraph). Their analysis demonstrates that even the finer-grained summaries at the paragraph level are quite abstractive.

The task of simplification also aims to make a text easier to understand, but without significantly filtering its content. From this perspective, abridgement is quite similar to simplification. However, as with some types of summarization, simplification is not necessarily concerned with maintaining the linguistic form of a text, since surface changes may be crucial for promoting readability. In contrast, abridgement seeks a stronger balance between increasing a text's readability while maintaining its

writing style. Simplification is often evaluated with reference to single sentences isolated from their passage context (Sun et al., 2021). Alternatively, we examine abridgement with respect to multi-passage text. Research on simplification has been constrained by a lack of high-quality publicly available datasets. Existing datasets have been derived from sources like Wikipedia (e.g. Coster and Kauchak, 2011) and news articles (Xu et al., 2015), but none have focused on literary text.

3 Creating an abridgement dataset

The ABLIT dataset is derived from 10 classic English literature books, listed in A.3. These books are in the public domain and available through Project Gutenberg[1]. A single author, Emma Laybourn, wrote abridged versions of these books that are also freely available[2]. The author explains:

> "This is a collection of famous novels which have been shortened and slightly simplified for the general reader. These are not summaries; each is half to two-thirds of the original length. I've selected works that people often find daunting because of their density or complexity: the aim is to make them easier to read, while keeping the style intact. It's hoped they will also appeal to students of English who are not quite ready to tackle the originals."

Informed by this perspective, we designed ABLIT to capture the alignment between passages in a text and its abridged version. In this case we define *alignment* as a textual entailment relation (e.g. Dagan and Glickman, 2004). An abridged passage is aligned with an original passage if the meaning of the original entails that of the abridged.

After obtaining the original and abridged books from their respective sites, we split the books into chapters using manually defined pattern matching. A single instance in ABLIT consists of the original and abridged version of one chapter. Obviously, these versions already form a very broad alignment unit, but our goal was to discover finer-grained levels of alignment. We chose to use sentences as the minimal alignment units, since they are intuitive units of expression in text and can be detected automatically[3]. ABLIT annotates sentence boundaries by indexing their position in the text, which enables all whitespace characters (most importantly, line breaks marking paragraphs) to be preserved.

3.1 Automated alignments

We pursued an automated approach to establish initial alignments between the original and abridged sentences for each chapter. It follows the same dynamic programming scheme used to create the Wikipedia Simplification dataset (Coster and Kauchak, 2011). We refer to a group of adjacent sentences in a text as a *span*. We define the length of a span by the number of sentences it contains. Each span o of length o_n in the original version of a chapter is paired with a span a of length a_m in the abridged version. The value of a_m can be zero, allowing for the possibility that an original sentence is aligned with an empty string. Based on a review of the assessment set described below in Section 3.2, we made the assumption that in the resulting sequence of aligned pairs, the positions of o and a in their corresponding texts will always succeed the respective positions of the previous pairs in the sequence (i.e. no criss-crossing alignments). For each pair of o and a, we score the likelihood that they should be aligned. This score is based on a similarity metric $sim(o, a)$ indicating the degree to which o entails a. Additionally, the scoring function considers the length of the spans in order to optimize for selecting the narrowest alignment between the original and abridged text. For instance, if a one-to-one alignment exists such that the meaning of a single sentence in the abridgement is fully entailed by a single original sentence, these sentences should form an exclusive alignment. To promote this, we adjust $sim(o, a)$ by a penalty factor pn applied to the size of the pair, where $size = \max(o_n, a_m)$. Ultimately, the alignment score for a given span pair (o, a) is: $\max(0, sim(o, a) - ((size - 1) * pn))$. At each sentence position in the original and abridged chapters, we score spans of all lengths $[1, o_n]$ and $[0, a_m]$, then select the one that obtains the highest score when its value is combined with the accumulated score of the aligned spans prior to that position. Once all span pairs are scored, we follow the backtrace from the highest-scoring span in the final sentence position to retrieve the optimal pairs

[1] gutenberg.org
[2] englishliteratureebooks.com
[3] We used nltk.org for all sentence segmentation and word tokenization. For analyses pertaining to words, words are lowercased without any other normalization (e.g. lemmatization).

for the chapter. Below we refer to each resulting span pair (o, a) in this list as an alignment *row*.

3.2 Assessment of automated alignments

We applied this automated alignment approach to the first chapter in each of the ten books in ABLIT, which we designated as an *assessment* set for investigating the quality of the output rows. We instantiated $sim(o, a)$ as the ROUGE-1 (unigram) precision score[4] between the spans, where a is treated as the hypothesis and o is treated as the reference. Here we refer to this score as $R\text{-}1_p$. It effectively counts the proportion of words in a that also appear in o. Using qualitative judgment of a sample of output rows, we performed a grid-search optimization of o_n in $[1, 6]$ and a_m in $[0, 6]$ and selected $o_n = 3$ and $a_m = 5$. We similarly optimized pn values in $[0, 0.25]$ and selected pn = 0.175. Smaller values of pn yielded rows that were not minimally sized (i.e. they should have been further split into multiple rows), whereas larger values tended to wrongly exclude sentences from rows. The resulting output consisted of 1,126 rows, which were then reviewed and corrected by five human validators recruited from our internal team. A.2 describes the interface for this task. We found that inter-rater agreement was very high (Cohen's κ = 0.983) and the few disagreements were easily resolved through discussion to reach a consensus. The validators reported spending about 10-15 minutes on each chapter.

After establishing these gold rows for the assessment set, we evaluated the initial automated rows with reference to the gold rows. To score this, we assigned binary labels to each pair of original and abridged sentences, where pairs that were part of the same row were labeled with the positive class and all other pairs were labeled with the negative class. Given these labels for the rows automatically produced with the $R\text{-}1_p$ scoring method compared against the labels for the gold rows, the F1 score of the automated rows was 0.967. A clear drawback to using unigram overlap to measure similarity is that it does not account for differences in word order. However, taking this into account by using bigrams instead of unigrams to calculate ROUGE precision (i.e. $R\text{-}2_p$) reduced the F1 to 0.935, likely because it added more sparsity to the overlap units. We also evaluated other methods besides ROUGE for computing $sim(o, a)$, in particular cosine similarity between spans encoded as vectors by pretrained language models. A.1 reports the results for these alternative methods, none of which outperform $R\text{-}1_p$. Finding that a discrete word-based metric captures similarity between the original and abridged text better than methods based on distributional semantics, we can conclude that the abridgements preserve much of the verbatim original text.

