SWIPE: A Dataset for Document-Level Simplification of Wikipedia Pages

Philippe Laban **Jesse Vig** **Wojciech Kryscinski**
Shafiq Joty **Caiming Xiong** **Chien-Sheng Jason Wu**
Salesforce AI
{plaban, jvig, wojciech.kryscinski, sjoty, cxiong, wu.jason}@salesforce.com

Abstract

Text simplification research has mostly focused on sentence-level simplification, even though many desirable edits—such as adding relevant background information or reordering content—may require document-level context. Prior work has also predominantly framed simplification as a single-step, input-to-output task, only implicitly modeling the fine-grained, span-level edits that elucidate the simplification process. To address both gaps, we introduce the SWIPE dataset, which reconstructs the *document-level editing* process from English Wikipedia (EW) articles to paired Simple Wikipedia (SEW) articles. In contrast to prior work, SWIPE leverages the entire revision history when pairing pages in order to better identify simplification edits. We work with Wikipedia editors to annotate 5,000 EW-SEW document pairs, labeling more than 40,000 edits with proposed 19 categories. To scale our efforts, we propose several models to automatically label edits, achieving an F-1 score of up to 70.6, indicating that this is a tractable but challenging NLU task. Finally, we categorize the edits produced by several simplification models and find that SWIPE-trained models generate more complex edits while reducing unwanted edits.

1 Introduction

Text simplification (TS) aims to make complex documents accessible to larger audiences by lowering the barrier of reading for children, non-native speakers, and novice readers in technical domains. TS has primarily been approached in a sentence-level sequence-to-sequence (seq2seq) manner, following the methodology of mature NLG tasks such as machine translation. Prior work framed at the sentence level has focused on simplification edits that occur within sentence units, such as lexical replacements (Glavaš and Štajner, 2015) and sentence splitting (Narayan and Gardent, 2015; Sulem et al., 2018). Yet, many simplification operations, such as background elaborations (Srikanth and Li, 2020) or content reordering (Zhong et al., 2020) require document-level context.

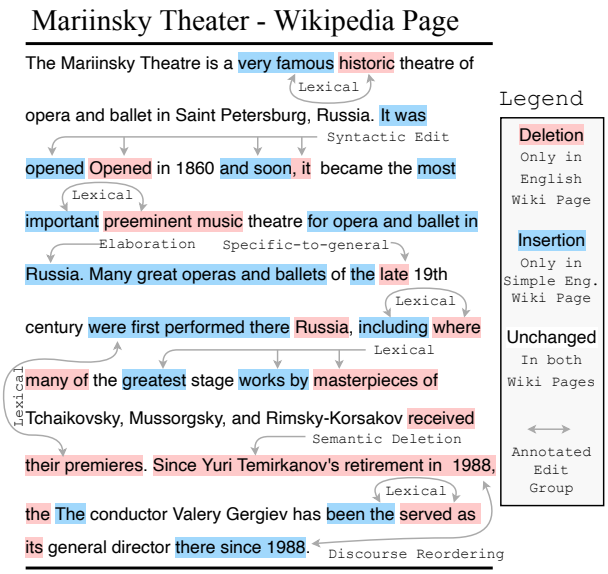


Figure 1: Sample from SWIPE, a Wikipedia-based dataset for document-level simplification. Many edits in SWIPE require document-level context.

A major roadblock to advances in document-level simplification has been the lack of large-scale and high-quality datasets. The two most popular sources of data for the English language are either the news-based Newsela which is not available publicly or the combination of English Wikipedia (EW) and Simple English Wikipedia (SEW)[1], which is large-scale but requires non-trivial processing to align Wikipedia articles with their simplified versions (Jiang et al., 2020). The alignment task has predominantly been framed as finding pairs of semantically similar sentences within the latest revisions of EW and SEW pages.

Our first contribution is to adapt the Wikipedia content alignment task to document-level granularity. We explore the entire *revision history* of Wikipedia pages and match individual revisions of

[1] https://simple.wikipedia.org

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 10674–10695
July 9-14, 2023

SEW pages with best-aligned EW revisions, rather than rely on the most recent revisions which might yield factually misaligned pairs due to outdated information. By applying our alignment method to the entire revision history of SEW – and processing two orders of magnitude more content – we create the SWIPE dataset, a high-quality and large-scale document-level simplification dataset. SWIPE consists of 145,161 document pairs, which we processed into an alignment sequence composed of three operations: *unchanged text*, *insertion*, and *deletion*. Figure 1 provides an illustrative alignment sequence of a SWIPE sample.

Our second contribution is a comprehensive analysis of edits that occur in SWIPE. We propose a 19-category edit taxonomy based on prior work and expanded for document-level edits. The categories are organized into four coarse-grained classes representing simplification objectives: Lexical, Syntactic, Semantic, and Discourse-level edits. We collaborate with active SEW editors to annotate 5,000+ alignment sequences of SWIPE. The collected annotations of around 40,000 edits reveal that all four edit classes are prevalent in SWIPE (each occurs in at least 40% of annotated documents). Document-level context is required for at least 43% of edits, and diverse edits often co-occur within documents, as SEW editors combine editing strategies when producing SEW pages.

Our third contribution is to propose models that can automatically identify edit categories and models that generate document-level simplified text. For the task of edit identification, our best model achieves a categorization F-1 score of 70.6, leaving room for future improvement. When analyzing simplification models based on the edits they produce, we find that SWIPE-trained models can produce more complex edits than prior work while generating fewer undesirable edits that potentially introduce factually incorrect content. We release the SWIPE data, the models, and experimental code publicly[2].

2 Related Work

2.1 Simplification Datasets

Simple Wikipedia was leveraged by prior work to create some of the first large-scale simplification resources, such as PWKP (Zhu et al., 2010) and SEW (Coster and Kauchak, 2011), which popularized the field framed on sentence-level simplification. Subsequent work found shortcomings in initial datasets due to low-quality alignment (Xu et al., 2015), and three main avenues for improvement were proposed. First, some work proposed to favor higher quality data sources such as Newsela (Xu et al., 2015; Srikanth and Li, 2021). However, Newsela is only available under a restrictive license, which has limited its accessibility within the research community. Second, manual annotation of smaller-scale but higher-quality evaluation sets can complement existing resources, such as HSplit (Sulem et al., 2018), TurkCorpus (Xu et al., 2016), and ASSET (Alva-Manchego et al., 2020). Finally, more advanced alignment methods were proposed to improve the automatic creation of Wikipedia-based datasets, creating Wiki-Auto (Jiang et al., 2020) and CATS (Štajner et al., 2018).

Recent work has explored simplification beyond sentence-level granularity, with some methods focused on the paragraph level (Devaraj et al., 2021; Laban et al., 2021). The D-Wikipedia dataset (Sun et al., 2021) is the closest in format to SWIPE, but analysis in Section 3.4 reveals that it is of limited quality due to a lack of filtering. With SWIPE, we extend prior work by implementing an advanced automatic alignment method to create a large-scale dataset for document-level simplification.

2.2 Categorizing Simplification Edits

Given a simplification dataset, automatic alignment methods enable the extraction of atomic edits that simplify the complex text. Prior work has analyzed such edits to gain insights and compare datasets. The most common analysis revolves around measuring the frequency of different editing operations (i.e. insertions, deletions, replacements) (Coster and Kauchak, 2011; Vásquez-Rodríguez et al., 2021). Some work has proposed annotating the operations with linguistically motivated categories that give a reason for the edit. Since most simplification resources are at the sentence granularity, edit categorizations have focused on lexical and syntactic phenomena that frequently occur within individual sentences (Aluísio et al., 2008; Scarton and Specia, 2018; Cardon et al., 2022).

Some work has leveraged Newsela to study edits that require document-level context, such as elaborations (Srikanth and Li, 2020) and content selection (Zhong et al., 2020). Other works such as arxivEdits(Jiang et al., 2022), EditEval(Dwivedi-Yu et al., 2022) or PEER(Schick et al., 2022) have

[2] https://github.com/Salesforce/simplification

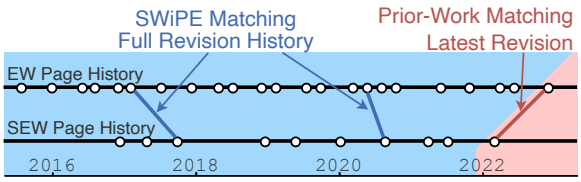


Figure 2: SWIPE matching considers the entire revision history of pages, enabling better page alignment.

studied the general problem of document editing, and have either not considered simplification edits or grouped all simplification edits within a single category. Edit categories used in SWIPE are based on existing categorization and expanded with edits less frequently, studied such as discourse and semantic edits that require document-level context.

2.3 Wikipedia Revision History

Wikipedia revision history has been used in NLP resources, from automatic grammatical error correction (Boyd, 2018; Max and Wisniewski, 2010), to vandalism detection (Chin et al., 2010; Heindorf et al., 2015), paraphrase generation (Nelken and Yamangil, 2008; Dutrey et al., 2010) or fact verification (Schuster et al., 2021). With SWIPE, we show that Wikipedia's revision history in conjunction with advanced alignment methods can be a powerful tool to create simplification datasets.

3 Creating SWIPE

3.1 Page Matching

To create a simplification dataset based on Wikipedia, pages from EW must be matched with their counterpart simplified pages in SEW. We follow prior work and leverage Wikidata (Jiang et al., 2020), Wikimedia's knowledge base, to extract Wikidata entries with both EW and SEW Wikipedia pages, and obtain a total of 226,861 page pairs, which form the basis for our dataset.

