On Operations in Automatic Text Simplification

Rémi Cardon
CENTAL, IL&C
UCLouvain, Belgium
remi.cardon@uclouvain.be

Adrien Bibal
University of Colorado
Anschutz Medical Campus, USA
adrien.bibal@cuanschutz.edu

Abstract

This paper explores the literature of automatic text simplification (ATS) centered on the notion of operations. Operations are the processed of applying certain modifications to a given text in order to transform it. In ATS, the intent of the transformation is to simplify the text. This paper overviews and structures the domain by showing how operations are defined and how they are exploited. We extensively discuss the most recent works on this notion and perform preliminary experiments to automatize operations recognition with large language models (LLMs). Through our overview of the literature and the preliminary experiment with LLMs, this paper provides insights on the topic that can help lead to new directions in ATS research.

1 Introduction

Automatic Text Simplification (ATS) is a natural language processing (NLP) task that consists in modifying a text in order to make it more readable or understandable. Generally, ATS systems work at the sentence leval. They take a sentence as an input and produce a modified version of it, with the objective of making it simpler for a given audience. To characterize the modifications that are performed or aimed at, a lot of different works established various sets of operations. For a broad definition, an operation is a change performed on a textual unit, for example the deletion of a clause or the reformulation of a complex expression with simpler terms. Simplifying sentences or documents typically involves more than one operation.

In this work, we investigate the ATS literature to gather what it says about operations. Indeed, while it is always present at every level of works on ATS, since the task appeared, operation as a concept has received little attention. Our first intention is to provide the community with a structured review of the literature centered on operations, in particular how and why they are used. We also hope to bring a new perspective to feed the current reflection on evaluation in ATS and ultimately on the definition of the task. We intend this paper to benefit both newcomers to the field – as we summarize elements from a large number of works of the domain – and active members of the community – as our observations enable new insights.

The contributions of this paper are the following:

- a detailed history and discussion of the role of operations in ATS;
- an overview of recently proposed typologies, along with a comparison and a discussion of the current role of operations in ATS;
- a review of the current means and goals to automate the annotation of operations;
- a preliminary experiment on the automation of linguistic operations identification using large language models.

In order to develop these contributions, the paper is organized as follows. We first report the definition of the different types of operations in the literature and how they are exploited (Section 2). Then we look closely at three recent papers that focus on typologies (Section 3). After that we address the question of automatic operation identification – why and how it is performed – and propose a preliminary experiment for the task with large language models (Section 4). We finally discuss our insights in Section 5 and the limitations of our work in Section 6, to finally conclude in Section 7.

2 Categorizing Operations in ATS

This section aims at giving a clear and detailed categorization of what is called "simplification operations" in the ATS literature (Section 2.1), and how they have been operationalized (Section 2.2).

2.1 What Operations Are

This section reports on the operations that are found in the literature. While surveying the literature, we did not find two identical sets of operations. In consequence, we do not attempt at producing an exhaustive catalogue of individual operations. There are two main objectives here: one is to clarify the principles that guide how operation sets can be put together, and the other one is to give a good view on what nuances can exist in the analysis and annotation of operations.

We divide the presentation using two broad types of operations: linguistically-based operations and string edits. In order to introduce the distinction between the two, consider Example (1) below, taken from the ASSET corpus (Alva-Manchego et al., 2020):

(1) Original: *Despite this, Farrenc was paid less than her male counterparts for nearly a decade.*
Simple: *Farrenc was paid less than her male co-workers for almost ten years.*

One operation in this example can intuitively be described as the deletion of the segment "*Despite this,*". The distinction between operation types depends on how this segment is characterized. On the one hand, linguistically-based operation types characterize the segment as a single linguistic unit. In this example, the operation may be described as the deletion of a sentence complement (its grammatical function) or of an adverbial phrase (its grammatical nature). On the other hand, string edits consider textual units as strings of individual tokens. In this example, most approaches that use string edits in fact describe this segment deletion as three operations: deletion of the token "*Despite*", deletion of the token "*this*" and deletion of the token ",".

ATS is largely focused on sentences, we mainly report on operations occurring within that level. As they appeared first in the literature, we start with linguistically-based operations (Section 2.1.1). We then move on to string edits (Section 2.1.2). We then report on operations described above the sentence level (section 2.1.3). Note: throughout this paper, we call linguistically-based operations "linguistic operations", operations on strings of tokens "edits", and we use "operations" to refer to any type of operation.

2.1.1 Linguistically-Based Operations

The very first works on ATS aimed at simplifying text as an input for other systems. In consequence, they were focused on the sentence structure, i.e. syntactic simplification (Siddharthan, 2014). The goal of these works was to reduce sentence complexity for downstream natural language processing (NLP) tasks, such as machine translation or information retrieval. Those approaches consist in manually designing simplification rules that modify constituency or dependency trees. An example of rule is the extraction of appositives (Chandrasekar et al., 1996), which is used to create two simple sentences from a complex one. In fact, this work concentrates on presenting two methods to only perform this specific operation. As syntactic operations can be the result of dependency or constituency trees, the linguistic elements they address can be denoted by their grammatical function (e.g., appositive, modifier, etc.) or their grammatical nature (e.g., relative clause, noun phrase, adjective, etc.).

With the appearance of works that focus on text simplification for human readers (which aim at improving readability or understandability), the scope of considered operations expanded. The operations can be syntactic and similar to the works mentioned above, such as recognizing a type of clause to delete or to extract in order to form a new simple sentence or to reorder sentence elements (Zhu et al., 2010). They can also be lexical, such as paraphrase or synonymy (lexical simplification has become a specific line of research and its details are out of scope of this paper, see Saggion et al. (2022) for more details). Operations can also occur at the morphological level, such as changing the mood or tense of a verb (Gala et al., 2020).

2.1.2 String Edits

The second type of operations is composed of operations applied to sentences considered as sequences of tokens. These operations are usually referred to in the literature as edits or string edits. They are considered at the token level and their name is self-explanatory. The operations that are always present in typologies of this kind are DELETE and ADD (also called INSERT). In order to account for all the token changes between two sentences, a (non-) operation is needed: KEEP. Depending on the goal for the operations in a given context, the list is adjusted. For instance, Alva-Manchego et al. (2017) introduce an operation called REWRITE, which they

define as "a special case of REPLACE where the words involved are isolated (not in a group of same operation labels) and belong to a list of non-content words". On occasions they can be considered at the n-gram level. It is the case in the calculation of SARI (Xu et al., 2016), for example. Contrarily to the linguistic type, these basic operations can be combined to form new operations. An example of this is REPLACE, which is sometimes described as an operation in itself, and sometimes as a combination of DELETE and ADD. Another one is MOVE, which is sometimes considered as a REPLACE where the deleted token is the same as the added one.