3.3 Partial validation strategy

The time spent on validating this assessment set indicated that it would require significant resources to fully review rows for all book chapters. Meanwhile, our evaluation with an F1 result of 0.967 revealed that we can expect the majority of automated rows to be correct. Thus, we considered how to focus effort on correcting the small percentage of rows that would contain erroneously aligned spans. A qualitative examination of these rows in the assessment set showed that their $R\text{-}1_p$ similarity scores were lower than those of the correct rows. There were two particular cases where lower-scoring rows tended to be incorrect. The first was rows with two or more sentences in the abridged span. The second and more common case was when a row was adjacent to another row where the original span was aligned with an empty span (i.e. a_m = 0). Often at least one abridged sentence in the low-scoring row should have actually been paired with the adjacent original span. We thus did an experiment where a human validator reviewed only the assessment rows with scores < 0.9 that qualified as one of the two above cases. Selectively applying corrections to just these rows boosted the F1 score of the entire assessment set from 0.967 up to 0.99. We therefore decided to apply this strategy of partially validating automated rows to create the train set for ABLIT.

3.4 Full dataset

To construct the rest of the ABLIT dataset, we ran the automated alignment procedure on all other chapters, and then applied the partial validation strategy described above. Because we previously confirmed high inter-rater agreement, each chapter was reviewed by a single validator. Generalizing from the assessment set, we estimate that 99% of the rows in this train set are correct. To ensure an absolute gold standard for evaluating models, we set aside five chapters in each of the books and then fully validated their rows as we did with the assessment set. We repurposed the assessment set to be a development set that we used accordingly in our experiments. Ultimately, ABLIT consists of

[4] Using github.com/Diego999/py-rouge

Original Span	Abridged Span
[The letter was not unproductive.] [It re-established peace and kindness.]	[The letter re-established peace and kindness.]
[Mr. Guppy sitting on the window-sill, nodding his head and balancing all these possibilities in his mind, continues thoughtfully to tap it, and clasp it, and measure it with his hand until he hastily draws his hand away.]	[Mr. Guppy sitting on the window-sill, taps it thoughtfully, until he hastily draws his hand away.]
[At last the gossips thought they had found the key to her conduct, and her uncle was sure of it; and what is more, the discovery showed his niece to him in quite a new light, and he changed his whole deportment to her accordingly.]	[At last the gossips thought they had found the key to her conduct, and her uncle was sure of it .] [The discovery altered his whole behaviour to his niece.]
[They trooped down into the hall and into the carriage, Lady Pomona leading the way.] [Georgiana stalked along, passing her father at the front door without condescending to look at him.]	[They trooped downstairs, Georgiana stalking along.] [She passed her father at the front door without condescending to look at him.]

Table 1: Examples of alignment rows. Sentence boundaries are denoted by brackets ([]). We highlight preserved words in blue and underline the reordered ones. Added words are in green.

808, 10, and 50 chapters in the train, development, and test sets, respectively. Table 1 shows some examples of alignment rows in the dataset.

4 Characterizing abridgements

	Train	Dev	Test (Chpt Mean)
Chpts	808	10	50
Rows	115,161	1,073	9,765 (195)
O_{pars}	37,227	313	3,125 (62)
A_{pars}	37,265	321	3,032 (61)
O_{sents}	122,219	1,143	10,431 (209)
A_{sents}	98,395	924	8,346 (167)
$\%A_{sents}$	80.5	80.8	80.0
O_{wrds}	2,727,491	29,908	231,874 (4,637)
A_{wrds}	1,718,900	17,630	143,908 (2,878)
$\%A_{wrds}$	63.0	58.9	62.1

Table 2: Number of chapters (Chpts), alignment rows ($Rows$), paragraphs ($pars$), sentences ($sents$), and words ($wrds$) across all original (O) and abridged (A) books. The per-chapter means appear for the test set.

O_{sents}	A_{sents}	Train	Dev	Test
1	1	75.8	74.7	75.7
1	0	17.4	17.3	17.3
2+	1	4.3	4.8	4.6
1	2+	2.1	3.2	1.9
2+	2+	0.3	0.0	0.5

Table 3: Distribution of row sizes by number of sentences ($sents$) in original (O) and abridged (A) spans

4.1 Overview

Table 2 lists the size of ABLIT in terms of alignment rows, paragraphs, sentences, and words (see Table A.10 for these numbers compared by book). Here we call attention to the numbers for the fully-validated test set, but the numbers for the train set closely correspond. The development set slightly varies from the train and test set for a few statistics, likely due to its small size. Judging by the test set, the abridged chapters have almost the same number of paragraphs as the original, but they have 80% of the number of sentences ($\%A_{sents}$) and ≈62% of the number of words as the original ($\%A_{wrds}$).

Table 3 pertains to the size of the original and abridged spans in each row, where size is the number of sentences in each span. The table shows the relative percentage of rows of each size. The majority of test rows (≈76%) contain a one-to-one alignment between an original and abridged sentence (i.e. O_{sents} = 1, A_{sents} = 1). Meanwhile, ≈17% contain an original sentence with an empty abridged span (O_{sents} = 1, A_{sents} = 0). A minority of rows (≈5%) have a many-to-one alignment (O_{sents} = 2+, A_{sents} = 1) and a smaller minority (≈2%) have a one-to-many alignment (O_{sents} = 1, A_{sents} = 2+). Many-to-many alignments (O_{sents} = 2+, A_{sents} = 2+) are more rare (0.5%).