3.2 Revision Matching

By design, each Wikipedia page is a living document that is continuously updated and revised. When an editor creates a SEW page, it is common practice to select a particular revision of the corresponding EW page as a starting point and introduce a series of simplifying edits.

Most existing Wikipedia-based simplification datasets rely on matching the latest revisions of page pairs at the time of dataset creation, overlooking page revision history. Considering that EW pages are typically updated more frequently than SEW pages, such approaches might lead to misalignment in the created datasets, thus lowering the data quality. In this work, we leverage the full revision history of both the EW and SEW pages with the goal of obtaining higher-quality examples of document-level simplification. We propose the task of automatic revision matching, illustrated in Figure 2.

For the 226,861 page pairs, we obtain the entire revision history of the EW and SEW pages and extract up to 200 full-text revisions using Wikipedia's API. We obtain 22 million revisions: on average 94 revisions per EW page, and 4 per SEW page. The matching process consists of finding the EW revision that aligns best with each SEW revision. If a SEW page has multiple revisions, we include several revisions in the dataset, as long as the SEW revisions differ significantly and match distinct EW revisions (i.e., Levenshtein similarity $\leq$0.3).

We manually annotated 2,000 revision pairs with an alignment label (0/1) and conducted an exploratory study of several baseline models, with full details in Appendix A. Based on the findings, we select the NLI-based SummaC model (Laban et al., 2022a), which was originally proposed for inconsistency detection in summarization, as the final alignment model. The model achieved a strong performance of 91.5 recall and 84.2 F-1 on a held-out test set.

It is possible for SEW revisions to match none of its paired EW revisions if the SummaC model predicts that all pairs are unaligned. This occurs frequently, for example when a SEW page is written without being based on the relevant EW page. In total, matches occur for 133,744 page pairs, leading to a total of 145,161 revision-pair matches.

In Section 4, Wikipedia editors participating in SWIPE's annotation could flag samples they deemed unaligned. Of the roughly 5,000 annotated samples, just 4% were flagged as unaligned, validating the high precision of the matching process.

3.3 SWIPE Statistics

We focus the dataset on the introduction section of each Wikipedia page, as prior work has shown that including all sections leads to a large imbalance in terms of length (Xu et al., 2015).

The average compression ratio from EW to SEW page in SWIPE document pairs is 0.87, suggesting that SEW pages are not significantly shorter than

their EW matches. In fact, 26% of document pairs have a compression ratio larger than 1, indicating that is not infrequent for the simplification of a document to be longer than the original document.

3.4 Comparison with Prior Work

We perform an analysis of D-Wikipedia, an existing document-level simplification dataset that was created without considering the revision history and without filtering pages based on alignment quality.

We find that of the 132,546 samples in the training portion of D-Wikipedia, only 49,379 (or 37%) pass the alignment filtering we applied to create SWIPE. Models trained on noisy datasets due to low-quality alignment have been shown to exhibit undesirable behavior, such as hallucinating facts in summarization (Maynez et al., 2020; Kryściński et al., 2020), which is likely to occur in simplification as well. This analysis illustrates that matching revisions from the entire revision history is an essential step in creating large-scale, high-quality simplification datasets based on Wikipedia.

4 Edit-Level Annotation

In upcoming sections, we use the term *document* to refer to a particular page version. Given two matched documents, they can be represented as a single *alignment sequence* using a string-alignment algorithm such as Levenshtein (Levenshtein, 1966). An alignment sequence consists of a series of three operations: *unchanged text*, *inserted text*, and *removed text*, as illustrated in Figure 1. To understand the types of edits that occur in SWIPE, we collaborated with Simple Wikipedia editors to annotate a subset of the dataset.

4.1 Annotation Procedure Definition

The annotation procedure of a document pair consists of selecting groups of edit operations (i.e., insertions and deletions) and assigning them to an edit category from a predefined list. A document pair is considered fully annotated once each edit operation is assigned to at least one edit group.

Edit groups can consist of a single edit operation (e.g. the Background Elaboration in Figure 1), or multiple operations (e.g. four operations for the syntactic edit). Operations can be part of multiple groups, which enables group overlap (e.g., the second to last deletion in Figure 1 is part of Semantic Deletion and Discourse Reordering groups).

We choose to treat each operation as atomic and do not allow the annotator to manually split edit operations further. Although this could be limiting for longer edits, we believe this sets a common ground for annotation, as work in extractive QA has shown that disagreement of span boundaries affects dataset quality (Rajpurkar et al., 2016). Analysis in Section 4.4 examines the prevalence of overlap and interleaving of edits in the dataset.

Edit Category	N	%∃	#O	%I	%D	%I+D
● Lexical Edit	6798	61.7	2.1	0.3	0.2	**99.5**
● Entity Edit	359	6.4	1.5	7.2	**57.1**	35.7
● Sentence Split	3010	43.8	2.3	42.0	0.3	**57.7**
● Sentence Fusion	334	6.0	2.4	5.7	29.0	**65.3**
● Syntactic Deletion	1889	28.1	1.1	0.2	**98.1**	1.7
● Syntactic Generic	2615	36.2	1.5	31.1	27.8	**42.6**
● Reordering	2379	34.6	2.5	0.6	0.4	**99.0**
● Anaphora Resolut.	302	5.4	1.8	21.9	7.9	**70.2**
● Anaphora Insert.	362	6.4	1.8	20.4	0.6	**79.0**
● Elaboration - Bkgrd	805	12.9	1.4	**93.2**	0.4	6.5
● Elaboration - Exple	139	2.4	1.5	**95.7**	0.0	4.3
● Elaboration - Generic	3195	36.0	1.2	**95.9**	1.1	2.9
● Semantic Deletion	12928	76.8	2.0	0.4	**98.8**	0.8
● Specific-to-General	332	5.7	2.1	0.0	6.9	**93.1**
● Format	2688	35.3	1.9	9.7	10.5	**79.7**
● Noise Deletion	693	10.6	1.6	2.2	**93.7**	4.2
● Fact Correction	290	5.0	2.3	4.5	2.8	**92.8**
● Extraneous Info	3028	36.5	2.2	**99.4**	0.1	0.5
● Miscellaneous	241	3.6	1.7	**68.9**	1.7	29.5

Table 1: Edit categories in SWIPE. Categories belong to five classes: ● lexical, ● syntactic, ● discourse, ● semantic, and ● non-simpl. **N**: number of annotated instances, **%∃**: percentage of documents with the edit, **#O**: average group size, **%I, %D, %I+D**: distribution over operation type (insert-only, delete-only, replace)

4.2 Edit Categorization

Edit categories were formalized by combining prior-work categorizations (Siddharthan, 2014; Cardon et al., 2022). Three of the authors then iteratively annotated common samples in batches of 10-20 and introduced new categories specific to document-level simplification that did not arise in sentence-level-based work. We measured inter-annotator agreement at each iteration using Fleiss' Kappa and halted once no new category was introduced and the agreement level was above 0.7.

The final categories are organized into four higher-level classes: **Lexical** edits that simplify word units; **Syntactic** edits that simplify sentence structure; **Discourse** edits that deal with multi-sentence simplification; **Semantic** edits that add or remove information within the document. An additional class handles all **Non-Simplification** edits.

	In-Domain 71,702 cats.			OOD 3 cats.
	Train	**Valid**	**Test**	**Test**
Manually Annotated	3,861	484	484	377
Silver Annotated	126k	7k	7k	-

Table 2: Number of docs in each split of SWIPE

Each class is subdivided into categories, for a total of 19 categories. For example, the Syntactic class contains *Sentence Splitting*, *Sentence Fusion*, *Syntactic Deletion,* and *Syntactic Generic*. Classes and edit categories are listed in Table 1. A document with a definition and a canonical example of each category was prepared and later used to onboard annotators (Appendix B).

4.3 Annotation Collaboration

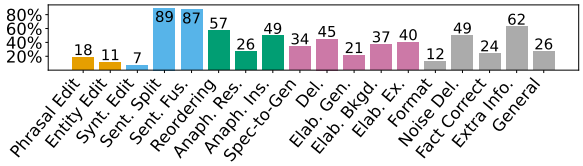
We collaborated with active Simple Wikipedia editors to annotate SWIPE. We contacted the 50 all-time top editors of Simple English Wikipedia on their public Wikipedia talk pages[3] with a high-level description of our project and prompted them to participate for a remuneration of US$25/hour.

In total, six SEW editors replied to the initial message. They were given a 1-hour onboarding task to attentively read through edit category definitions and annotate ten warm-up documents spanning all edit categories. The SEW editors were invited to join a Slack channel to discuss borderline and unclear examples. Upon completion, the authors of the paper reviewed the warm-up document annotations, and annotation errors were discussed with the participants before proceeding with the actual annotation.

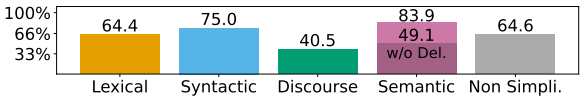
In total, 3 SEW editors successfully completed the onboarding, and we recruited an additional editor with a linguistic background recommended by one of the editors (not an active SEW editor). Over a period of two months, annotators identified edits in over 5,000 unique document alignment sequences. During the annotation process, annotations were periodically reviewed and feedback was given to annotators. Annotating a single sequence took an average of 1.3 minutes, and the annotation effort cost approximately US$2,500.

To inspect annotation quality, 329 alignment sequences were annotated by several annotators. The agreement level is measured using Fleiss' Kappa and averages 0.62 for the five category classes, indicating moderate agreement. Appendix C provides category-specific agreement levels, which vary across categories. Even with clear edit category definitions, there remains subjectivity in the annotation procedure.