2.1.3 Operations Above the Sentence Level

At this time, there are not many works that address simplification above the sentence level in the literature. We report our findings for discourse, paragraph and document levels here.

Discourse A few works focus on simpflification at the discourse level. Wilkens et al. (2020) propose text simplification through coreference resolution. In their typology of operations, Gonzalez-Dios et al. (2018) introduce discourse-level operations: coreference resolution and change of discourse markers.

Paragraph-level Only one work can be found on paragraph simplification that mentions broad operation types (Devaraj et al., 2021). The operations described in this work are paraphrasing, word/sentence deletion, and summarization.

Document-level Sun et al. (2021) propose six operations, following Alva-Manchego et al. (2019b): sentence joining, sentence splitting, sentence deletion, sentence reordering, sentence addition, and anaphora resolution. In another work, Cripwell et al. (2023) mention copy, rephrase, split and delete as document-level operations. Laban et al. (2023) propose a dataset for document-level simplification where they also establish a typology of operations. Most of the operations of this typology are common sentence-level operations. They characterize the operations that involve adding or removing sentences under the "Semantic edits" category. Those three works have three very different approaches to describing operations related to document-level simplification.

2.2 How Operations Are Used

We now describe the operationalization of the operation types we identified in the previous section. We divide the presentation into four stages that usually occur in research works on ATS: data analysis or creation, system design, automatic evaluation and human evaluation.

2.2.1 Data Analysis and Creation

Often in the literature, researchers have analyzed the corpus they created or collected to indicate what they contain in terms of linguistic operations. This has been made for a variety of languages: Spanish (Bott and Saggion, 2014), Italian (Brunato et al., 2014, 2022), French (Koptient et al., 2019), German (Stodden et al., 2023), Brazilian Portuguese (Caseli et al., 2009), Basque (Gonzalez-Dios et al., 2018) and English (Amancio and Specia, 2014). In order to facilitate the annotation of operations, Stodden and Kallmeyer (2022) have proposed a dedicated tool. The transformation labels can be customized in the tool, with the default labels being *delete*, *insert*, *merge*, *reorder*, *split* and *lexical simplification*. The creators of the French corpus ALECTOR (Gala et al., 2020) used linguistic operations as guidelines for annotators to manually simplify texts. The result is a parallel document-level corpus. Cardon et al. (2022) built on existing typologies in order to study the ASSET test set (Alva-Manchego et al., 2020), a corpus made for the test and validation of ATS systems. They released the corpus with the annotated operations, called ASSET$_{ann}$. Several evaluation corpora (WikiSmall and WikiLarge (Zhang and Lapata, 2017), TurkCorpus, TurkCorpus (Xu et al., 2015), MSD (Cao et al., 2020), ASSET (Alva-Manchego et al., 2020) and WikiManual (Jiang et al., 2020)) have been analyzed in terms of string edits (Vásquez-Rodríguez et al., 2021b). While considering different types of operations, in their respective conclusions both Cardon et al. (2022) and Vásquez-Rodríguez et al. (2021b) make the case for caring about the distribution of operations in the datasets used in ATS.

2.2.2 System Design

Historically, linguistic operations were used as rules, as we mentioned in Section 2.1.1. In consequence, they were the heart of the definition of the task and the system design, i.e. ATS consisted in the application of precisely pre-defined operations. A lot of different rule-based approaches

have been proposed to do so, we refer the reader to Siddharthan (2014) and Saggion (2017) for more information. Rule-based approaches are still being explored today (Todirascu et al., 2022; Chatterjee and Agarwal, 2021; Evans and Orasan, 2019).

As manually crafting rules could be costly, another approach is to build a system that will learn operations on a corpus. A famous work using this approach is Woodsend and Lapata (2011). Their method, applied to Wikipedia data, uses a quasi-synchronous grammar to learn three types of rules based on constituency trees: syntactic rules, lexical rules and sentence splitting. Comparing their work to Zhu et al. (2010), they state that their model is "a more general model not restricted to specific rewrite operations" as an explanation of why it reaches better performance. We believe this statement epitomizes a turn in ATS research, where the presence of operations shift from the definition of the task (including system design) to the output of a model. The difference between this type of approach and the more recent neural approaches is that it produced explicit operations or rules, interpretable by humans. Neural models are expected to learn rules during training and apply them during inference (Nisioi et al., 2017; Štajner et al., 2022), but there is currently no identified way of accessing the operations that were learned.

Opaque neural models do not mark the complete disappearance of operations in task definition and system design in all ATS works. Some systems incorporate edits within a neural architecture (Alva-Manchego et al., 2017; Dong et al., 2019). More recently, a line of research has been focused on what has been called "controllable" text simplification (Martin et al., 2020; Maddela et al., 2021; Sheang and Saggion, 2021). The general idea is to prepend "control tokens" to the inputs to gain control on the ratio between the input and the output for a selection of attributes. Those attributes can be, for instance, sentence length, word frequency or syntactic tree depth. With this type of approach, operations are not made explicit, but the attributes influence their amount. For instance, variations to the sentence length ratio will have an impact on the amount of deletions.

2.2.3 Automatic Evaluation

Edits are present in the broadly used evaluation metric SARI (Xu et al., 2016). It counts the n-grams that were kept, added or deleted between the input and the reference(s) and between the output and the reference(s). An F1 score is calculated for each of the edits and each of the n-grams size (usually from 1 to 4) and the final score is the average of those scores. EASSE, the commonly used evaluation suite for ATS (Alva-Manchego et al., 2019a), reports the amounts of additions and deletions. Cardon et al. (2022) used linguistic operations to analyze the behavior of automatic metrics. SAMSA (Sulem et al., 2018) is an evaluation metric that evaluates the semantics of sentences that are the result of a split operation. More recently, Heineman et al. (2023) incorporate operation annotations in the training of a recent ATS metric, LENS (Maddela et al., 2023), and show that the metric gets more sensitive to their edit ratings. Automatic evaluation is a part of ATS that has started exhibiting promising perspectives for putting more thought on the integration of operations in ATS works.