4.2 Lexical relations

As demonstrated by the success of the R-1_p metric for creating alignment rows (Section 3.2), an original span and an abridged span typically align if most of the words in the abridged are contained in the original. Table 4 shows the binned distribution of the R-1_p scores for the rows. Rows with an exact score of 0.0 (≈17% of rows in the test set) consist almost exclusively of original spans aligned to empty spans, which is why this number is comparable to the second line of Table 3. Many

rows have perfect scores of exactly 1.0 (55%), signifying that their abridged span is just an extraction of some or all of the original span. The abridged spans where this is not the case (i.e. they have some words not contained in the original) still copy much of the original: 24% of test rows have a $R\text{-}1_p$ score above 0.75 and below 1.0, while only a small minority (≈4%, the sum of lines 1-3 in the table) have a score above 0.0 and below 0.75.

Score Bin	Train	Dev	Test
0.0	17.5	17.6	17.4
(0.0, 0.25]	0.1	0.2	0.1
(0.25, 0.5]	0.5	0.9	0.6
(0.5, 0.75]	2.6	4.6	2.9
(0.75, 1.0)	23.9	31.5	24.0
1.0	55.5	45.2	55.0

Table 4: Binned distribution of $R\text{-}1_p$ scores for rows

	Train	Dev	Test
O_{rmv}	40.9	45.9	41.9
O_{prsv}	59.1	54.1	58.1
A_{add}	6.3	8.3	6.4
A_{prsv}	93.7	91.7	93.6
$Rows_{rmv}$	71.1	80.3	73.2
$Rows_{prsv}$	82.5	82.7	82.6
$Rows_{add}$	37.4	48.8	39.4
$Rows_{reord}$	11.8	16.5	11.7

Table 5: *Top:* the % of original words that are removed or preserved from the abridgement. *Middle:* the % of abridged words that are added or preserved. *Bottom:* the % of alignment rows with each lexical relation.

To expand on this analysis, we enumerate the common and divergent words between the words o_{wrds} in the original span and the words a_{wrds} in the abridged span. The words that appear in o_{wrds} but not a_{wrds} are removed words, i.e. $o_{rmv} = |o_{wrds} - a_{wrds}|$. All other original words are preserved in the abridgement, i.e. $o_{prsv} = |o_{wrds} - o_{rmv}|$. Accumulating these counts across all original spans $o \in O$, the top section of Table 5 indicates the percentages of removed and preserved words among all original words. In the test set, ≈42% of original words are removed, and thus ≈58% are preserved. Next, we count the added words in the abridgement, which are those that appear in a_{wrds} and not o_{wrds}, i.e. $a_{add} = |a_{wrds} - o_{wrds}|$. All other abridged words are preserved from the original, i.e. $a_{prsv} = |a_{wrds} - a_{add}|$. Accumulating these counts across all abridged spans $a \in A$, the middle section of Table 5 indicates that only ≈6% of words in the test set abridgements are added words, and thus ≈94% of abridged words are preservations.

We also report the number of rows where these removal, preservation, and addition relations occur at least once. For instance, if $o_{rmv} > 0$ for the original span in a given row, we count that row as part of $Rows_{rmv}$. The bottom section of Table 5 shows the percentage of rows with each relation among the total number of rows in the dataset. In ≈73% of the test rows, the abridged span removes at least one word from the original. In ≈83% of rows, the abridged span preserves at least one word from the original. In ≈39% of rows, the abridged span adds at least one word that does not appear in the original. We considered the possibility that preserved words could be reordered in the abridgement. To capture this, we find the longest contiguous sequences of preserved words (i.e. "slices") in the abridged spans. A row is included in $Rows_{reord}$ if least one pair of abridged slices appears in a different order compared to the original span. This reordering occurs in ≈12% of rows.

It is clear from this analysis that the abridgements are quite loyal to the original versions, but they still remove a significant degree of text and introduce some new text. The examples in Table 1 highlight these relations. We can qualitatively interpret from the examples that some added words in the abridged span are substitutions for removed original words (e.g. "tap" > "taps" in the second example, "changed" > "altered" in the third example). See A.4 for additional discussion about how some of these relations pertain to common NLI tasks.

5 Predicting what to abridge

Garbacea et al. (2021) points out that a key (and often neglected) preliminary step in simplification is to distinguish text that could benefit from being simplified versus text that is already sufficiently simple. This is also an important consideration for abridgement, since it seeks to only modify text in places where it improves readability. Accordingly, we examined whether we could automatically predict the text in the original that should be removed when producing the abridgement. As explained in Section 4, a removed word could mean the author

substituted it with a different word(s) in the abridgement, or simply excluded any representation of its meaning. However, both cases indicate some change is applied to that word. We modeled this through a binary sequence labeling task. Given a passage with original tokens o_{toks} and corresponding abridged tokens a_{toks}, we assigned each token t in o_{toks} the label of `preserved` (l=0) if it also appeared in a_{toks}, and otherwise the label of `removed` (l=1) if it did not appear in a_{toks}. Thus the task was to predict the label sequence $[l_1, l_2, ...l_n]$ from the token sequence $[t_1, t_2, ...t_n]$.

5.1 Model inputs

We can derive a token-label sequence from each alignment row, by which each original span corresponds to a single input instance. However, the size of these spans varies across rows. To produce models that handle texts where these span boundaries are not known in advance, we consider consistent-length passages whose boundaries can be automatically inferred. Thus the ABLIT interface can provide pairs where a fixed-length passage from the original chapter (i.e. a sentence, paragraph, or multi-paragraph chunk) is aligned to its specific corresponding abridged version. We enable this by finding the respective positions of the longest common word sequences between the original and abridged spans. Each of these overlapping subsequences is represented as a slice of the original text with indices (o_i, o_j) mapped to a slice of the abridged text (a_i, a_j). Then, given a passage in the original text with indices (o_l, o_m), we find all enclosed slices (o_i, o_j) where $o_i >= o_l$ and $o_j <= o_m$. For each slice we retrieve its corresponding abridgement slice (a_i, a_j). Given the earliest text position $\min a_i$ and latest position $\max a_j$ among these abridgement slices, the full abridgement for the passage at (o_l, o_m) is the text covered by the indices ($\min a_i$, $\max a_j$). As an example, consider the first line in Table 1. If retrieving abridgements for sentence-length passages, the first sentence in the original span "The letter was not unproductive." will yield "The letter" as the abridgement. The second original sentence "It re-established peace and kindness" will yield the abridgement "re-established peace and kindness". Varying passage size enables us to assess how much context beyond a single row is beneficial in modeling abridgements. See A.5 for additional details.