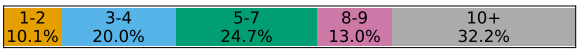
[3] https://en.wikipedia.org/wiki/Help:Talk_pages



(a) Percentage of edits that cross sentence boundaries



(b) Percentage of docs that include edits from category group



(c) Distribution of the number of edits per document

(d) Distribution of distinct category classes within documents

Figure 3: Summary of annotation analysis

Wikipedia categories are assigned to pages that identify page themes (e.g., Technology). In total, 71,705 Wikipedia categories appear in SWIPE. We set aside three categories – Materials, Economics, and Desserts – containing 377 pairs, which we fully annotated as a more challenging out-of-domain (OOD) test set. The rest of the annotation was performed on a random sample of all other categories. Table 2 summarizes the number of documents in each portion of the dataset.

4.4 Annotation Analysis

Figure 3 summarizes SWIPE annotations statistics. In Figure 3a, we break down the percentage of edits that cross sentence boundaries by edit category. Overall, 43% of edits are multi-sentence in nature, confirming that sentence-level simplification overlooks a large fraction of edits. This analysis likely undercounts multi-sentence edits, as anaphora and lexical consistency edits might be applied in a single sentence but require implicit document context.

Each category class occurs in 40-85% of document pairs (Fig. 3b). Semantic edits are most common due to the widespread Semantic Deletion category, with all other Semantic categories occurring in 49.6% of documents. On average, each annotated document has 15.2 edit operations

(6.3 insertions, 8.9 deletions), which are consolidated into 7.8 edit groups (see Figure 3c for the full distribution). Non-simplification edits, which correspond to undesirable edits related to formatting, the deletion of noise such as spam edits, or the introduction of extraneous information occur in 64.6% of document pairs, confirming the noisy nature of Wikipedia-based datasets. In Section 5.4, we explore an automated cleaning process to remove non-simplification edits.

To understand the diversity of edit categories that occur within each simplified document, we count how many of the four category classes occur jointly in simplified documents. The distribution is plotted in Figure 3d, revealing that a majority of annotated documents contain edits from three or four distinct category classes, confirming that SEW editors combine diverse editing strategies when simplifying EW pages into SEW pages.

We find that individual operations belong to a single group roughly 95% of the time, meaning that edit group overlap is rare, but find instances of operations belonging to up to 4 groups. Category pairs that overlap most often are (Reordering, Phrasal Edit) and (Reordering, Sentence Splitting).

In summary, the annotated portion of SWIPE reveals that all four category classes are prevalent on SEW, that at least 43% of edits require document-level context, and that producing SEW pages often requires combining edits from the full range of edit categories.

5 Automatic Edit Identification

We investigate whether edits can be identified automatically, which could automate annotation of the entire SWIPE dataset – estimated to require 2,900 hours of manual work – or facilitate analysis of generative simplification models.

5.1 Task Definition

The input to the edit identification task is a document pair's alignment sequence, which is composed of a series of edit operations (Figure 1); the task is to group (potentially overlapping) edit operations and assign each group to an edit category, matching the format of the annotations.

Evaluation is performed with four metrics. **Category F-1** and **Class F-1** evaluate the predicted categories (19 possible values) and associated higher-level classes (5 possible values) for each edit operation, irrespective of group. We use weighted,

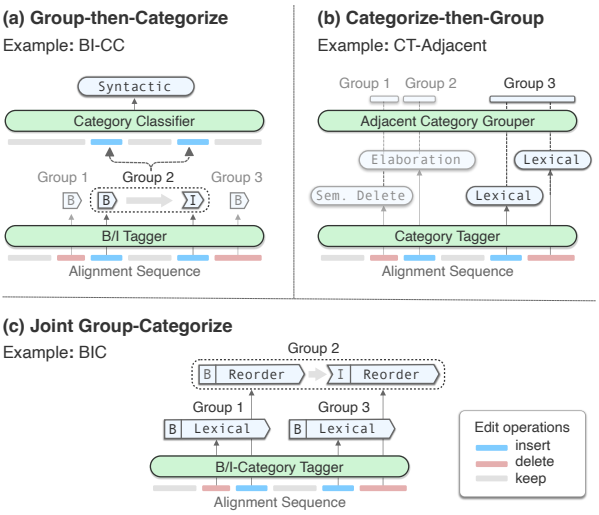


Figure 4: Overview of edit identification models

multi-label F1 since an edit operation may belong to multiple categories (e.g. for overlapping groups).

The other two metrics consider group assignment and category jointly. **%Exact** is the percentage of reference groups for which there is an identical group in the predictions. **%Partial** is the percentage of reference groups for which a predicted group of the same category has an operation set overlap of at least 0.5 Jaccard index.

5.2 Edit Identification Models

We implemented three varieties of edit identification models, illustrated in Figure 4 and described below. Additional details on model architectures are presented in Appendix D.1.

The *Group-then-Categorize* approach uses an initial grouper model to propose category-agnostic edit groups, and a second classification model to assign a category to each group (Figure 4a). We experiment with three grouper models. The *oracle* grouper uses the groups available in the annotations. The *adjacency* grouper applies the heuristic that adjacent edit operations (with no unchanged text between them) are within the same group. The *BI* grouper is a learned sequence-tagging model that segments edit operations into groups by outputting *B* (Beginning of group) or *I* (Inside of group) for each edit operation. In the next stage, each predicted group is passed to the *Category Classification (CC)* model; the input group is represented as an adjusted alignment sequence in which only the edit operations of the group are included. We refer to the three variants of this two-stage pipeline as **Oracle-CC**, **Adjacent-CC**, and **BI-CC**.

The *Categorize-then-Group* approach first predicts the category of each edit operation and then groups operations based on the predicted categories (Figure 4b). For the first stage, we propose *Category Tagger (CT)*, an NER-style sequence tagging model that takes as input a formatted alignment sequence and predicts one or more categories for each edit operation. For the second stage, we explore three grouper models: the *single* grouper performs no grouping, the *adjacent* grouper bundles adjacent edit operations of the same category, and the *rules* grouper applies category-specific rules detailed in Appendix D.1. By combining the stages, we obtain **CT-single**, **CT-adjacent**, and **CT-rules**.

In addition to two-stage models, we implemented two *joint* models that simultaneously group and categorize edit operations. **BIC** (Fig. 4c) is a sequence tagger that combines the label space of the **BI** and **C**ategory taggers; for each edit operation, BIC outputs one or more categories, each paired with a BI indicator for segmenting groups *within that category*. This category-specific BI notation supports richer forms of groupings, e.g., interleaved groups as illustrated in Figure 4c. The **Seq2seq** model is a fine-tuned sequence-to-sequence model that takes as input an XML-formatted alignment sequence and outputs an expanded XML in which edit categories and groups are identified. With all of the above models, we use RoBERTa-large (Liu et al., 2019) and BART-Large (Lewis et al., 2020) models for NLU and NLG components, respectively. Training details may be found in Appendix D.1.

The **Op Majority** baseline predicts the majority class for each operation type: *Semantic Deletion* for delete operations (54% of all deletions), and *Lexical* for insert operations (20% of all insertions).

5.3 Results

All models were trained on the training set of annotations, and hyperparameters were selected using the validation set. Table 3 summarizes experimental results on the in-domain test set.

Overall, the joint BIC model – trained to predict grouping and categories together – achieved the highest performance across the board, showing the benefits of joint over multi-step approaches. Appendix D.2 provides a category-specific breakdown of BIC model performance, revealing that the model excels at identifying edits of common categories (with top-5 F-1 performance coming in the seven most common categories), but struggles with less common categories.

Model Name	Cat F1	Class F1	%Part	%Exact
Op Majority	26.1	30.3	-	-
Adjacent-CC	56.7	60.4	48.2	50.8
BI-CC	64.4	67.8	56.4	60.0
Oracle-CC	78.2	81.4	-	-
CT-Single	69.7	**74.1**	27.8	27.8
CT-Adjacent	69.7	**74.1**	58.3	60.8
CT-Rules	69.7	**74.1**	58.4	62.1
BIC	**70.6**	**74.0**	**59.7**	**64.7**
Seq2Seq	51.3	55.4	42.5	45.7

Table 3: Edit identification results on in-domain test set

With the Group-then-Categorize models, as grouping quality increases, performance improves as well. When oracle groups are available, the categorization model achieves a 78.2 F-1 score at the category level, indicating that categorizing isolated edits is much less challenging than identifying overlapping edits in entire documents.

The Seq2seq model outperforms the majority baseline, but trails other models, showing that the added flexibility of generative modeling is not beneficial to edit identification in this case.

We report results on the out-of-domain test set in Appendix D.3. We do not observe a consistent performance drop on the unseen Wikipedia categories, giving evidence that most models generalize across categories. In Appendix D.3, we also benchmark the models' computational efficiency and find that BIC performs favorably compared to pipelined approaches and can process 18.9 documents per second on a single GPU, demonstrating another benefit of joint modeling.

5.4 Dataset Silver Annotation

We use the BIC model to automatically annotate all documents in SWIPE, identifying over one million edits, including more than 90,000 elaborations. Category-specific statistics are in Appendix C.

We refine SWIPE into a **cleaned** version by automatically reversing edits tagged in the Non-Simplification class. In Section 6, we determine whether models trained on the cleaned SWIPE are less prone to generating unwanted edits, such as ones including extraneous information.

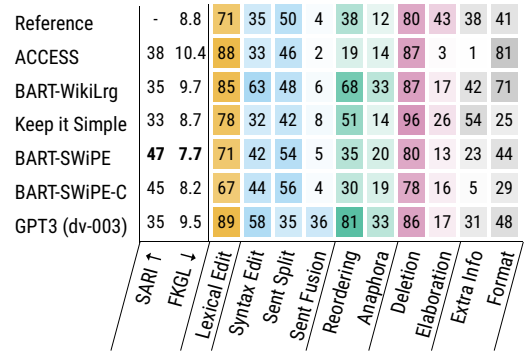


Figure 5: Analysis of generated simplifications: SARI, FKGL, and percentage of identified edit categories.