2.2.4 Human Evaluation

The typical framework for the human evaluation of ATS outputs is to ask human judges to rate them according to three criteria, using 5-point Likert scales (Stodden, 2021). Yamaguchi et al. (2023) offer a method for analyzing ATS systems' outputs, according to simplification strategies and simplification errors. Cumbicus-Pineda et al. (2021) propose a structured framework for manually evaluating outputs according to the changes that were performed. Nisioi et al. (2017) asked two annotators to count the number of changes and state whether they are correct. In case of disagreement, a third annotator was asked to take a side. The type of change that was considered is not specified, the only information is that it can be applied at the phrase level and not only at the token level. Cooper and Shardlow (2020) established a 6-category typology of changes, some of them include both linguistic operations and edits.

3 Recent Advances on Simplification Typologies

In this section, we discuss in details the recent papers that are anchored in ATS and that focus mainly on observing the changes from an original sentence to its (attempted) simplification. We identified three such papers that we present chronologically in Section 3.1. After their presentation, we compare the three typologies in Section 3.2.

3.1 Typology Description

For each of the typologies we describe, we report the following information: the goal of the typology, the type of operations it contains and how many there are, the way it was built, the structure, the reasoning followed for annotation (if present in the original paper), the amount of inter-rater agreement, and finally the availability of guidelines and data.

Cardon et al. (2022). The main goal of the typology is to assess the content of a corpus. The authors explicitly mention that they cannot assess simplicity without the participation of members of a target audience, and that a detailed analysis of resources with linguistic operations can be used to select adequate data regarding the targeted application of a system. As stated in Section 2.2.1, this typology is composed of linguistic operations, which are inherited from past works on ATS corpora manual analysis in different languages. The authors added an "error" label to discard sentence pairs where the simplification is not grammatical or not semantically related to the original sentence. If used, no further annotation is performed. The authors present a structure for the rest of the operations (26 items), by mapping subsets to edits, namely deletions, additions and replacements. Other operations are described as too inconsistent to be mapped to edits, such as verbal voice change or transition from impersonal form to personal form. The authors also organize subsets that correspond to lexical and syntactic operations. A substantial inter-rater agreement is reported, with a trade-off between granularity and agreement. The annotation guide and the annotated data are available.

Yamaguchi et al. (2023). The main goal is the evaluation of ATS systems' outputs. the authors propose three different typologies: one for errors (4 items), one for content strategy (30 items), and one for surface strategy (22 items). The error set is composed of four labels "inappropriate deletion", "inappropriate addition", "inappropriate paraphrase" and "non-sentence". The other operation sets are built by the authors according to manual observations made in two stages. First they analyzed Newsela complex-simple sentence pairs (obtained after a manual alignment, as Newsela is not aligned at the sentence level) to produce a set of operations. Then, they added new operations by analyzing ATS systems' outputs. There are operations above the sentence level in this typology, such as "move a sentence" (within a document). During annotation, the first decision was to identify whether the operation under consideration is an error. If it is not, then a detailed decision tree is available for content and surface strategies. The decision trees were built by trial and error by two authors, and applied by the third one as a means of validation. The authors report a very high inter-rater agreement. The decision trees for content and surface strategies are available. As they used Newsela, the authors specify that the annotated data cannot be shared due to the terms of use.

Heineman et al. (2023). The main goal is the evaluation of ATS systems' outputs. This typology is structured in four parts: edit selection, information change, edit type classification and edit efficacy/severity rating. The first part is to identify whether the operation is an insertion, a deletion, a substitution, a reorder, a split or a structure change. The second part concerns the degree of semantic change divided into three categories: conceptual, syntactic and lexical. The authors present one category separately: grammar error, arguing that grammar and semantics are independent. For conceptual changes, there is a distinction between the operations that add information or the ones that remove information. Insertion is mapped to conceptual with more information, deletion is mapped to conceptual with less information. Reorder, split and structure change are mapped to syntax, and substitution can be mapped to three categories: conceptual with more information, conceptual with less information, and lexical. For each of these subcategories, a list of specific characterizations (there are 21 across all subcategories) is provided, which indicate a success (e.g., "elaboration" for a good insertion, or "generalization" for a good deletion), a failure (e.g., "bad deletion" for deletions, "information rewrite" or "complex wording" for a bad lexical edit). Some of these characterizations have the same name as a failure and as a success (e.g., "structure change" can be both). The authors report a general low inter-rater agreement that is broken down by edit type. It appears that the agreement is rather high for deletions and splits, and low for the other types. Examples are given for each individual fine-grained category. The authors state that they plan on releasing the data in the future, the paper being currently under review and available as a pre-print only.

3.2 Typology Comparison

For readability purposes, in this section we refer to the typologies as the first letter of the first author's name: C for Cardon et al. (2022)'s typology, Y for Yamaguchi et al. (2023)'s typology and H for Heineman et al. (2023)'s typology.

The three works use very different approaches for annotating the content of complex-simple sentence pairs. C adopts a classical approach based on existing works while Y and H propose a new framework. Y is the one with the most operations, and two detailed decision trees for annotation. The decision trees may explain the very high inter-rater agreement they obtained. Besides, Y is the most analytical and does not seem to leave much room for subjectivity, except for error characterization, while H states that instances could be annotated with several operations, as such they can be ambiguous. Error identification is the first step of C and Y, while H performs this characterization last. Y and H analyze errors at the operation level while C applies it to the whole sentence. The choice for a specific framework between those three should be driven by the type and granularity of information that is considered useful. H is the one with the least operations, the annotation process is clear and appears that it can be made quickly while giving an overview of what the differences are in complex-simple pairs. The room for ambiguity or subjectivity may impair reproductibility, while allowing for adaptation to different use cases. C is more detailed and clearly oriented towards linguistic operations. It can be adapted at different levels of granularity (e.g. grouping synonym, hyperonym and hyponym to one paraphrasing category). Y is the one that yields the more information, but also seems to be the most time-consuming.

All three works report different obstacles and limitations in the operation annotation task. Automating the task would facilitate this process of knowledge acquisition. In the next section, we propose to discuss the review the automation of operation annotation, as well as a preliminary experiment with large language models.