5.2 Experiment

Passage	Toks	P	R	F1
Rows	26	0.692	0.442	0.532
Sentences	24	0.677	0.453	0.535
Paragraphs	81	0.686	0.460	0.546
Chunks (S=10)	303	0.670	0.501	0.569
All=`removed`	-	0.415	1.000	0.583

Table 6: F1 scores of abridgement label prediction for test set with models trained on varying passage sizes. **Toks** is the mean number of tokens in each passage type.

Model: To predict abridgement labels (`preserved`/`removed`), we used a ROBERTA-based sequence labeling model, which has been applied to several other NLI tasks (Liu et al., 2019). We divided chapters according to varying passage sizes and trained a separate model on the token-label sequences[5] associated with each passage size. The passages were either sentences (detected by NLTK), paragraphs (detected by line breaks), or multi-paragraph 'chunks'. Each chunk consisted of one or more paragraphs of S sentences, such that paragraphs were combined into the same chunk when their total number of sentences did not exceed S. As an additional reference, we trained a model where each passage was an original span directly taken from a single alignment row. As explained in Section 5.1, this passage length (termed Rows) is divined from the dataset in oracle fashion. We did not train a model on the full chapters as inputs because the average length of these inputs (5,044 tokens) greatly exceeds the ROBERTA limit of 512. See A.6 for more details about the model.

Results: Each model was evaluated on instances of the corresponding passage size in the test set. Table 6 displays the model results in terms of the precision (P), recall (R), and F1 score of predicting that a token should be removed. We compare these results with the baseline of labeling all tokens in the chapter as `removed` (final line). For chunks, we performed a grid search over values of S and report the best result obtained at S=10. The F1 results show that the longest passage size (Chunks) yields the best predictions, suggesting the importance of chapter context beyond that given by the span in a

[5] A "token" in this case is a sub-token unit defined by the ROBERTA tokenizer, rather than a whitespace-separated "word" pertaining to Section 4.

single row. The consistent discrepancy in precision and recall across all models indicates that they tend to over-predict the number of tokens that should be preserved, thus missing many that should be removed. This results in an overall F1 that is lower than what occurs when all tokens are removed.

6 Producing abridgements

The above results show that anticipating what parts of a text should be changed when writing its abridged version is not trivial. The full task of producing an abridgement implicitly involves inferring these `preserved`/`removed` labels while additionally predicting the specific text that dictates these labels. We examine models that have been applied to tasks related to abridgement to establish benchmarks for this new task, with the intent that these benchmarks will inspire future work.

6.1 Models

We consider the following models to generate an abridged version of an original chapter:

Naive Extractive Baselines: As a reference point for our evaluation metrics, we report the performance of very weak baselines. In particular, we copy the entire original text as the abridgement (COPY). Alternatively, we select T percent of original tokens (RANDEXTTOKS) as the abridgement.

Extractive Methods based on Labels: Using the best prediction model from Section 5 to apply labels to tokens, we extract all original tokens labeled as `preserved` to form the abridgement (EXTTOKS). To reveal the maximum performance we can expect from EXTTOKS, we also run this approach using the gold labels instead of predicted labels (PERFECTEXTTOKS). We note that extracting tokens is not a conventional approach to summarization, since it can result in violations of grammatical fluency within sentences. Therefore we also consider the more conventional approach of extracting sentences. In particular, we form an abridgement by selecting a subset of sentences in the original chapter where at least P percent of tokens are labeled as `preserved` (EXTSENTS).

Generation Models: Extractive methods cannot introduce words into the abridgement that are not in the original, so for this we need to consider generation models. In particular, we examine two transformer-based sequence-to-sequence models that have been used for various generation tasks including summarization: T5-BASE (Raffel et al., 2020) (termed TUNEDT5 here) and BART-BASE (Lewis et al., 2020) (termed TUNEDBART). We fine-tuned both models on the ABLIT train set, specifically on inputs consisting of chunks with 10 sentences, since this passage size yielded the best label prediction. To assess the impact of these models' observation of ABLIT, we compare them with abridgements produced by prompting the non-finetuned T5-BASE to perform zero-shot summarization (ZEROSHOTT5). See A.7 for more details about these models. For all models, we generated an abridgement for an original chapter by dividing the chapter into chunks, generating output for each chunk (with 5-beam decoding), then concatenating the outputs to form the complete abridgement.

6.2 Evaluation metrics

We evaluate the predicted abridgements through comparison with the human-authored reference abridgements. First, we measure the overall word-based similarity between the predicted abridgement a_{pred} and reference abridgement a_{ref} using ROUGE, the most common metric for evaluating summarization. We specifically report the ROUGE F1 score of longest common subsequences (R-L), which rewards longer overlapping text between the predicted and reference. Additionally, we assess how accurately a_{pred} removed words from the original. These removal scores, Rmv_p and Rmv_r, are comparable to precision and recall in Section 5, except that they are not order-sensitive. We also compute an addition score Add that assesses how many words not in the original were correctly added to a_{pred}. Because of the sparsity of added words relative to removed words, we combine the precision and recall of this measure and report the F1. See A.8 for the formal definition of these metrics.