6 Text Simplification Baselines

We leverage SWIPE and its cleaned alternative to fine-tune two BART-large models: **BART-SWIPE** and **BART-SWIPE-C** and compare them to recent simplification systems. We experiment with two existing simplification systems: **ACCESS** (Martin et al., 2020), a state-of-the-art controllable sentence-level simplification model trained on Wikilarge (Zhang and Lapata, 2017), and **Keep it Simple** (KIS) (Laban et al., 2021), an unsupervised paragraph-level model optimized to produce lexical and syntactic edits. We also train **BART-Wikilarge** a BART-large model trained on Wikilarge to understand the effect of the dataset under a fixed pretrained model. Finally, we include a prompt-based **GPT3-davinci-003** using a task prompt that did not specify edit categories to apply. Model details and example outputs are in Appendix E.

We run experiments on the validation set of SWIPE. For each model, we report the n-gram-based SARI score (Xu et al., 2016), the Flesch-Kincaid Grade Level (Kincaid et al., 1975), and the distribution of edit categories identified by BIC (merged into 10 groups). Results are in Figure 5.

SWIPE-trained models achieve the highest performance in terms of SARI, confirming a similarity to reference simplifications, and the lowest estimated grade-level scores, validating the model's ability to improve readability.

The ACCESS sentence-level model performs moderately well on the SARI metric, but worst on the grade-level estimation, and makes fewer complex edits such as reorderings or elaborations, confirming that sentence-level models focus on simpler edits, such as lexical and syntactic edits.

All other models attempt a large proportion of all edits, including a large number of edits tagged as extraneous information (i.e., information not in the original document). When simplified by human editors, extraneous information often comes from other documents or background knowledge and is not likely harmful. On the contrary, recent NLG work has shown that model-generated extraneous information is often hallucinated, can be factually incorrect, and is undesirable. Example model outputs in Appendix E.2 show example problematic outputs from the KIS and BART-Wikilarge models which include factual errors, for example confusing centimeters and micrometers, or the length and width of a hair.

The KIS, BART-Wikilarge, BART-SWIPE, and GPT-3 models all produce a larger proportion of extraneous information edits than elaborations, confirming prior work showing that problematic hallucinations can occur for the simplification task as well (Devaraj et al., 2022). BART-SWIPE-C is able to produce elaborations while having a reduced rate of extraneous information, giving preliminary evidence that the edit-based dataset cleaning process we adopt can mitigate – but not solve – the generation of extraneous information.

Similar to recent work in summarization showing that zero-shot GPT3 can tie or surpass supervised models (Goyal et al., 2022; Liu et al., 2022), we observe that GPT3 can generate a wide range of simplification edits and does not mirror priors of the dataset – such as producing more sentence splits than fusions – indicating it has potential for use as a general-purpose simplification model. Similar to prior work, GPT3-based candidates score poorly on reference-based metrics.

We note that the analysis is preliminary, and future work should assess the efficacy, factual consistency, and simplicity of generated edits with target readers as done in prior work (Laban et al., 2021) to gain a thorough understanding of model performance.

7 Discussion & Future Work

Edit-Based Evaluation of Generators. In Section 6, we compare baseline simplification models based on the types of edits they produce. This analysis is based on automatically identified edits by the BIC model we trained, which likely includes errors. We expect that BIC's errors should affect all of the models' candidates equally, and should not significantly affect overall trends. More manual analysis is required to establish the effectiveness

of edits (i.e. whether the applied edits successfully simplify the document), as well as whether edits are factual and reflect the original document's content.

Extraneous Information in Simplification. In Section 5.4, we create a version of the SWiPE dataset where we remove edits that require extraneous information for a generation. We however choose to release the original dataset which includes those edits, as they could be valuable for future work, for example, approaches that might retrieve relevant documents prior to simplifying or to generate negative samples which can be used to stress-test models (Laban et al., 2022b).

Out-of-Domain Testing. We created an out-of-domain test set by selecting three Wikipedia categories that would be entirely isolated as a test set, to establish whether models would be capable of generalizing to unseen categories. In Section 5, we did not observe a meaningful gap in model performance between the in-domain and out-of-domain test sets, indicating that the Wikipedia categories we selected are not dissimilar enough from in-domain categories. Future work could explore other axes to create challenging out-of-domain test sets, for instance, based on page author identity, or publication time.

Edit-Based Models. In Section 6, we experiment with models that approach text simplification as a sequence-to-sequence model task and do not explicitly represent the editing process. However, recent progress in text-editing models (Malmi et al., 2022) could provide an avenue for better models in text simplification, which could be more efficient computationally and explainable in their generations. It is likely that text-editing models trained for sentence-level simplification (Malmi et al., 2019; Agrawal et al., 2021) can be expanded using SWIPE to generate a wider set of edits that can leverage document-level context.

Plan-then-Execute Models. Prior work in conditional generation tasks such as story generation (Martin et al., 2018), data-to-text generation (Puduppully et al., 2019), or summarization (Narayan et al., 2021) have decomposed the task in two steps, involving first the generation of a high-level plan, followed by an execution step that generates the output conditioned on the desired plan. The SWIPE resource can enable such research in the field of simplification, as the precise edit-based annotations we collected can serve as a basis for a plan to condition a generation model on. Plan-then-execute models enrich models with an intermediary representation that can be modified by a potential user, enabling customizable simplification applications.

Towards Practical Simplification. Practical implementations of text simplification, such as the news website Newsela (Xu et al., 2015) which simplifies the news to make it accessible to multiple grade-level tiers, require document-level understanding and editing. We hope the SWIPE dataset and models can play a part in making textual content more accessible, for example by improving access to scientific documents (August et al., 2022) or news coverage diversity (Laban et al., 2023).

8 Conclusion

We introduce SWIPE, a large-scale document-level simplification dataset based on Wikipedia. SWIPE is created by collecting pairs of pages from the English and Simple English Wikipedia and matching their revision histories to build document pairs that align in terms of content presented. We collaborated with Simple Wikipedia editors to annotate 5,000 document pairs in SWIPE, finding that many complex edits that require document-level context such as elaborations frequently occur in the dataset. We experimented with the automatic identification of edits, finding that even though the task is challenging, some models are able to achieve performance above 0.7 F-1 at edit categorization, making them viable to analyze model-generated simplifications. An analysis of generative simplification models reveals that sentence-level models are limited in the types of edits they propose and that document-scoped models are likely to produce hallucinated content. Finally, a model fine-tuned on a cleaned version of SWIPE produces less extraneous content while continuing to generate complex edits, pointing towards simplification models that can generate complex yet factually consistent edits.

Acknowledgments

We would like to thank the Simple Wikipedia editors and other participants that participated in the data annotation that led to the creation of SWIPE.

9 Limitations

SWIPE focuses on the English language. Although it is possible that some aspects of the work – such as the edit categorization – might transfer to the study of text simplification in other languages,

we focus on the English language. As of the writing of this paper, there is no equivalent of Simple English Wikipedia for other languages on Wikipedia, and creating similar resources for other languages would require finding other resources.

Difficulty in Retracing Original Editing. By matching revisions of Wikipedia pages that are factually aligned, and working with SEW editors to annotate the edits, we attempted to match the process used to create the resource. It is however not possible to recruit all 5,000+ SEW editors and for some page pairs the annotations are another editor's best attempt to reconstruct the intended edits by the original editor.

Improving Annotation Reproducibility. The analysis we conduct in Section 4.2 reveals that our annotators achieve moderate agreement on samples repeatedly annotated. More detailed analysis reveals that agreement is generally strong from common edit categories such as Lexical Edits, semantic deletions, or sentence splitting, but is lower for more infrequent categories. Better training of annotators on tail categories could therefore likely improve annotation. We also found that discussion amongst annotators of a sample often led to eventual consensus. Therefore collecting multiple annotations per sample, and allowing for discussion when multiple interpretations occur could help improve annotation quality, but at an increased cost.

10 Ethical Considerations

The models and datasets utilized in the project primarily reflect the culture of the English-speaking populace. Gender, age, race, and other socioeconomic biases may exist in the dataset, and models trained on these datasets may propagate these biases. Text generation tasks such as simplification have previously been shown to contain these biases.

In our collaboration with Wikipedia Editors to produce the annotations for SWIPE, we ensured to remunerate the participants fairly ($25/hour), including for fully or partially completing the onboarding task. Participants could communicate with us to voice concerns, could work at their own pace, and choose to stop working on the project at any time. Finally, we ensured to anonymize the annotations by not including personally identifiable information in any version of the dataset (annotator identity is instead marked as `annotator1`, `annotator2`, etc.).

We note that the models we use are imperfect and can make errors. When interpreting our models' outputs, results should be interpreted not in terms of certainty but probability. For example, if one of the simplification models generates edits that introduce non-trivial information, it is possible for this information to be hallucinated and not factually correct. Model outputs should therefore be checked, or a warning that content was machine-generated should be given to the reading audience.

To build the SWIPE dataset, we relied on several datasets as well as pre-trained language models. We explicitly verified that all datasets and models are publicly released for research purposes and that we have proper permission to reuse and modify the models.

References

Sweta Agrawal, Weijia Xu, and Marine Carpuat. 2021. A non-autoregressive edit-based approach to controllable text simplification. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3757–3769.