4 Automation of Operations Annotation

One interest of simplification typologies is to help understand and annotate the operations used to transform a complex sentence into one or more simpler sentences. In case of large corpora, it may be difficult to ask experts to annotate the operations involved in each transformation in the corpus. In such a case, it may be useful to automatically annotate the operations involved in all simplifications in the corpus.

This section proposes an overview of the current automation possibilities for the annotation of operations. Section 4.1 starts by presenting the currently used methods. Section 4.2 shows how large language models (LLMs) currently perform in this automation task.

4.1 Methods for Automatic Operation Annotation

As presented in Section 2.2, edits are now part of neural architectures and have been used to produce automated analyses of corpora. To achieve this, these edits need to be automatically identified. We report here how this is done in the literature, as the methods are varied. They often rely on the automatic alignment of tokens between two sentences.

Alva-Manchego et al. (2017) use the tool proposed by Sultan et al. (2014). Based on the alignments, they use heuristics to assign edit labels. To detect edits, Vásquez-Rodríguez et al. (2021a) and Vásquez-Rodríguez et al. (2021b) adapt the Wagner-Fischer algorithm – so that it can work at the token level instead of the character level – for alignment, and use heuristics to characterize the edits. EASSE (Alva-Manchego et al., 2019a) relies on MASSAlign (Paetzold et al., 2017) for alignment (or SimAlign (Jalili Sabet et al., 2020) as indicated in Alva-Manchego et al. (2021)), and heuristics for characterization. In EditNTS, Dong et al. (2019) implement their own neural-programmer interpreter to identify the edits.

Narayan and Gardent (2016) propose an approach that learns sentence splitting and phrase deletion. To do so, they rely on DRS (discourse representation structure (Kamp, 1984)) and graphs, using Boxer 1.00 (Curran et al., 2007), to produce those representations.

For linguistic operations, to the best of our knowledge nothing exists in the literature. One attempt at characterizing translation operations can be found in Zhai et al. (2019), which can be considered as a related task.

4.2 Prospect of Automation using LLMs

To the best of our knowledge, no work attempted to automatically annotate operations using large language models (LLMs). LLMs are not new in the literature. Indeed, first LLMs like GPT-1 (Radford

et al., 2018) and T5 (Raffel et al., 2020) have been present for a few years. In the scientific literature, the number of papers about large language models started to exponentially grow with the release of InstructGPT and ChatGPT (Zhao et al., 2023). Due to their ever increasing performance, LLMs offer a new avenue to solve machine learning and natural language processing problems.

4.2.1 Experimental Setup

In order to test the ability of LLMs to perform the task of annotating operations, we performed preliminary experiments with BLOOM (Scao et al., 2022), BLOOMchat (SambaNova Systems and Together Computer, 2023), GPT-2 (Radford et al., 2019), GPT-3.5 (OpenAI, 2022) and Bard (Manyika, 2023). It appeared that GPT-3.5 was the only LLM capable of providing outputs that were making sense for our task. Indeed, all other LLMs provided outputs that are not worth reporting here. The remainder of this section will therefore focus on the use of GPT-3.5 (more specifically GPT-3.5-turbo) with temperature frequency and presence penalties at 0.

The goal of the LLM is to annotate the operations used in a transformation using each of the three typologies presented in Section 3.1. In order to obtain appropriate results, many different prompts have been tried, with different formulations of the problem.

4.2.2 Prompts and Results

The question we explore in this work is: can the LLM annotate pairs of sentence with operations when a typology of operations is provided as a list? In the prompt, we sometimes included or excluded the mention that the sentences were in English, and included or excluded the explicit mention of "simplification". Concerning the formulation of the problem, we provide the full description of each typology and a complex-simple pair and ask the model to identify the corresponding operations. We show here the final prompt pattern that was used[1]:

> You are an expert in linguistics. I will provide you with a taxonomy of operations that can be performed on sentences. The taxonomy is composed of C categories: [list of categories].
>
> Here are the operations contained in the [1st category]:
> [list of operations]
>
> Here are the operations contained in the [2nd category]:
> [list of operations]
>
> [...]
>
> Here are the operations contained in the [Cth category]:
> [list of operations]
>
> Given the above taxonomy, what are the operations used to transform [sentence 1] into [sentence 2]?

Without annotated data from all taxonomies (see Section 3.1), we used examples provided in the papers to check if the LLM could reproduce the annotations provided as examples. Due to space constraints, we only show examples that we think are representative of the LLM behavior on this task.

In all cases, the LLM always uses the operations in the provided taxonomy. However, very often the operations proposed by the LLM differ from the ones provided as examples in the papers. It is worth noting that while some operations proposed by the LLM are different from typology's papers are undoubtedly incorrect, others are interesting and worth considering.

An example with both correct and incorrect annotations is in the following example (from Cardon et al. (2022)):

> From: "Meteora earned the band multiple awards and honors."
>
> To: "Meteora won the band awards."

The output provided by the LLM is

> [...]
> [2] 1. (C3) Delete proposition: Remove the proposition "earned" from the sentence.
> 2. (C4) Insert modifier: Add the modifier "the band" to specify which awards are being referred to.
> 3. (C5) Delete modifier: Remove the modifier "multiple" from the sentence.
> 4. (C7) Delete for consistency: Remove the word "and" to maintain consistency with the desired sentence structure.
> 5. (C9) Delete other: Remove the word "honors" as it is not mentioned in the desired sentence.
> 6. (C10) Replace with synonym: Replace the word "earned" with the synonym "won" to convey the same meaning.
> 7. (C14) Replace plural with singular: Change the word "awards" to its singular form "award" to match the desired sentence.
> 8. (C15) Replace segment with a pronoun: Replace the segment "Meteora" with the pronoun "it" to refer to the band.
> [...]

While indeed, some operations are correct (e.g., operation 3, as provided in example by Cardon et al.

[1] See Appendix A for the three complete prompts.

[2] In all reported outputs, we cut the introductory paragraph that rephrases the question that was asked and the final paragraph that summarizes the answer.

(2022)), others are undoubtedly incorrect (e.g., operation 7, as "awards" is plural in both sentences).

In other examples, some incorrect annotations are more subject to interpretation. This is the case, for instance, of the following example from Heineman et al. (2023) for Heineman et al.'s typology:

> From: "Éric Gauthier is also a novella author."
>
> To: "Éric Gauthier, famous for his soloist dancing career, is also a novella author."