6.3 Results

Table 7 reports the mean token length (Toks) and metric scores of the abridgements associated with each model for the test set chapters. Where applicable, we selected the T and P parameters based on the results of a grid search on the development set. The results again convey that abridgement is largely a text extraction task, though a challenging one. The relatively low R-L score of ZEROSHOTT5 confirms that ABLIT is quite different from the existing summarization datasets that T5-BASE is trained on. The high R-L of PERFECTEXTTOKS validates that precisely identifying which words to

Name	Description	Toks	R-L	Rmv_p	Rmv_r	Add
HUMAN	Reference (a_{ref})	2,878	-	-	-	-
COPY	Duplicate original	4,637	0.739	0.000	0.000	0.000
RANDEXTTOKS (T=0.6)	$T\%$ randomly selected original tokens	2,787	0.753	0.692	0.706	0.000
EXTTOKS	Original tokens predicted as `preserved`	3,160	0.818	0.826	0.688	0.006
PERFECTEXTTOKS (upper bound for EXTTOKS)	Original tokens where gold label is `preserved`	2,664	0.950	0.915	0.997	0.034
EXTSENTS (P=0.65)	Original sentences with at least $P\%$ tokens predicted as `preserved`	2,856	0.792	0.745	0.716	0.001
TUNEDT5	Generate from finetuned T5	3,833	0.727	0.843	0.387	0.275
TUNEDBART	Generate from finetuned BART	3,673	0.780	0.837	0.460	0.365
ZEROSHOTT5	Generate from non-finetuned T5	1,157	0.416	0.477	0.931	0.019

Table 7: Scores of predicted abridgements on evaluation metrics. For all metrics, higher scores are more optimal.

Original	Reference	TUNEDBART
The windows were half open because of the heat, and the Venetian blinds covered the glass,–so that a gray grim light, reflected from the pavement below, threw all the shadows wrong, and combined with the green-tinged upper light to make even Margaret's own face, as she caught it in the mirrors, look ghastly and wan.	The windows were half open because of the heat, and Venetian blinds covered the glass, giving the light a green tinge that made her face in the mirrors look ghastly and wan.	The windows were half open because of the heat, and the Venetian blinds covered the glass - so that a grey grim light, reflected from the pavement below, threw all the shadows wrong, and made even Margaret's own face look ghastly and wan.
We must suppose little George Osborne has ridden from Knightsbridge towards Fulham, and will stop and make inquiries at that village regarding some friends whom we have left there. How is Mrs. Amelia after the storm of Waterloo? Is she living and thriving? What has come of Major Dobbin, whose cab was always hankering about her premises?	We must now make inquiries at Fulham about some friends whom we have left there. How is Mrs. Amelia? Is she living and thriving? What has become of Major Dobbin?	We must suppose little George Osborne has ridden towards Fulham, and will stop and make inquiries about some friends whom we have left there. How is Mrs. Amelia after the storm of Waterloo? Is she living and thriving? What has come of Major Dobbin, whose cab was always hankering about her premises?

Table 8: Abridgements predicted by TUNEDBART for excerpts of North and South and Vanity Fair

remove goes far in producing the abridgement. The high Rmv_p scores for all approaches that observe ABLIT show they can all preserve the original text reasonably well. Knowing which words to remove is harder, particularly for the generation models, as indicated by their low Rmv_r. The extractive methods have no inherent opportunity to obtain an Add score that is non-trivially above 0[6]. The generation models do show a small benefit here in correctly adding some new words to the abridgement.

The examples in Table 8 qualitatively represent the evaluation outcome for the TUNEDBART model. The predicted abridgements for these excerpts remove some of the same original text as the reference and also add a few words consistent with the reference, but they still retain more of the original text than the reference. Additional examples of predicted abridgements are shown in A.9.

[6]It is possible for Add to be slightly above 0 with the extractive approaches due to tokenization; see A.8.

7 Conclusion

In this paper, we introduced ABLIT, a corpus of original and abridged versions of English literature. ABLIT enables systematic analysis of the abridgement task, which has not yet been studied from an NLP perspective. Our work demonstrates that while abridgement is related to existing tasks like summarization and simplification, it is marked by the challenge of maintaining loyalty to the original text. Our experiments motivate an opportunity for models that better meet this objective. We also envision future resources that generalize the abridgement task to other texts beyond English literature.

8 Ethical Considerations

As stated in the introduction, all data and code used in this work will be freely available upon publication of the paper. All text included in the dataset is in the public domain. Additionally, we explicitly confirmed approval from the author of the abridged books to use them in our work. For the data validation task, the validators were employed within our institution and thus were compensated as part of their normal job role. Given that the dataset is derived directly from published books, it is possible that readers may be offended by some content in these books. The validators did not report any subjective experience of this. With regard to our modeling approaches, large pretrained models like the ones we use here for generating abridgements have a well-known risk of producing harmful content (e.g. Gehman et al., 2020). For the generation models fine-tuned on ABLIT, we did not subjectively observe any such text in the sample output we assessed. We judge that our controlled selection of training data reduces this risk, but does not eliminate it. Accordingly, future applications of abridgement can similarly consider careful data curation for mitigating this risk.

References

Atef Chaudhury, Makarand Tapaswi, Seung Wook Kim, and Sanja Fidler. 2019. The shmoop corpus: A dataset of stories with loosely aligned summaries. *arXiv preprint arXiv:1912.13082*.

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual language model pretraining. *Advances in neural information processing systems*, 32.

William Coster and David Kauchak. 2011. Simple English Wikipedia: A new text simplification task. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 665–669, Portland, Oregon, USA. Association for Computational Linguistics.

Ido Dagan and Oren Glickman. 2004. Probabilistic textual entailment: Generic applied modeling of language variability. *Learning Methods for Text Understanding and Mining*, 2004:26–29.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Cristina Garbacea, Mengtian Guo, Samuel Carton, and Qiaozhu Mei. 2021. Explainable prediction of text complexity: The missing preliminaries for text simplification. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1086–1097, Online. Association for Computational Linguistics.

Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A. Smith. 2020. RealToxicityPrompts: Evaluating neural toxic degeneration in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3356–3369, Online. Association for Computational Linguistics.

Anna Kazantseva. 2006. An approach to summarizing short stories. In *Student Research Workshop*, pages 55–62.

Wojciech Kryściński, Nazneen Rajani, Divyansh Agarwal, Caiming Xiong, and Dragomir Radev. 2021. Booksum: A collection of datasets for long-form narrative summarization.

Faisal Ladhak, Bryan Li, Yaser Al-Onaizan, and Kathleen McKeown. 2020. Exploring content selection in summarization of novel chapters. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5043–5054, Online. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Rada Mihalcea and Hakan Ceylan. 2007. Explorations in automatic book summarization. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 380–389, Prague, Czech Republic. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21:1–67.