Sandra M Aluísio, Lucia Specia, Thiago AS Pardo, Erick G Maziero, Helena M Caseli, and Renata PM Fortes. 2008. A corpus analysis of simple account texts and the proposal of simplification strategies: first steps towards text simplification systems. In *Proceedings of the 26th annual ACM international conference on Design of communication*, pages 15–22.

Fernando Alva-Manchego, Louis Martin, Antoine Bordes, Carolina Scarton, Benoît Sagot, and Lucia Specia. 2020. Asset: A dataset for tuning and evaluation of sentence simplification models with multiple rewriting transformations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4668–4679.

Tal August, Lucy Lu Wang, Jonathan Bragg, Marti A Hearst, Andrew Head, and Kyle Lo. 2022. Paper plain: Making medical research papers approachable to healthcare consumers with natural language processing. *arXiv preprint arXiv:2203.00130*.

Adriane Boyd. 2018. Using wikipedia edits in low resource grammatical error correction. In *Proceedings of the 2018 EMNLP Workshop W-NUT: The 4th Workshop on Noisy User-generated Text*, pages 79–84.

Rémi Cardon, Adrien Bibal, Rodrigo Souza Wilkens, David Alfter, Magali Norré, Adeline Müller, Patrick Watrin, and Thomas François. 2022. Linguistic corpus annotation for automatic text simplification evaluation. In *EMNLP 2022*.

Si-Chi Chin, W Nick Street, Padmini Srinivasan, and David Eichmann. 2010. Detecting wikipedia vandalism with active learning and statistical language models. In *Proceedings of the 4th workshop on Information credibility*, pages 3–10.

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.

Ashwin Devaraj, Iain Marshall, Byron C Wallace, and Junyi Jessy Li. 2021. Paragraph-level simplification of medical texts. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4972–4984.

Ashwin Devaraj, William Sheffield, Byron C Wallace, and Junyi Jessy Li. 2022. Evaluating factuality in text simplification. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7331–7345.

Camille Dutrey, Houda Bouamor, Delphine Bernhard, and Aurélien Max. 2010. Local modifications and paraphrases in wikipedia's revision history. *Procesamiento del lenguaje natural*, 46:51–58.

Jane Dwivedi-Yu, Timo Schick, Zhengbao Jiang, Maria Lomeli, Patrick Lewis, Gautier Izacard, Edouard Grave, Sebastian Riedel, and Fabio Petroni. 2022. Editeval: An instruction-based benchmark for text improvements. *arXiv preprint arXiv:2209.13331*.

Goran Glavaš and Sanja Štajner. 2015. Simplifying lexical simplification: Do we need simplified corpora? In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 63–68.

Tanya Goyal, Junyi Jessy Li, and Greg Durrett. 2022. News summarization and evaluation in the era of gpt-3. *arXiv preprint arXiv:2209.12356*.

Stefan Heindorf, Martin Potthast, Benno Stein, and Gregor Engels. 2015. Towards vandalism detection in knowledge bases: Corpus construction and analysis. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 831–834.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python.

Chao Jiang, Mounica Maddela, Wuwei Lan, Yang Zhong, and Wei Xu. 2020. Neural crf model for sentence alignment in text simplification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7943–7960.

Chao Jiang, Wei Xu, and Samuel Stevens. 2022. arxivedits: Understanding the human revision process in scientific writing. *arXiv preprint arXiv:2210.15067*.

J. Peter Kincaid, Robert P. Fishburne Jr., Richard L. Rogers, and Brad S. Chissom. 1975. Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel. Technical report, Naval Technical Training Command Millington TN Research Branch.

Wojciech Kryściński, Bryan McCann, Caiming Xiong, and Richard Socher. 2020. Evaluating the factual consistency of abstractive text summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9332–9346.

Philippe Laban, Tobias Schnabel, Paul Bennett, and Marti A Hearst. 2021. Keep it simple: Unsupervised simplification of multi-paragraph text. 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 6365–6378.

Philippe Laban, Tobias Schnabel, Paul N Bennett, and Marti A Hearst. 2022a. Summac: Re-visiting nli-based models for inconsistency detection in summarization. *Transactions of the Association for Computational Linguistics*, 10:163–177.

Philippe Laban, Chien-Sheng Wu, Wenhao Liu, and Caiming Xiong. 2022b. Near-negative distinction: Giving a second life to human evaluation datasets. In *Conference on Empirical Methods in Natural Language Processing*.

Philippe Laban, Chien-Sheng Wu, Lidiya Murakhovs' Ka, Xiang'Anthony' Chen, and Caiming Xiong. 2023. Designing and evaluating interfaces that highlight news coverage diversity using discord questions. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, pages 1–21.

Vladimir I. Levenshtein. 1966. Binary codes capable of correcting deletions, insertions and reversals. *Soviet physics. Doklady*, 10:707–710.

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.

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*.

Yixin Liu, Alexander R Fabbri, Pengfei Liu, Yilun Zhao, Linyong Nan, Ruilin Han, Simeng Han, Shafiq Joty, Chien-Sheng Wu, Caiming Xiong, et al. 2022. Revisiting the gold standard: Grounding summarization evaluation with robust human evaluation. *arXiv preprint arXiv:2212.07981*.

Eric Malmi, Yue Dong, Jonathan Mallinson, Aleksandr Chuklin, Jakub Adamek, Daniil Mirylenka, Felix Stahlberg, Sebastian Krause, Shankar Kumar, and Aliaksei Severyn. 2022. Text generation with text-editing models. *arXiv preprint arXiv:2206.07043*.

Eric Malmi, Sebastian Krause, Sascha Rothe, Daniil Mirylenka, and Aliaksei Severyn. 2019. Encode, tag, realize: High-precision text editing. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5054–5065.

Lara Martin, Prithviraj Ammanabrolu, Xinyu Wang, William Hancock, Shruti Singh, Brent Harrison, and Mark Riedl. 2018. Event representations for automated story generation with deep neural nets. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Louis Martin, Éric Villemonte De La Clergerie, Benoît Sagot, and Antoine Bordes. 2020. Controllable sentence simplification. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4689–4698.

Aurélien Max and Guillaume Wisniewski. 2010. Mining naturally-occurring corrections and paraphrases from wikipedia's revision history. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On faithfulness and factuality in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1906–1919.

Shashi Narayan and Claire Gardent. 2015. Unsupervised sentence simplification using deep semantics. In *International Conference on Natural Language Generation*.

Shashi Narayan, Yao Zhao, Joshua Maynez, Gonçalo Simões, Vitaly Nikolaev, and Ryan McDonald. 2021. Planning with learned entity prompts for abstractive summarization. *Transactions of the Association for Computational Linguistics*, 9:1475–1492.

Rani Nelken and Elif Yamangil. 2008. Mining wikipedia's article revision history for training computational linguistics algorithms. In *Proceedings of the AAAI Workshop on Wikipedia and Artificial Intelligence: An Evolving Synergy*, pages 31–36.

Ratish Puduppully, Li Dong, and Mirella Lapata. 2019. Data-to-text generation with content selection and planning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 6908–6915.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In *Conference on Empirical Methods in Natural Language Processing*.

Carolina Scarton and Lucia Specia. 2018. Learning simplifications for specific target audiences. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 712–718.

Timo Schick, Jane Dwivedi-Yu, Zhengbao Jiang, Fabio Petroni, Patrick Lewis, Gautier Izacard, Qingfei You, Christoforos Nalmpantis, Edouard Grave, and Sebastian Riedel. 2022. Peer: A collaborative language model. *arXiv preprint arXiv:2208.11663*.

Tal Schuster, Adam Fisch, and Regina Barzilay. 2021. Get your vitamin c! robust fact verification with contrastive evidence. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 624–643.

Advaith Siddharthan. 2014. A survey of research on text simplification. *ITL-International Journal of Applied Linguistics*, 165(2):259–298.

Neha Srikanth and Junyi Jessy Li. 2020. Elaborative simplification: Content addition and explanation generation in text simplification. *arXiv preprint arXiv:2010.10035*.

Neha Srikanth and Junyi Jessy Li. 2021. Elaborative simplification: Content addition and explanation generation in text simplification. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 5123–5137.

Sanja Štajner, Marc Franco-Salvador, Paolo Rosso, and Simone Paolo Ponzetto. 2018. Cats: A tool for customized alignment of text simplification corpora. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Elior Sulem, Omri Abend, and Ari Rappoport. 2018. Bleu is not suitable for the evaluation of text simplification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 738–744.

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.

Laura Vásquez-Rodríguez, Matthew Shardlow, and Sophia Ananiadou. 2021. The role of text simplification operations in evaluation.

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.

Wei Xu, Courtney Napoles, Ellie Pavlick, Quanze Chen, and Chris Callison-Burch. 2016. Optimizing statistical machine translation for text simplification. *Transactions of the Association for Computational Linguistics*, 4:401–415.

Xingxing Zhang and Mirella Lapata. 2017. Sentence simplification with deep reinforcement learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 584–594.

Yang Zhong, Chao Jiang, Wei Xu, and Junyi Jessy Li. 2020. Discourse level factors for sentence deletion in text simplification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9709–9716.

Zhemin Zhu, Delphine Bernhard, and Iryna Gurevych. 2010. A monolingual tree-based translation model for sentence simplification. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 1353–1361.

A Revision Matching

Given a single revision of a SEW page, the task objective is to identify revisions of the matching EW page that could have been used as a starting point by a Wikipedia editor.

To gain a better understanding of the task at hand, we manually annotated a subset of 2,000 revision pairs from the created dataset. Prior work for sentence-level alignment has shown a relationship between content alignment and shallow string alignment (such as Levenshtein distance). To determine whether string-alignment methods are adequate for document-level alignment, we annotated samples across the entire range of string-alignment similarities, annotating 200 document pairs in each 0.1 range of Levenshtein ratio between [0,1.0].