The output provided by the LLM is

> [...]
> Conceptual Operation:
> (C1) Elaboration - Adding the information "famous for his soloist dancing career" to elaborate on Éric Gauthier.
>
> Syntax Operation:
> (S1) Word-level Reorder - Reordering the words "famous for his soloist dancing career" to come before "is also a novella author."
>
> Lexical Operation:
> No lexical operations are performed in this transformation.
> [...]

While the elaboration was indeed detected, considering the addition of "famous for his soloist dancing career" in the sentence as a word-level reorder and not as a lexical addition can be questioned.

In some cases, LLM's outputs are different from the examples provided in the typology's papers, but seem correct nonetheless. This is the case of the following example from Yamaguchi et al. (2023) for Yamaguchi et al.'s typology:

> From: "It shows you that hard work will help you reach your goals"
>
> To: "They show you that hard work will help you reach your goals"

The output provided by the LLM is

> [...]
> Surface Strategies:
> - Replacement:
> - (S2) Replace at word level: Replace "It" with "They"
>
> Content Strategies:
> - No content change:
> - (C5) Remain unchanged: No changes to the content of the sentence are required.
> [...]

While there is indeed a replace operation at the word level, the LLM also considers that no change in content is induced by the change of "It" by "They", while Yamaguchi et al. (2023) consider on their end that a change in content occurred through a paraphrase for adjustment.

While some of the operations in our experiments have correctly been identified, it is worth noting that a larger portion of operations were incorrectly annotated. A particular issue that was common to all the LLMs tested is their lack of stability. Indeed, it was often witnessed that trivial changes (e.g., adding a comma or removing an irrelevant word in the prompt) could lead to important changes in the LLM's output (i.e. a different annotation), even with temperature set to 0. This shows how difficult, but very important, prompt engineering is.

Based on our review and analysis of the recent typologies, their automation and the prospect of the use of LLMs for this automation, the next section proposes some elements of discussions that can open the literature to new directions.

5 Discussion and Perspectives

Shardlow (2014) wrote that "Simplicity is intuitively obvious, yet hard to define." This also seems to be true for simplification. Recent works on ATS evaluation (Cardon et al., 2022; Stodden, 2021; Alva-Manchego et al., 2021) show the community's perplexity as to how to assess successful simplifications. After the exploration of the literature presented in this paper, we would like to highlight an important observation: we did not find two works using the exact same set of operations. This is true for both linguistic operations and string edits. While we may have left out relevant papers, we are confident that finding identical typologies would be more of a coincidence than an indication of stability. This finding sheds light on the fact that there is no prototypical and consensual view on ATS as an NLP task, from which specific use cases would derive.

We believe that ATS could benefit from a structured framework for thinking of and manipulating operations. There are several perspectives we identified that could help build such a framework. First, operations typologies are mostly built on observations made on corpora. Those corpora are rarely produced by experts in simple writing or experts of potential target audiences. In consequence, what is called "simplification operation" is often an operation observed in a corpus that is used in a way or another for ATS. In their annotation framework, Heineman et al. (2023) ask humans to judge whether operations are relevant for simplification.

We think more work is needed for defining the criteria to distinguish between an operation that actually simplifies a text, and one that does not. Ultimately, while useful, operation sets are mostly built without a clearly defined grounding. Identifying operations that are relevant for a given target audience is a line of research that would be beneficial for making ATS systems available to end users. Rennes et al. (2022) show that while some concrete insights exist, little is still known in that area.

Another part of building a set of operations is the level of analysis. As we have seen in section 2, some operations can be described as belonging to different categories. This is for example the case of coreference/anaphora resolution, which has been positioned at the discourse level (Wilkens et al., 2020), the document level (Laban et al., 2023) or the sentence level (Cardon et al., 2022). Besides this specific example, other decisions can be whether to mix different operation types (linguistic and edits, categorizing grammatical function or nature), or whether considering different paradigms in which operations can overlap (e.g. syntax, discourse, semantics). We argue that those choices should be made knowingly.

On a more practical level, we believe that extending automatic operation annotation to all operation types would be beneficial to the domain. As we have seen, both edit-based and controllable architectures mark the return of operations in the system design. Current evaluation practices also leverage the automatic identification of string edits. Those uses of edits yielded improvements at several levels. However, linguistic operations are more akin to how humans conceive simplification. For example, when deleting a segment, humans do not work token by token but identify a segment and delete it at once. Enabling a reasoning on operation that is closer to the human one, on large amounts of data, would help interpretation of ATS systems' decisions and ATS evaluation metrics' scorings. Efforts towards automated linguistic operations could also help in data curation. It could expand the possibility of exploiting knowledge from experts of specific audiences' needs, as those are formulated as linguistic operations (Siddharthan, 2014; Rennes et al., 2022).

Another perspective is to analyze and structure in more depth the operations at levels above the sentence. As we saw in section 2, there are only a few works that present typologies at that level, yet they already exhibit great disparities.

6 Limitations of this Study

Our study comes with a set of limitations that are mostly focused on our preliminary experiments using LLMs.

First of all, while we experimented with 5 LLMs (BLOOM, BLOOMchat, GPT-2, GPT-3.5 and Bard), many other exist in the literature. For instance, every month, new LLMs appear in the top of the Hugging Face leaderboard [3].Determining if the task is completely solvable using LLMs therefore requires a thorough investigation of many of the existing LLMs.

Second, the lack of stability mentioned in Section 4.2 stresses the importance of prompt engineering to solve the task. While we tested many different prompts in several different configurations, one can never be sure that another untested prompt would not solve the task at hand.

Finally, the lack of access to the annotated corpora of the studied typologies made it difficult to evaluate the LLM on many examples and to provide quantitative results. A corpus containing ground truth annotations for all typologies on the same examples would allow to quantitatively evaluate the performance of LLMs.

7 Conclusion

This paper structured the ATS literature around the question of operations. Indeed, this overlooked angle led us to analyze recent typologies that have been proposed and to highlight their particular features, as well as the differences between them. We described what operations are found in the literature, how they are used and identified (manually and automatically) and provided insights that we hope can help spur new directions for research. In addition to a structured approach of the literature, we also proposed a preliminary experiment investigating the potential of large language models (LLMs) in the automatic annotation of operations. We show that albeit the new opportunities offered by LLMs, linguistic operations identification does not seem to be a trivial task.