Renliang Sun, Hanqi Jin, and Xiaojun Wan. 2021. Document-level text simplification: Dataset, criteria and baseline. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7997–8013, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Wei Xu, Chris Callison-Burch, and Courtney Napoles. 2015. Problems in current text simplification research: New data can help. *Transactions of the Association for Computational Linguistics*, 3:283–297.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32.

Weiwei Zhang, Jackie Chi Kit Cheung, and Joel Oren. 2019. Generating character descriptions for automatic summarization of fiction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7476–7483.

A Appendix

A.1 Details about Automated Alignments

Table 9 shows the results of all methods we assessed for computing similarity between original and abridged spans to create alignment rows. In addition to the $R\text{-}1_p$ word-based metric, we assessed vector-based similarity encoded by different configurations of pretrained language models: BERT (Devlin et al., 2019), XLNET (Yang et al., 2019), XLM (Conneau and Lample, 2019), and ROBERTA (Liu et al., 2019). We used the HuggingFace Transformers implementation of these models: https://huggingface.co/docs/transformers/index. For each model we report the best result from a grid search of over size penalty (pn) values $[0, 0.25]$. As displayed, the vectors that obtained the best F1 came from BERT (Devlin et al., 2019), particularly BERT-BASE-UNCASED, which consists of 110M parameters. See additional details about this model here: https://huggingface.co/bert-base-uncased. Ultimately, however, the result from BERT-BASE-UNCASED was still outperformed by $R\text{-}1_p$. As reported in Section 3.3, the resulting rows were further improved by applying the described partial validation strategy (final line of table).

A.2 Details about Validation Task

We utilized Google Sheets as an interface for the human validation tasks, which enabled validators to easily review and correct the alignment rows. We produced a single spreadsheet per chapter, where each spreadsheet row corresponded to an alignment row. For the partial validation strategy, we designed a Google Apps Script (https://developers.google.com/apps-script) that visually highlighted spreadsheet rows qualifying for partial validation so that validators could specifically attend to those rows.

For the development (assessment) and test sets, there were a few cases where the validators edited the spans themselves in order to correct sentence segmentation errors (e.g. wrongly segmenting after honorifics like "Mr.").

A.3 Size of ABLIT Compared by Book

Table 10 shows characteristics of the data for each book in terms of number of alignment rows, original words, and abridged words.

A.4 NLI Challenges in ABLIT

Table 11 shows some examples of rows in ABLIT where modeling the relation between the original and abridged span involves NLI challenges like abstractive paraphrasing, figurative language interpretation, commonsense reasoning, and narrative understanding.

A.5 Comment about Model Inputs

Because the method for converting rows into passages of a consistent length (i.e. sentences, paragraphs, chunks) relies on string matching, the boundaries of the abridged passage may be off by one or a few words, which becomes more minimal as the size of the passages increase. These tend to occur when a word at the end of the original passage is replaced by a synonym in the abridged passage. However, a manual review of our assessment set revealed that only 0.4% of sentences in the original text yielded abridgements with imprecise boundaries, and no paragraphs (and consequently no chunks) had this issue.

A.6 Details about Binary Prediction Model

For all passage sizes, we initialized models with the ROBERTA-BASE weights using the HuggingFace Transformers implementation: https://huggingface.co/docs/transformers/v4.16.2/en/model_doc/roberta#transformers.RobertaModel. ROBERTA-BASE consists of 125M parameters. See additional details here: https://huggingface.co/roberta-base. The maximum sequence length allowed by this model is 512, so we truncated all input tokens beyond this limit. We fine-tuned each model for 5 epochs, saving model weights after each epoch of training, and selected the model with the highest F1 score on the development set to apply to our test set. We used the AdamW optimizer (Loshchilov and Hutter, 2017) and a batch size of 16. It took ≈2 hours to train each model on a g4dn.2xlarge AWS instance. During evaluation, any input tokens beyond the model length limit were assigned the default label of `preserved`. The result for each model reported in Table 6 is based on a single run of the training procedure.

A.7 Details about Generation Models

Both TUNEDT5 and TUNEDBART were fine-tuned using the HuggingFace trans-

Similarity Metric	pn	**P**	**R**	**F1**
Vector cosine similarity				
BERT-BASE-UNCASED	0.21	0.963	0.952	0.957
BERT-BASE-CASED	0.22	0.948	0.934	0.940
BERT-LARGE-UNCASED	0.21	0.934	0.919	0.926
BERT-LARGE-CASED	0.21	0.944	0.935	0.939
XLNET-BASE-CASED	0.22	0.753	0.731	0.742
XLNET-LARGE-CASED	0.21	0.583	0.564	0.573
XLM-MLM-EN	0.21	0.821	0.816	0.818
ROBERTA-BASE	0.21	0.738	0.717	0.727
ROBERTA-LARGE	0.21	0.592	0.573	0.582
Word overlap similarity				
R-1_p	0.175	0.964	0.969	0.967
R-2_p	0.175	0.912	0.958	0.935
R-1_p + partial human validation	0.175	0.990	0.991	0.990