Revision pairs were annotated by the authors of the paper with binary `Aligned`/`Unaligned` labels. A document pair was assigned the `Aligned` label if all the information in the SEW document was mentioned in the EW document, or if any new information can be seen as a useful addition for the purpose of simplifying information present both in the SEW and EW pages. The most common reason for a document pair to be marked as `Unaligned` is when the SEW document contains additional sentences or paragraphs that provide information that does not directly assist the information on the EW page.

The annotated data were randomly split into training, validation, and testing splits (1400-300-300 examples). We experimented with a diverse set of zero-shot and supervised methods for the task of page-pair alignment prediction, which we briefly introduce below. For models that predict real-valued scores, we selected a threshold based on the best validation performance.

	Validation			Test		
Model Name	**P**	**R**	**F1**	**P**	**R**	**F1**
Majority	59.0	100.0	74.2	62.6	100.0	77.0
ΔPublish	60.1	97.7	74.2	63.3	96.8	76.6
Lev. Ratio	63.1	100.0	76.3	65.9	96.3	78.3
Partial Lev. R	64.9	97.2	77.8	66.7	94.2	78.1
Ent. Overlap	79.8	82.5	81.1	75.9	75.1	75.5
$\text{SummaC}_{\text{Doc}}$	77.0	92.7	84.1	77.9	91.5	**84.2**
Supervised	88.9	81.4	**85.0**	83.9	85.2	**84.5**

Table 4: Performance of models on the page-pair alignment task. Top-to-bottom: baselines, string alignment, NER, NLI, and supervised models. Precision, recall, and F-1 reported on validation and test sets.

Baselines. `Majority` always predicts the majority class (Aligned), and `ΔPublish` produces a score based on the difference in publication time of the two revisions.

String-Alignment. `Levenshtein Ratio` is the negated normalized Levenshtein distance, and `Partial Levenshtein Ratio` finds the longest common subsequence (LCS) between the two documents, and computes the LCS's Levenshtein Ratio, allowing penalty-free deletion/insertions at the extrema of either document.

Entity-based. `Entity Overlap` uses spaCy's NER model (Honnibal et al., 2020) to extract named entities from both revisions and computes the Jaccard index between the entity sets as a score, with the assumption that newly introduced entities can be a signal of new and unaligned information.

NLI-based. NLI models such as the SummaC model (Laban et al., 2022a) have been successfully adapted to semantic similarity tasks, such as factual inconsistency detection in summarization. We include $\text{SummaC}_{\text{Doc}}$ in our experiments.

Supervised. We finetune a `RoBERTa-Large` on the 1,400 training samples, and select the final model based on the checkpoint that achieved the highest F1 score of 82.8 on the validation set.

Table 4 summarizes results. The ΔPublish and Levenshtein-based methods only narrowly outperform the majority class baseline in terms of F1 performance, confirming recent findings on the limitations of string-based similarity measures (Jiang

et al., 2020). Entity Overlap performs moderately strongly on the validation set but fails to generalize on the test set. Finally, the NLI-based SummaC model and the supervised model both largely outperform other models, and both achieve test F-1 scores of around 84.

We select the SummaC model (Laban et al., 2022a) for the dataset creation process, as it achieves similar performance to the supervised model in terms of F1, but with a higher recall (and lower precision). We favor recall for this application, as it will lead to a potentially larger dataset. We note that this choice might come at the cost of some of the samples in the dataset not being high-quality matches.

B SWIPE Edit Definitions

Below is a reproduction of the definitions provided during the onboarding of annotators.

B.1 Introduction.

Edits can be attributed to one of four high-level goals:

- **Lexical** edits are focused on simplifying word units, replacing rare/technical terms – a single word or a phrase – with simpler/more familiar terms.
- **Syntactic** edits are focused on simplifying sentence units, simplifying the structure of a sentence, for example shortening sentences, or reordering clauses within a sentence.
- **Discourse** edits deal with multi-sentence-level understanding, for instance by making connections between sentences more explicit, or reordering content so that required information appears before advanced information.
- **Semantic** edits deal with the addition or removal of information to improve readability at the document level, for example through the deletion of information that is not needed for a preliminary understanding of a document, or elaborations that introduce needed background or practical examples to help a broader audience understand the document.

Any edit that does not fit any of the primary simplification goals is categorized as a Non-simplification. Other edits are typically artifacts of the dataset, for example, a fact correction in Wikipedia revisions, or format cleaning (change of spelling or capitalization).

We now give a definition of each edit. Annotators were additionally provided a canonical example of each category, which we omit in the paper, but will include upon publication on an open-source repository.

B.2 Lexical Edits

- **Lexical - Entity**. Any edit that specifically targets the simplification of an entity (person, organization, location) for example the removal of a person's middle name or the replacement of a scientific name with a common name.
- **Lexical**. Any edit that replaces a complex or technical word or phrase with a more common/simple/accessible word or phrase. If the target phrase is a named entity, then the edit should be labeled with the more specific **Lexical - Entity**.

B.3 Syntactic Edits

- **Sentence Split.** An edit that leads to a single sentence being divided into two or more shorter sentences. In order for the split to be fluent, words are typically removed and inserted at the sentence boundary. If non-connector content is added, then it is not only a sentence split.
- **Sentence Fusion.** An edit that leads to several (two or more) sentences being merged into a single (potentially longer) sentence. Content is typically removed from original sentences to join the sentences fluently.
- **Syntactic Deletion.** An edit that deletes words in a sentence with the primary objective of compressing the sentence but does not remove information. If information is removed, see **Semantic - Deletion**.
- **Syntactic Generic.** An edit that modifies the syntax of the sentence, for example through re-ordering of clauses or changing verb tense.

B.4 Discourse Edits

- **Reordering.** An edit (or typically several edits) that re-orders content to improve narrative flow, for example moving up background content to ease comprehension. The re-ordering can happen within a single sentence, or across multiple sentences.

- **Anaphora Resolution.** An edit that replaces the repeated or implicit mention of an entity – typically a pronoun – with a resolved mention of the entity (i.e., that doesn't require prior context).

- **Anaphora Insertion.** An edit that replaces an explicit mention of an entity with an indirect mention, such as a pronoun. The pronoun is typically a short common, which can reduce sentence complexity by decreasing length and word complexity. Note: this is the inverse of the **Anaphora Resolution** edit.

B.5 Semantic Edits

- **Specific-to-General.** An edit that substitutes or removes low-level detail in exchange for a higher-level description (like replacing a city with its country). The detail deletion typically is judged as not essential and can be replaced by the higher-level portion. There must be a high-level content addition, otherwise, if it is only deletion, it is likely a **Semantic - Deletion**.

- **Elaboration - Background.** An edit that inserts content – a phrase or a full sentence – adding pre-requisite information for related content in the document. Typically, the background is inserted before the content it supplements.

- **Elaboration - Example.** An edit that inserts a concrete example of an abstract concept or phenomenon described in the document. Typically, the example is inserted after the content it concretizes.

- **Elaboration - Generic.** Any edit that adds information but cannot be categorized as a "Background" or "Example" elaboration. The insertion can be a phrase or a full sentence.

- **Semantic - Deletion.** An edit that removes content from the original document, typically because it is not essential to a simple comprehension of the document. The deletion can remove a part of a sentence or an entire sentence. Note that there can be many deletions within a single document, particularly when the original document is lengthy.

B.6 Non-Simplification Edits

- **Format.** An edit that modifies solely the formatting of the document, including punctuation, capitalization, spelling (for example UK to US spelling), or entity format (such as a date).

- **Noise Deletion.** An edit that fixes noisy content in the original document, such as a trailing partial sentence, or Wikipedia-specific formatting and jargon.

- **Fact Correction.** An edit that corrects a specific fact in the original document, most often updating the recency of the fact.

- **Extraneous Information.** Any edit that introduces facts that are not meant to simplify or add context to the information already present. Typically adds related but secondary information that is not needed in the simplified text. The insertion could be within a sentence or an entire sentence.

- **NonSim - General.** Any other edit that does not contribute to (Lexical, Syntactic, Discourse, Semantic) simplification, but does not fit in any other category.

C Agreement Level & Silver Statistics

Table 5 summarizes additional statistics of SWIPE. We find that the BIC model identifies edits at roughly the same rate as the manual annotation, with a few exceptions for long-tail categories such as Elaborations or Specific-to-Generic, this is due to low model recall on infrequent categories.

Overall, the class-level agreement level stands around 0.62, measured using Cohen's Kappa on 329 document pairs that were annotated by multiple editors. Table 5 provides category-specific Cohen's Kappa, with the main trend showing higher agreement for frequent categories (Semantic Deletion, Sentence Split, Lexical), and lower agreement for infrequent categories. The agreement level is particularly low for elaboration categories, however, when merging the three categories of elaborations into a super-category, we measure an agreement level of 0.4, indicating that some agreement exists at a coarser level. Future work can therefore choose to combine the elaboration categories to remove disagreement from the annotations.