We believe that this task may be an important one to address so as to have a better definition of the task, which would facilitate the implementation in real-world settings.

[3] https://huggingface.co/spaces/HuggingFaceH4/open_llm_leaderboard

8 Lay Summary

Automatic text simplification (ATS) systems take a text as an input and the output is expected to be a text with the same meaning, that is easier to read. Any change that is performed to transform the input into the output is called an operation. Operations are therefore the core of the simplification process. "Operation" is a generic term that can cover a variety of different phenomena. Some examples of linguistic operations are clause deletion, replacing a word by a more frequent one, or splitting a complex sentence into two simple ones. Operations can also be considered from the perspective of tokens. In that case token deletion, insertion and preservation are considered operations. Operations have been present in all the stages of ATS works, such as corpus creation and analysis, system design and system evaluation (human or automatic).

In this paper, we explore the literature in automatic text simplification from the perspective of operations. While they are always present in works on ATS, operations have rarely been the main focus of scrutiny by the community. Research on evaluation for ATS has gained traction recently, which involves the manual annotation of operations in ATS corpora or system outputs. We compare three different typologies produced in works on ATS evaluation and contrast them. We also perform preliminary experiments in order to check whether annotating with those three typologies is an easy task to automate, with LLMs.

Our findings expose an absence of stability in the sets of operations that are used in ATS, as there are no two identical ones in the papers we surveyed. Our comparison of the three recent typologies illustrates this absence of a common reference, in terms of defining, structuring and using operations. We find that automating linguistic operation annotation is not a trivial task. However, we believe facilitating the integration of such operations in system design and evaluation would enable new perspectives for ATS.

Our paper is intended for newcomers to the field, as a point of reference to have a better understanding of what operations are, how they have been used throughout ATS research. We believe that active members of the community can also find interesting insights, as the perspective of operations can bring interesting elements to the current reflection around ATS evaluation and how to tailor systems for end users.

Acknowledgments

We thank the anonymous reviewers for their valuable comments and suggestions. Rémi Cardon is supported by the UCLouvain through the FSR Incoming Fellowship Postdoc program. Adrien Bibal is supported by a Belgian American Educational Foundation (BAEF) grant.

References

Fernando Alva-Manchego, Joachim Bingel, Gustavo Paetzold, Carolina Scarton, and Lucia Specia. 2017. Learning how to simplify from explicit labeling of complex-simplified text pairs. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 295–305, Taipei, Taiwan. Asian Federation of Natural Language Processing.

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 Annual Meeting of the Association for Computational Linguistics*, pages 4668–4679.

Fernando Alva-Manchego, Louis Martin, Carolina Scarton, and Lucia Specia. 2019a. EASSE: Easier automatic sentence simplification evaluation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and the International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations*, pages 49–54.

Fernando Alva-Manchego, Carolina Scarton, and Lucia Specia. 2019b. Cross-sentence transformations in text simplification. In *Proceedings of the Workshop on Widening NLP*, pages 181–184.

Fernando Alva-Manchego, Carolina Scarton, and Lucia Specia. 2021. The (un)suitability of automatic evaluation metrics for text simplification. *Computational Linguistics*, 47(4):861–889.

Marcelo Amancio and Lucia Specia. 2014. An analysis of crowdsourced text simplifications. In *Proceedings of the 3rd Workshop on Predicting and Improving Text Readability for Target Reader Populations (PITR)*, pages 123–130, Gothenburg, Sweden. Association for Computational Linguistics.

Stefan Bott and Horacio Saggion. 2014. Text simplification resources for Spanish. *Language Resources and Evaluation*, 48:93–120.

Dominique Brunato, Felice Dell'Orletta, and Giulia Venturi. 2022. Linguistically-based comparison of different approaches to building corpora for text simplification: A case study on italian. *Frontiers in Psychology*, 13.

Dominique Brunato, Felice Dell'Orletta, Giulia Venturi, and Simonetta Montemagni. 2014. Defining an annotation scheme with a view to automatic text simplification. In *Proceedings of the Italian Conference on Computational Linguistics and of the International Workshop EVALITA*, pages 87–92.

Yixin Cao, Ruihao Shui, Liangming Pan, Min-Yen Kan, Zhiyuan Liu, and Tat-Seng Chua. 2020. Expertise style transfer: A new task towards better communication between experts and laymen. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1061–1071, Online. Association for Computational Linguistics.

Rémi Cardon, Adrien Bibal, Rodrigo Wilkens, David Alfter, Magali Norré, Adeline Müller, Watrin Patrick, and Thomas François. 2022. Linguistic corpus annotation for automatic text simplification evaluation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1842–1866.

Helena M Caseli, Tiago F Pereira, Lucia Specia, Thiago AS Pardo, Caroline Gasperin, and Sandra Maria Aluísio. 2009. Building a brazilian portuguese parallel corpus of original and simplified texts. *Advances in Computational Linguistics, Research in Computer Science*, 41:59–70.

R. Chandrasekar, Christine Doran, and B. Srinivas. 1996. Motivations and methods for text simplification. In *COLING 1996 Volume 2: The 16th International Conference on Computational Linguistics*.

Niladri Chatterjee and Raksha Agarwal. 2021. Depsym: A lightweight syntactic text simplification approach using dependency trees. In *Proceedings of the SEPLN Workshop on Current Trends in Text Simplification*.

Michael Cooper and Matthew Shardlow. 2020. CombiNMT: An exploration into neural text simplification models. In *Proceedings of the Language Resources and Evaluation Conference*, pages 5588–5594.

Liam Cripwell, Joël Legrand, and Claire Gardent. 2023. Document-level planning for text simplification. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics*, pages 993–1006. Association for Computational Linguistics.

Oscar M Cumbicus-Pineda, Itziar Gonzalez-Dios, and Aitor Soroa. 2021. Linguistic capabilities for a checklist-based evaluation in automatic text simplification. In *Proceedings of the SEPLN Workshop on Current Trends in Text Simplification*.

James Curran, Stephen Clark, and Johan Bos. 2007. Linguistically motivated large-scale NLP with C&C and boxer. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 33–36, Prague, Czech Republic. Association for Computational Linguistics.