Table 9: Extended results for accuracy of automated alignment methods

	Train			**Dev**			**Test**		
Book (Orig Author)	Rows (Chpts)	O_{wrds}	$\%A_{wrds}$	Rows	O_{wrds}	$\%A_{wrds}$	Rows	O_{wrds}	$\%A_{wrds}$
Bleak House (Charles Dickens)	17,948 (62)	390,846	63.2	24	935	20.0	1,746	38,132	62.9
Can You Forgive Her? (Anthony Trollope)	16,494 (74)	350,092	62.2	94	3,216	49.5	1,339	27,660	61.2
Daniel Deronda (George Eliot)	12,735 (64)	333,283	61.6	158	3,524	61.9	786	25,332	49.1
Mansfield Park (Jane Austen)	5,744 (42)	159,863	67.0	91	3,564	62.1	795	22,607	66.1
North and South (Elizabeth Gaskell)	8,922 (46)	193,335	67.9	184	4,907	68.5	1,169	23,159	70.0
Shirley (Charlotte Bronte)	12,027 (31)	235,888	63.2	253	5,987	57.4	1,031	23,369	60.4
The Way We Live Now (Anthony Trollope)	19,355 (94)	392,552	60.3	166	4,345	53.7	1,122	23,238	60.7
Tristram Shandy (Laurence Sterne)	4,805 (305)	216,982	66.7	5	439	77.0	69	3,972	72.3
Vanity Fair (W. M. Thackeray)	11,682 (62)	334,770	59.8	18	717	60.9	738	23,609	57.4
Wuthering Heights (Emily Bronte)	5,449 (28)	119,880	66.3	80	2,274	68.3	970	20,796	71.0
All	115,161 (808)	2,727,491	63.0	1,073	29,908	58.9	9,765	231,874	62.1

Table 10: Statistics for each book in the AbLit dataset, in terms of number of alignment rows, total original word (O_{wrds}), and proportional length of abridgement relative to original ($\%A_{wrds}$). The number of chapters in the train set for each book is shown; there is 1 chapter per book in the development set and 5 chapters per book in the test set.

Original Span	Abridged Span	Type of Challenge
Still there was not a word.	No one spoke.	Paraphrasing: abridgement has same meaning as original but no word overlap
But it is time to go home; my appetite tells me the hour.	But it is time to go home; I am hungry.	Interpretation of figurative language: abridgement replaces phrase "appetite tells me the hour" with more literal term "hungry"
"Daniel, do you see that you are sitting on the bent pages of your book?"	"Daniel, you are sitting on the bent pages of your book."	Change in dialogue act: question in original is transformed into statement in abridgment
While she was at Matching, and before Mr. Palliser had returned from Monkshade, a letter reached her, by what means she had never learned. "A letter has been placed within my writing-case," she said to her maid, quite openly. "Who put it there?"	While she was at Matching, a letter reached her, by what means she never learned, although she suspected her maid of placing it inside her writing-case.	Dialogue interpretation: abridgement summarizes the narrative event (suspecting maid of placing letter) conveyed by the spoken utterances in the original text ("A letter has been placed . . . she said to her maid.)
"If you will allow me, I have the key," said Grey. Then they both entered the house, and Vavasor followed his host up-stairs.	Mr. Grey unlocked the door of his house, and Vavasor followed him upstairs.	Commonsense inference: abridgement involves knowledge that doors are unlocked by keys, which is not explicit in the original text
George Osborne was somehow there already (sadly "putting out" Amelia, who was writing to her twelve dearest friends at Chiswick Mall), and Rebecca was employed upon her yesterday's work.	George Osborne was there already, and Rebecca was knitting her purse.	Narrative inference: "knitting her purse" in the abridgement is the event referenced by "yesterday's work" in the original, and resolving this requires knowledge of the previous text in the chapter
But Kate preferred the other subject, and so, I think, did Mrs. Greenow herself.	But Kate preferred the subject of the Captain, and so, I think, did Mrs. Greenow herself.	Elaboration: abridgement specifies "Captain" is the "other subject" implied in the original

Table 11: Examples of alignment rows that represent a difficult language understanding problem

formers library, in particular this script: http://github.com/huggingface/transformers/blob/master/examples/pytorch/summarization/run_summarization.py. TUNEDT5 was initialized from T5-BASE (Raffel et al., 2020), which consists of ≈220M parameters. See additional details here: https://huggingface.co/t5-base. For this model, we prepended the prefix "summarize: " to the target (i.e. the abridged passage), consistent with how T5-BASE was trained to perform summarization. TUNEDBART was initialized from BART-BASE (Lewis et al., 2020), which consists of 140M parameters. See additional details here: https://huggingface.co/facebook/bart-base. For both TUNEDT5 and TUNEDBART, we used a maximum length of 1024 for both the source (original passage) and target (abridged passage), and truncated all tokens beyond this limit. We evaluated each model on the development set after each epoch and concluded training when cross-entropy loss decreased, thus saving the model weights with the optimal loss. We used a batch size of 4. For all other hyperparameters we used the default values set by this script, which specifies AdamW for optimization. It took ≈3 hours to train each model on a g4dn.4xlarge AWS instance. The result for each model reported in Table 7 is based on a single run of the training procedure.

A.8 Details about Evaluation Metrics

The formal definition of the removal metrics are as follows. If $o_{rmv}(a_{pred})$ are the words in the original that are not in the predicted abridgement, and $o_{rmv}(a_{ref})$ are the words in the original that are not in the reference abridgement, then we consider the number of correctly removed words: $Correct\_Rmv = |o_{rmv}(a_{pred}) \cap o_{rmv}(a_{ref})|$. The precision of this measure $Rmv_p = \frac{Correct\_Rmv}{o_{rmv}(a_{pred})}$ is the proportion of correctly removed words among all words that the predicted abridgement removed. The recall $Rmv_r = \frac{Correct\_Rmv}{o_{only}(a_{ref})}$ is the proportion of correctly removed words among all words that the reference abridgement removed.

The formal definition of the addition metric is as follows. If $a_{add}(a_{pred})$ are the words in the predicted abridgement that do not appear in the original, and $a_{add}(a_{ref})$ are those in the reference abridgement that do not appear in the original, then we consider the number of correctly added words: $Correct\_Add = |a_{add}(a_{pred}) \cap a_{add}(a_{ref})|$. The precision of this measure $Add_p = \frac{Correct\_Add}{a_{add}(a_{pred})}$ is the proportion of correctly added words among all added words in the predicted abridgement. The recall $Add_r = \frac{Correct\_Add}{a_{add}(a_{ref})}$ is the proportion of correctly added words among all added words in the reference abridgement. The final score Add is the F1 of Add_p and Add_r.