Edit Category	Manual N	Manual %∃	Silver N	Silver %∃	κ
● Lexical Edit	6789	61.7	246k	62.0	0.62
● Entity Edit	359	6.4	9553	5.7	0.36
● Sentence Split	3010	43.8	93k	41.1	0.83
● Sentence Fusion	334	6.0	8141	4.6	0.34
● Syntactic Deletion	1889	28.1	45k	24.5	0.47
● Syntactic Generic	2615	36.2	65k	31.6	0.40
● Reordering	2379	34.6	75k	32.2	0.50
● Anaphora Resolut.	302	5.4	13k	7.6	0.30
● Anaphora Insert.	362	6.4	11k	7.2	0.73
● Elaboration - Bkgrd	805	12.9	1164	0.7	0.18
● Elaboration - Exple	139	2.4	139	0.1	0.05
● Elaboration - Generic	3195	36.0	91k	37.6	0.09
● Semantic Deletion	12928	76.8	343k	73.6	0.83
● Specific-to-General	332	5.7	1227	0.8	0.25
● Format	2688	35.3	82k	35.2	0.58
● Noise Deletion	693	10.6	14k	7.9	0.58
● Fact Correction	290	5.0	4581	2.4	0.37
● Extraneous Info	3028	36.5	105k	37.0	0.69
● Miscellaneous	241	3.6	1820	0.8	0.0

Table 5: Edit categories in SWIPE. For the manually and silver-annotated portions of the dataset, **N**: number of annotated instances, **%∃**: percentage of documents with edit, κ is Cohen's Kappa measuring inter-annotator agreement level

D Identification Models Supplemental

This Section provides the additional content related to Section 5 of the paper.

D.1 Model Specifics

We provide the implementation and training detail of each model included in the experiments of Section 5:

The **Category Classification (CC)** model, used in the Adjacent-CC, BI-CC, and Oracle-CC pipelined approaches is implemented as a finetuned RoBERTa-large model with a sequence classification head (i.e. a model that generates a single prediction for the entire sequence). The model was trained on a processed version of the training portion of SWIPE, in which each document pair was leveraged to create several samples, each based on a single group in the annotations. For each new sample, an adjusted alignment sequence is created by reverting all edit operations that are not part of the sample's considered group. The model receives the adjusted alignment sequence and must predict the category of the represented edit. Crucially, the CC model is expecting to see a single category per input alignment sequence and does not consider overlapping and multi-category edits. The model we use in experiments was trained with a batch size of 16, Apex half-precision, for seven epochs at a learning rate of 10^{-5}. The best checkpoint based on validation F-1 was selected, achieving a validation F-1 score of 77.5. We note that there's a crucial mismatch between train and prediction time in CC-based pipelines, as the CC model is trained on oracle groups, and at prediction time, certain configurations provide the model with imperfect groups (such as the Adjacent and BI groupers), which likely negatively affects performance. The training of the final model took roughly 1 hour on a single A100 GPU, and roughly 50 runs were conducted in iterations of model training.

The **BI** model, used in the grouping stage of the BI-CC model is a RoBERTa-large sequence tagging model that receives as input an alignment sequence and must predict for each edit operation whether the operation is at the beginning of (B) or inside (I) an edit group. We used an XML-like language to represent the alignment sequence for the model, using two operation starts (`<insert>` and `<delete>`) and two operation ends (`</insert>` and `</delete>`) which were added as special tokens to the model's vocabulary. The model was then trained to generate each operation's binary B/I tag at the corresponding beginning delimiter token. The model was trained using half-precision, and a learning rate of 10^{-5} for 10 epochs, selecting the model with the highest F-1 binary accuracy on the validation set of SWIPE. The training of the final model took roughly 25 minutes on a single A100 GPU, and roughly 20 training runs were conducted in iterations of model training.

The **Category Tagging (CT)** model, used in the first stage of the CT-Single, CT-Adjacent, and CT-Rules models, follows a similar architecture as the BI model described above, but outputs one of the 19 simplification categories for each edit operation instead of a B/I indicator. Additionally, CT uses a *multi-label* token-classification head to handle the case of multiple categories for an edit operation (e.g. for overlapping edit groups). For training, we used a batch size of 8 and a learning rate of 10^{-5} for 10 epochs. The final checkpoint was selected based on validation-set performance. The training of the final model took approximately 20 minutes on a single A100 GPU, and roughly 10 training runs were conducted in iterations of model training.

The **Rules** grouping method used in the second stage of the CT-Rules model, relied on category-specific statistics in the training portion of SWIPE. Categories were split into two sub-groups: contiguous and global. For each category, we analyzed the percentage of annotated of edits of the given category that were contiguous (adjacent) in their operation group. For each edit category, if a majority of annotated cases were contiguous, the edit category was labeled as *contiguous*, otherwise, it was labeled as *global*. For categories marked as contiguous, the model generated groups for predicted operation types based on contiguous boundaries (identical to the Adjacent grouping method), and all operations of a given global category were organized into a single group.

The **BIC** model uses an identical model architecture to the CT model described above, but expands the label space from 19 category labels to 57 joint category-BI labels. Specifically, for each category label <*cat*>, two additional labels are considered: <*cat*-B> and <*cat*-I>, indicating whether the operation is at the beginning or end of a group of this category. At training time, an edit operation is tagged with <*cat*> if the category is present and additionally with either <*cat*-B> or <*cat*-I> according to the operation's position within the annotated group. At inference time, the model outputs one or more of the 57 joint labels at each edit operation's start token. If <*cat*> is predicted for a given category, then the associated BI label is chosen based on whether <*cat*-B> or <*cat*-I> has the higher predicted probability. For training, we used a batch size of 8 and a learning rate of 10^{-5} for 10 epochs. The model checkpoint was selected based on validation-set performance. The training of the final model took approximately 20 minutes on a single A100 GPU, and roughly 15 training runs were conducted in iterations of model training.

The **Seq2seq** model was implemented based on a BART-large model that we fine-tuned on a seq2seq task using an XML representation of the alignment sequence. Example processing of the illustrative Figure 1 would be:

> Input: "The Mariinsky Theater is a <INS>very famous</INS> <DEL>historic</DEL> theater of opera and balet ..."
>
> Output: "The Mariinsky Theater is a <B;lexical>very famous</INS> <I;lexical>historic</DEL> theater of opera and balet ..."

As illustrated in the example, the model was trained to replace generic operation beginning tags with a joint tag representing the category and the BI tag of the operation. The vocabulary of the model was expanded to include the 38 tokens representing all combinations of (category x (B,I)) tags. The model was trained on the preprocessed data following a standard sequence-to-sequence formulation, with a batch size of 6, a learning rate of $2 * 10^{-5}$, for ten epochs, and the model with the lowest validation loss was selected as a final model. Training of the final model required roughly one hour of training, and roughly 20 training runs were conducted in iterations of model training.

Edit Category	N	Cat F-1	%Part	%Exact
● Semantic Deletion	12928	87.8	73.0	76.3
● Lexical Edit	6789	70.4	61.6	64.8
● Elaboration - Generic	3195	40.8	34.9	35.1
● Extraneous Info	3028	75.3	47.7	55.0
● Sentence Split	3010	83.5	55.6	69.9
● Format	2688	73.3	60.5	65.6
● Syntactic Generic	2615	70.7	63.0	63.3
● Reordering	2379	51.1	27.1	51.1
● Syntactic Deletion	1889	54.0	47.9	47.9
● Elaboration - Bkgrd	805	23.0	26.3	26.3
● Noise Deletion	693	61.1	48.7	48.7
● Anaphora Insert.	362	50.5	42.9	42.9
● Entity Edit	359	39.2	39.7	39.7
● Sentence Fusion	334	50.7	27.4	32.3
● Specific-to-General	332	17.2	15.9	15.9
● Anaphora Resolut.	302	62.7	57.1	57.1
● Miscellaneous	241	45.2	28.9	31.6
● Fact Correction	290	47.7	31.8	40.9
● Elaboration - Exple	139	11.1	16.7	16.7

Table 6: Breakdown of BIC model per edit category. Categories are sorted in order of frequency in the dataset, and we report the three metrics that can be computed at the category level. Categories belong to five classes: ● lexical, ● syntactic, ● discourse, ● semantic, and ● non-simplification.

D.2 BIC Performance Breakdown

Table 6 reports the performance of the BIC model, individualized by category. We find that performance generally improves on categories as the number of examples in the dataset increases, giving evidence that further annotations of tail categories could lead to improved performance of the BIC model.

Model Name	Cat F1	Class F1	%Part	%Exact	Doc/s
Op Majority	36.5	40.1	-	-	2.7k
Adjacent-CC	59.5	61.7	43.5	46.5	3.4
BI-CC	67.8	69.4	54.1	57.6	2.5
Oracle-CC	83.5	85.2	-	-	2.7
CT-Single	73.5	76.3	28.4	28.4	**23.3**
CT-Adjacent	73.5	76.3	58.6	61.8	23.2
CT-Rules	73.5	76.3	55.9	59.6	23.2
BIC	**74.9**	**76.6**	**57.3**	**62.1**	18.9
Seq2Seq	44.6	47.2	30.7	34.4	0.1

Table 7: Out-of-domain test set edit identification results. **Doc/s** reports the throughput of each model in documents per second.

D.3 Out-of-domain identification performance & model throughput

Table 7 presents the results analogous to Table 3 but for the out-of-domain test set. We do not observe a marked drop in performance, indicating that either the identification models are capable of generalizing to unseen Wikipedia categories, or that selected OOD categories are not truly out of distribution. We discuss the OOD test set selection further in the Limitations section.

We compute the throughput of each model to provide insights into the computational cost of identifying edits in document pairs. All models were benchmarked by the time they took to identify edits in the entire validation set (i.e., roughly 500 document pairs), using a single A-100 GPU on the same server, and we report normalized documents per second throughput (**Doc/s**). All models were tested at batch-size 1, which could disadvantage some neural methods. Results are summarized in the right-most column of Table 7. We find that the BIC model is the second-fastest neural method behind CT-based models, confirming that joint modeling of the edit identification task positively affects both performance and efficiency.

E Generation Models Supplemental

This Section provides the additional content related to Section 6 of the paper.