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

Yue Dong, Zichao Li, Mehdi Rezagholizadeh, and Jackie Chi Kit Cheung. 2019. EditNTS: An neural programmer-interpreter model for sentence simplification through explicit editing. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 3393–3402.

Richard Evans and Constantin Orasan. 2019. Sentence simplification for semantic role labelling and information extraction. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, pages 285–294, Varna, Bulgaria. INCOMA Ltd.

Núria Gala, Anaïs Tack, Ludivine Javourey-Drevet, Thomas François, and Johannes C. Ziegler. 2020. Alector: A parallel corpus of simplified French texts with alignments of misreadings by poor and dyslexic readers. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 1353–1361, Marseille, France. European Language Resources Association.

Itziar Gonzalez-Dios, María Jesús Aranzabe, and Arantza Díaz de Ilarraza. 2018. The corpus of Basque simplified texts (CBST). *Language Resources and Evaluation*, 52(1):217–247.

David Heineman, Yao Dou, Mounica Maddela, and Wei Xu. 2023. Dancing between success and failure: Edit-level simplification evaluation using SALSA. In *arXiv:2305.14458*.

Masoud Jalili Sabet, Philipp Dufter, François Yvon, and Hinrich Schütze. 2020. SimAlign: High quality word alignments without parallel training data using static and contextualized embeddings. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1627–1643.

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, Online. Association for Computational Linguistics.

Hans Kamp. 1984. *A Theory of Truth and Semantic Representation*, pages 1–42. De Gruyter Mouton, Berlin, Boston.

Anaïs Koptient, Rémi Cardon, and Natalia Grabar. 2019. Simplification-induced transformations: typology and some characteristics. In *Proceedings of the 18th BioNLP Workshop and Shared Task*, pages 309–318, Florence, Italy. Association for Computational Linguistics.

Philippe Laban, Jesse Vig, Wojciech Kryscinski, Shafiq Joty, Caiming Xiong, and Chien-Sheng Wu. 2023. SWiPE: A dataset for document-level simplification of Wikipedia pages. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10674–10695, Toronto, Canada. Association for Computational Linguistics.

Mounica Maddela, Fernando Alva-Manchego, and Wei Xu. 2021. Controllable text simplification with explicit paraphrasing. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3536–3553, Online. Association for Computational Linguistics.

Mounica Maddela, Yao Dou, David Heineman, and Wei Xu. 2023. LENS: A learnable evaluation metric for text simplification. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16383–16408, Toronto, Canada. Association for Computational Linguistics.

James Manyika. 2023. An overview of Bard: An early experiment with generative AI.

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

Shashi Narayan and Claire Gardent. 2016. Unsupervised sentence simplification using deep semantics. In *Proceedings of the 9th International Natural Language Generation conference*, pages 111–120, Edinburgh, UK. Association for Computational Linguistics.

Sergiu Nisioi, Sanja Štajner, Simone Paolo Ponzetto, and Liviu P. Dinu. 2017. Exploring neural text simplification models. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 85–91.

OpenAI. 2022. Introducing ChatGPT.

Gustavo Paetzold, Fernando Alva-Manchego, and Lucia Specia. 2017. MASSAlign: Alignment and annotation of comparable documents. In *Proceedings of the International Joint Conference on Natural Language Processing: System Demonstrations*, pages 1–4.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Evelina Rennes, Marina Santini, and Arne Jonsson. 2022. The Swedish simplification toolkit: – designed with target audiences in mind. In *Proceedings of the Workshop on Tools and Resources to Empower People with REAding DIfficulties (READI) at the Language Resources and Evaluation Conference*, pages 31–38.

Horacio Saggion. 2017. *Automatic Text Simplification*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers.

Horacio Saggion, Sanja Štajner, Daniel Ferrés, Kim Cheng Sheang, Matthew Shardlow, Kai North, and Marcos Zampieri. 2022. Findings of the TSAR-2022 shared task on multilingual lexical simplification. In *Proceedings of the Workshop on Text Simplification, Accessibility, and Readability (TSAR-2022)*, pages 271–283, Abu Dhabi, United Arab Emirates (Virtual). Association for Computational Linguistics.

SambaNova Systems and Together Computer. 2023. BLOOMChat: A new open multilingual chat LLM.

Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, et al. 2022. BLOOM: A 176b-parameter open-access multilingual language model. In *arXiv:2211.05100*.

Matthew Shardlow. 2014. A survey of automated text simplification. *International Journal of Advanced Computer Science and Applications(IJACSA), Special Issue on Natural Language Processing 2014*, 4(1).

Kim Cheng Sheang and Horacio Saggion. 2021. Controllable sentence simplification with a unified text-to-text transfer transformer. In *Proceedings of the 14th International Conference on Natural Language Generation*, pages 341–352, Aberdeen, Scotland, UK. Association for Computational Linguistics.

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

Regina Stodden. 2021. When the scale is unclear-analysis of the interpretation of rating scales in human evaluation of text simplification. In *Proceedings of the SEPLN Workshop on Current Trends in Text Simplification*.

Regina Stodden and Laura Kallmeyer. 2022. TS-ANNO: An annotation tool to build, annotate and evaluate text simplification corpora. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 145–155, Dublin, Ireland. Association for Computational Linguistics.

Regina Stodden, Omar Momen, and Laura Kallmeyer. 2023. DEplain: A German parallel corpus with intralingual translations into plain language for sentence and document simplification. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16441–16463.

Elior Sulem, Omri Abend, and Ari Rappoport. 2018. Semantic structural evaluation for text simplification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 685–696, New Orleans, Louisiana. Association for Computational Linguistics.

Md Arafat Sultan, Steven Bethard, and Tamara Sumner. 2014. Back to basics for monolingual alignment: Exploiting word similarity and contextual evidence. *Transactions of the Association for Computational Linguistics*, 2:219–230.

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

Amalia Todirascu, Rodrigo Wilkens, Eva Rolin, Thomas François, Delphine Bernhard, and Núria Gala. 2022. HECTOR: A hybrid TExt SimplifiCation TOol for raw texts in French. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 4620–4630, Marseille, France. European Language Resources Association.

Laura Vásquez-Rodríguez, Matthew Shardlow, Piotr Przybyła, and Sophia Ananiadou. 2021a. Investigating text simplification evaluation. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 876–882.