Regarding the above-zero scores of the extractive methods on the Add metric, there are two reasons for this. One reason is that the prediction model uses sub-tokens while the Add metric analyzes whitespace-separated words. Consequently, one sub-token may be predicted as `preserved` while others within the same word are predicted as `removed`. Isolated from these other sub-tokens, the `preserved` sub-token will be recognized as a new added word in the abridgement. The other reason is that a single word in the original may be split into two words in the abridgement, or vice-versa. For example, we observed that "Mr." gets split into two tokens ("Mr", '.') in some contexts and is treated as one token ("Mr.") in others. If the original text represents this item as two tokens and both the extracted and reference abridgement represent it as a single token, then this single token will be counted as an added word in the extracted abridgement.

A.9 Examples of Produced Abridgements

Tables 12 and 13 below show excerpts of the abridgements produced by the EXTSENT and TUNEDBART models, alongside the original chapter text and human-authored reference abridgement. The sentences in each excerpt are lined up to better visualize their differences.

Original	Reference	EXTSENTS	TUNEDBART
Seven days glided away, every one marking its course by the henceforth rapid alteration of Edgar Linton's state.	In the next seven days Edgar Linton's state grew rapidly worse.		Seven days glided away, every one marking its course by the rapid alteration of Edgar Linton's state.
The havoc that months had previously wrought was now emulated by the inroads of hours.			The havoc that months had previously wrought was now emulated by the inroads of hours.
Catherine we would fain have deluded yet;	Catherine could no longer be deluded:		Catherine we would fain have deluded yet;
but her own quick spirit refused to delude her:		but her own quick spirit refused to delude her:	but her own quick spirit refused to delude her:
it divined in secret, and brooded on the dreadful probability, gradually ripening into certainty.	she brooded on the dreadful probability of her father's death, gradually ripening into certainty.		it brooded on the dreadful probability, gradually ripening into certainty.
She had not the heart to mention her ride, when Thursday came round;	She had not the heart to mention her ride when Thursday came round.	She had not the heart to mention her ride, when Thursday came round;	She had not the heart to mention her ride, when Thursday came round;
I mentioned it for her, and obtained permission to order her out of doors:	I obtained permission to send her out of doors:	I mentioned it for her, and obtained permission to order her out of doors:	I ordered her out of doors:
for the library, where her father stopped a short time daily-the brief period he could bear to sit up-and his chamber, had become her whole world.	for her father's chamber had become her whole world.		for the library, where her father stopped daily - the brief period he could bear to sit up - and his chamber, had become her whole world.
She grudged each moment that did not find her bending over his pillow, or seated by his side.	She grudged each moment that she did not spend bending over his pillow, or seated by his side.		She grudged each moment that did not find her bending over his pillow, or seated by his side.
Her countenance grew wan with watching and sorrow, and my master gladly dismissed her to what he flattered himself would be a happy change of scene and society;	She grew pale with watching, and my master gladly dismissed her to what he thought would be a happy change of scene;	Her countenance grew wan with watching and sorrow, and my master gladly dismissed her to what he flattered himself would be a happy change of scene and society;	Her countenance grew wan with watching and sorrow, and my master gladly dismissed her to what he flattered himself would be a happy change of scene and society;
drawing comfort from the hope that she would not now be left entirely alone after his death.	drawing comfort from the hope that she would not now be left entirely alone after his death.	drawing comfort from the hope that she would not now be left entirely alone after his death.	drawing comfort from the hope that she would not now be left entirely alone after his death.

Table 12: Abridgements for an excerpt of Wuthering Heights, Chapter 27

Original	Reference	EXTSENTS	TUNEDBART
It happened that when I came home from Deal I found a note from Caddy Jellyby (as we always continued to call her), informing me that her health, which had been for some time very delicate, was worse and that she would be more glad than she could tell me if I would go to see her.	When I came home from Deal I found a note from Caddy, informing me that her health, which had been for some time very delicate, was worse and that she would be very glad if I would go to see her.	It happened that when I came home from Deal I found a note from Caddy Jellyby (as we always continued to call her), informing me that her health, which had been for some time very delicate, was worse and that she would be more glad than she could tell me if I would go to see her.	It happened that when I came home from Deal I found a note from Caddy Jellyby informing me that her health, which had been for some time very delicate, was worse and that she would be more glad than she could tell me if I would go to see her.
It was a note of a few lines, written from the couch on which she lay and enclosed to me in another from her husband, in which he seconded her entreaty with much solicitude.	It was a short note, written from her bed.		It was a note of a few lines, written from the couch on which she lay and enclosed to me in another from her husband, in which he seconded her entreaty with much solicitude.
Caddy was now the mother, and I the godmother, of such a poor little baby–such a tiny old-faced mite, with a countenance that seemed to be scarcely anything but cap-border, and a little lean, long-fingered hand, always clenched under its chin.	Caddy was now the mother, and I the godmother, of such a poor little baby - such a tiny old-faced mite, with a little lean, long-fingered hand always clenched under its chin.	Caddy was now the mother, and I the godmother, of such a poor little baby–such a tiny old-faced mite, with a countenance that seemed to be scarcely anything but cap-border, and a little lean, long-fingered hand, always clenched under its chin.	Caddy was now the mother, and I the godmother, of such a poor little baby - such a tiny old-faced mite, with a countenance that seemed to be scarcely anything but cap-border, and a little lean, long-fingered hand, always clenched under its chin.
It would lie in this attitude all day, with its bright specks of eyes open, wondering (as I used to imagine) how it came to be so small and weak.	It would lie in this attitude all day, with its bright specks of eyes open, wondering (I used to imagine) how it came to be so small and weak.	It would lie in this attitude all day, with its bright specks of eyes open, wondering (as I used to imagine) how it came to be so small and weak.	It would lie in this attitude all day, with its bright specks of eyes open, wondering how it came to be so small and weak.
Whenever it was moved it cried, but at all other times it was so patient that the sole desire of its life appeared to be to lie quiet and think.	Whenever it was moved it cried, but at all other times it lay quiet.	Whenever it was moved it cried, but at all other times it was so patient that the sole desire of its life appeared to be to lie quiet and think.	Whenever it was moved it cried, but at all other times it was so patient that the sole desire of its life appeared to be to lie quiet and think.

Table 13: Abridgements for an excerpt of Bleak House, Chapter 50