E.1 Model Specifics

The **ACCESS** model was implemented using the original author's public code release[4], and the default conditioning parameters of 0.95 for length target, 0.75 for Levenshtein target, and 0.75 for the word-rank target.

The **Keep-it-Simple** model was implemented using the original author's public model release on the HuggingFace model hub[5]. As recommended by the authors, we used a beam search (beam size of 4) to generate candidates, selecting the beam with the highest likelihood as the final generated candidate.

The **BART-SWIPE** and **BART-SWIPE-C** models were trained on the standard and cleaned versions of the SWIPE dataset, using a standard sequence-to-sequence framing, in which the model received the original document as an input, and was trained to generate the simplified document. We trained the models with a learning rate of $2 * 10^{-5}$, a batch size of six for three epochs, and selected the final checkpoint based on validation loss, which reached 1.12 for **BART-SWIPE** and 0.78 for **BART-SWIPE-C**. Training required 6-10 hours for each model, on a single A-100 GPU, and 5 runs were completed in the development of the models. At generation time, we used beam search (beam size of 4) to generate candidate simplifications.

The **GPT3-davinci-003** model was implemented using OpenAI's API access to the GPT3 model, with the following prompt: "Simplify the document below so it is accessible to a wider audience. Start of document:", with newlines inserted to delimit the task definition, the document, and the expected output. We used default generation parameters provided in the interface, and estimate the cost of generation at $10 for the 500 documents in the validation set. We note that it is unclear whether GPT3 qualifies as a zero-shot model for simplification, since it is trained on Wikipedia (amongst others), and has therefore been trained on a superset of the data in SWIPE, although it has not seen the explicit revision pairing available in SWIPE.

E.2 Example Generations

In Tables 8-9, we provide the revision of the Wikipedia page about the "Millimeter", included in the validation set of SWIPE. The Tables then provide the alignment sequence of six candidate simplifications: the human-written reference in Simple English Wikipedia, and the outputs of the ACCESS, Keep it Simple, BART-SWIPE, BART-SWIPE-C and GPT3-davinci-003 models.

[4] https://github.com/facebookresearch/access

[5] https://huggingface.co/philippelaban/keep_it_simple

Complex Document – English Wikipedia
The micrometre (International spelling as used by the International Bureau of Weights and Measures; SI symbol: μm) or micrometer (American spelling), also commonly known as a micron, is an SI derived unit of length equaling 1x10-6 of a metre (SI standard prefix "micro-" = 10-6); that is, one millionth of a metre (or one thousandth of a millimetre, 0.001 mm, or about 0.000039 inch). The micrometre is a common unit of measurement for wavelengths of infrared radiation as well as sizes of biological cells and bacteria, and for grading wool by the diameter of the fibres. The width of a single human hair ranges from approximately 10 to 200 μm. The first and longest human chromosome is approximately 10μm in length.
Reference – Simple English Wikipedia
The A micrometre (International (its American spelling as used by the International Bureau of Weights and Measures is micrometer ; SI symbol : μm is μm) or micrometer(American spelling), also commonly known as is a micron, is an SI derived unit of length equaling 1x10-6 of in the SI measurement system. It can also be called a metre(SI standard prefix" micro-" = 10-6); that micron. It is , one millionth of a metre (or one thousandth of a millimetre, 0.001 mm, or about 0.000039 inch). The micrometre is a common unit of measurement for wavelengths of infrared radiation as well as sizes of biological cells and bacteria, and for grading wool by the diameter of the fibres. The width of a single human hair ranges from approximately 10 to 200 μm. The first and longest human chromosome is approximately 10μm in length.
ACCESS
The micrometre(International spelling as used by the International Bureau of Weights and Measures; SI symbol: μm mm) or micrometer(American spelling), also commonly known as a micron, is an SI derived unit of length equaling 1x10-6 1x10 - 6 of a metre(SI standard prefix" micro-" = 10-6) ; , that is, one millionth of a metre(or one thousandth of a millimetre, 0.001 mm, 01 mm or about 0.000039 inch) about00039. The micrometre is a common unit of measurement for wavelengths of infrared radiation as well as sizes of biological cells and bacteria , and for grading wool by the diameter of the fibres. The width of a single human hair ranges is found from approximately about 10 to 200 μm 200 mm. The first and longest human chromosome is approximately 10μm about 10mm in length.
Keep it Simple
The micrometre (International is a common spelling as used by the International Bureau of Weights and Measures ; . It is also used by the SI symbol: μm) or micrometer(American spelling) , also commonly known as a which is used by all of the measures in the same way, and so is the micron, which is an SI derived unit of length equaling equivalent to 1x10-6 of a metre (SI standard prefix" micro-" = 10-6); that is, one millionth of a metre(or one thousandth of a millimetre, 0.001 mm, or about 0.000039 inch). The micrometre is a common unit of measurement for wavelengths of infrared radiation as well as sizes of biological cells and bacteria, and for grading wool by the diameter of the fibres. The width of a single human hair ranges from approximately 10 to 200 μm. The first and longest human chromosome is approximately 10μm in length.

Table 8: "Millimeter" Wikipedia page, reference simplification from SEW, and model outputs from the ACCESS and Keep it Simple models. Deletions are in red and insertions in blue. See Table 9 for other model outputs.

BART-WikiLarge
The micrometre (International spelling as used by the International Bureau of Weights and Measures; SI symbol: μm) or micrometer(American spelling), also commonly (also known as the micrometer) is a micron, is an SI derived unit of length equaling 1x10-6 that is less than 1/10,000 of a metre (SI standard prefix" micro-" = 10-6); that is, one millionth of a metre(or one thousandth of a millimetre, 0.001 mm, or about 0.000039 inch). The micrometre micrometer(I) is a common the smallest unit of measurement for wavelengths the number of infrared radiation as well as sizes different types of biological cells and bacteria microorganisms, and for grading wool by the diameter different sizes of the fibres cells. The width length of a single human hair ranges range varies from approximately 10 5 to 200 μm 20 cm(1 to 2 in). The first and longest human chromosome is approximately 10μm 4.7 in length, the second is 6.5 in length.
BART-SWIPE
The micrometre (International spelling as used by the International Bureau of Weights and Measures; SI (SI symbol: μm) or micrometer(American spelling), also commonly known as a micron, is an SI derived unit of length equaling 1x10-6 of a metre(SI standard prefix" micro-" = 10-6); that . It is , equal to one millionth of a metre(or one thousandth 1/1000 of a millimetre millimeter , 0.001 mm, or about 0.000039 inch). The micrometre is a common unit of measurement for wavelengths of infrared radiation as well as sizes of biological cells and bacteria, and for grading wool by the diameter of the fibres. The width of a single human hair ranges from approximately is about 10 to 200 μm. The first and longest human chromosome is approximately 10μm in length.
BART-SWIPE-C
The A micrometre (International spelling as used by the International Bureau of Weights and Measures; SI symbol: μm) or micrometer(American spelling), also commonly (also known as a micron, is an SI derived unit of length equaling 1x10-6 of a metre(SI standard prefix" micro-" = 10-6); that is, equal to one millionth of a metre (or one thousandth of a millimetre, 0.001 mm, or about 0.000039 inch). The micrometre It is a common unit of measurement for of wavelengths of infrared radiation as well as sizes of biological cells and bacteria, and for grading wool by the diameter of the fibres. The width of a single human hair ranges from approximately 10 to 200 μm. The first and longest human chromosome is approximately 10μm in length.
GPT3-davinci-003
The A micrometre (International spelling as used by the International Bureau of Weights and Measures; SI symbol (symbol : μm) or micrometer(American spelling), also commonly known as is a micron, is an SI derived unit of length equaling 1x10-6 of a metre(SI standard prefix" micro-" = 10-6); that is, one millionth of a metre (or , or one thousandth of a millimetre , 0.001 mm, or about 0.000039 inch (0.001 mm). The micrometre is a common unit of measurement for wavelengths It's used to measure the wavelength of infrared radiation as well as sizes , the size of biological cells and bacteria, and for grading wool by the diameter of the wool fibres. The width of a A single human hair ranges from approximately 10 to 200 μm. The first is around 10-200 μm wide, and the longest human chromosome is approximately 10μm about 10 μm in length.

Table 9: Model outputs from the two SWIPE-trained models and the GPT3 zero-shot baseline. Deletions are in red and insertions in blue.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Section 8

☑ A2. Did you discuss any potential risks of your work?
Both in Section 8 and Section 9 (Ethical Considerations)

☑ A3. Do the abstract and introduction summarize the paper's main claims?
The introduction is Section 1

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

We used several datasets and models in Section 2-6, and cited each one.

☑ B1. Did you cite the creators of artifacts you used?
We used several datasets and models in Section 2-6, and cited each one.

☒ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Left blank.

☑ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
In the Ethical Considerations section (Section 9)

☑ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
The data we collected was appropriately anonymized, and no personally identifiable information about the annotators is in the paper or future releases of the data (upon publication).

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
In the limitation section and ethical consideration (Section 8-9)

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
See Table 2, Table 3, and the analysis in Sections 3.3 and 4.4.

C ☑ Did you run computational experiments?

In sections 3, 5, and 6.

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
We reported all model sizes, average GPU use and GPU-type in the Appendix D and E.

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
We report final hyperparameters of each model in the paper in Appendix D and E.

☒ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
We did not include descriptive statistics of our results.

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
We report the python packages we used in relevant modeling sections.

D ☑ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Section 4.

☑ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
In Appendix B

☑ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
In Section 4.3

☒ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
We only curated data from Wikipedia, which has a permissive license.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Not applicable. Left blank.

☒ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
Participants were recruited for having a particular skill (Wikipedia editors), and we did not report their demographics.