Laura Vásquez-Rodríguez, Matthew Shardlow, Piotr Przybyla, and Sophia Ananiadou. 2021b. The role of text simplification operations in evaluation. In *Proceedings of the SEPLN Workshop on Current Trends in Text Simplification*, pages 57–69.

Rodrigo Wilkens, Bruno Oberle, and Amalia Todirascu. 2020. Coreference-based text simplification. In *Proceedings of the 1st Workshop on Tools and Resources to Empower People with REAding DIfficulties (READI)*, pages 93–100, Marseille, France. European Language Resources Association.

Kristian Woodsend and Mirella Lapata. 2011. Learning to simplify sentences with quasi-synchronous grammar and integer programming. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 409–420, Edinburgh, Scotland, UK. Association for Computational Linguistics.

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

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.

Daichi Yamaguchi, Rei Miyata, Sayuka Shimada, and Satoshi Sato. 2023. Gauging the gap between human and machine text simplification through analytical evaluation of simplification strategies and errors. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 359–375.

Yuming Zhai, Pooyan Safari, Gabriel Illouz, Alexandre Allauzen, and Anne Vilnat. 2019. Towards recognizing phrase translation processes: Experiments on english-french. In *Proceedings of the International Conference on Computational Linguistics and Intelligent Text Processing*.

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, Copenhagen, Denmark. Association for Computational Linguistics.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. 2023. A survey of large language models. In *arXiv:2303.18223*.

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, Beijing, China. Coling 2010 Organizing Committee.

Sanja Štajner, Kim Cheng Sheang, and Horacio Saggion. 2022. Sentence simplification capabilities of transfer-based models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(11):12172–12180.

A Full Prompts for the Experiments

In this appendix, we show the prompts that we used for our experiments with gpt-3.5-turbo.

A.1 Prompt for Cardon et al.'s Taxonomy

You are an expert in linguistics. I will provide you with a taxonomy of operations that can be performed on sentences. The taxonomy is composed of two sets of operations: computational operations

and computational operation combinations.

Here are the operations contained in the set of computational operations:
(C1) Move
(C2) Insert proposition
(C3) Delete proposition
(C4) Insert modifier
(C5) Delete modifier
(C6) Insert for consistency
(C7) Delete for consistency
(C8) Insert other
(C9) Delete other
(C10) Replace with synonym
(C11) Replace with hyperonym
(C12) Replace with hyponym
(C13) Replace singular with plural
(C14) Replace plural with singular
(C15) Replace segment with a pronoun
(C16) Replace pronoun with its antecedent
(C17) Modify verbal features

Here are the operations contained in the set of computational operation combinations: (CC1) Active to passive
(CC2) Passive to active
(CC3) Part-of-speech change
(CC4) Split
(CC5) Merge
(CC6) To impersonal form
(CC7) To personal form
(CC8) Affirmation to negation
(CC9) Negation to affirmation

Given the above taxonomy, what are the operations used to transform [sentence 1] into [sentence 2].

A.2 Prompt for SALSA's Taxonomy

You are an expert in linguistics. I will provide you with a taxonomy of operations that can be performed on sentences. The taxonomy is composed of three categories: conceptual operations, syntax operations and lexical operations.

Here are the operations contained in the category of conceptual operations:
(C1) Elaboration
(C2) Generalization

Here are the operations contained in the category of syntax operations:
(S1) Word-level Reorder
(S2) Component-level Reorder
(S3) Sentence Split

Here are the operations contained in the category of lexical operations:
(L1) Structure Change
(L2) Paraphrase
(L3) Insertion
(L4) Deletion

Given the above taxonomy, what are the operations used to transform [sentence 1] into [sentence 2].

A.3 Prompt for Yamaguchi et al.'s Taxonomy

You are an expert in linguistics. I will provide you with a taxonomy of operations that can be performed on sentences. The taxonomy is composed of two set of strategies: the first set contains the surface strategies and the second set contains the content strategies.

The surface strategies are categorized into 7 categories of operations: "replacement", "deletion", "addition", "integration", "splitting", "move" and "no transformation". Here are the operations contained in each of these 7 categories:
- Replacement:
(S1) Replace at punctuation level
(S2) Replace at word level
(S3) Replace at phrase level
(S4) Replace at clause level
(S5) Replace at sentence level

- Deletion:
(S6) Delete at punctuation level
(S7) Delete at word level
(S8) Delete at phrase level
(S9) Delete at clause level
(S10) Delete at sentence level

- Addition:
(S11) Add at punctuation level
(S12) Add at word level
(S13) Add at phrase level
(S14) Add at clause level
(S15) Add at sentence level

- Integration:
(S16) Integrate two sentences
(S17) Integrate more than two sentences

- Splitting:
(S18) Split by phrase
(S19) Split by clause

- Move:
(S20) Move constituents
(S21) Move a sentence

- No transformation:
(S22) Use an identical sentence

The content strategies are categorized into 5 categories of operations: ”no content change”, ”content deletion”, ”content addition”, ”content change” and ”document-level adjustment”. Here are the operations contained in each of these 5 categories:
- No content change:
(C1) Transform syntactic structure
(C2) Paraphrase into an abbreviation
(C3) Paraphrase into a non-abbreviation
(C4) Paraphrase into standard form
(C5) Remain unchanged

- Content deletion:
(C6) Delete introduction / conclusion
(C7) Delete a parallel element
(C8) Delete information for cohesion
(C9) Delete a modifier
(C10) Delete important information
(C11) Delete detail / extra information

- Content addition:
(C12) Add introduction / conclusion
(C13) Add a parallel element
(C14) Add contextual information
(C15) Add information for cohesion
(C16) Add a modifier
(C17) Add detail / extra information

- Content change:
(C18) Change aspect
(C19) Change modality
(C20) Paraphrase into a similar phrase
(C21) Paraphrase into an explanatory expression
(C22) Paraphrase into a direct expression
(C23) Paraphrase into a brief expression
(C24) Paraphrase into a concrete expression
(C25) Paraphrase into an essential point
(C26) Paraphrase into a different view

- Document-level adjustment:
(C27) Change information flow
(C28) Delete for adjustment
(C29) Add for adjustment
(C30) Paraphrase for adjustment

Given the above taxonomy, what are the operations used to transform [sentence 1] into [sentence 2].