Document-Level Planning for Text Simplification

Liam Cripwell, Joël Legrand, Claire Gardent

Document-Level Planning for Text Simplification

Liam Cripwell
Université de Lorraine
CNRS/LORIA
liam.cripwell@loria.fr

Joël Legrand
Université de Lorraine
Centrale Supélec
CNRS/LORIA
joel.legrand@inria.fr

Claire Gardent
CNRS/LORIA
Université de Lorraine
claire.gardent@loria.fr

Abstract

Most existing work on text simplification is limited to sentence-level inputs, with attempts to iteratively apply these approaches to document-level simplification failing to coherently preserve the discourse structure of the document. We hypothesise that by providing a high-level view of the target document, a simplification plan might help to guide generation. Building upon previous work on controlled, sentence-level simplification, we view a plan as a sequence of labels, each describing one of four sentence-level simplification operations (copy, rephrase, split, or delete). We propose a planning model that labels each sentence in the input document while considering both its context (a window of surrounding sentences) and its internal structure (a token-level representation). Experiments on two simplification benchmarks (Newsela-auto and Wiki-auto) show that our model outperforms strong baselines both on the planning task and when used to guide document-level simplification models.

1 Introduction

Text simplification aims to transform a given text into a simpler version of itself that preserves the core meaning such that it can be better understood by a wider audience (Gooding, 2022). Simplification has also been shown to be a useful preprocessing step for downstream NLP tasks such as relation extraction (Miwa et al., 2010; Niklaus et al., 2016) and machine translation (Chandrasekar et al., 1996; Mishra et al., 2014; Li and Nenkova, 2015; Štajner and Popovic, 2016).

Previous research has mostly considered the simplification of isolated sentences. Much work has focused on training a statistical or a neural model on pairs of complex and simplified sentences assuming that such models will learn to perform simplification operations (e.g. sentence splitting, lexical simplification or syntactic rephrasing) implicitly from the inductive bias present in the training data (Zhang and Lapata, 2017; Nisioi et al., 2017; Jiang et al., 2020). However, because the training data is obtained using distant supervision techniques and is often imbalanced in terms of simplification operations (many of which occur infrequently (Jiang et al., 2020)), system outputs have been found to be overly conservative, often making no changes or being limited to the paraphrasing of short word sequences (Alva-Manchego et al., 2017; Maddela et al., 2021). In addition, these systems provide limited capacity for controllability and are unable to express alternative variants of the simplified text (Alva-Manchego et al., 2017; Cripwell et al., 2021).

In response, controllable simplification systems have been proposed which either constrain attributes of the output (length, amount of paraphrasing, lexical and syntactic complexity) (Martin et al., 2020) or explicitly specify which simplification operation to perform (Alva-Manchego et al., 2017; Dong et al., 2019; Malmi et al., 2019; Scarton et al., 2020; Maddela et al., 2021; Cripwell et al., 2022).

To guide the simplification of full documents, we combine the power of data-driven neural generative models with strategies from controllable simplification. Our hypothesis is that document-level simplification can be facilitated by a plan specifying how each complex input sentence should be transformed to yield a simplified version of that document - should it be copied, deleted, split or rewritten?

We make the following contributions: We present a model for predicting document simplification plans which leverages both the context of sentences and their internal structure (the words they consists of). We create the data necessary to train this model by labelling complex sentences in simplification corpora with the simplification operation that relates it to the corresponding simplified sentence. We compare our planning model

Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics, pages 993–1006

with several alternative neural architectures and we briefly examine the impact of planning on document simplification.

Experiments on two simplification benchmarks (Newsela-auto and Wiki-auto) show that our model outperforms strong baselines both on the planning task and when used to guide document-level simplification models. [1]

2 Related Work

Document-Level Simplification. There is limited existing work on document-level text simplification. Early attempts largely applied sentence-level techniques iteratively over a document (Woodsend and Lapata, 2011a; Alva-Manchego et al., 2019b). However, this is generally viewed as insufficient for certain operations and maintaining the discourse coherence of the document (Siddharthan, 2003; Alva-Manchego et al., 2019b).

There are several works that address subproblems of simplification that only consider a limited set of operations, like paraphrasing and sentence re-ordering (Lin et al., 2021), insertion (Srikanth and Li, 2021) or deletion (Zhong et al., 2020; Zhang et al., 2022). Others fully address simplification but only extend inputs to the level of paragraphs without clearly differentiating the problem from the sentence-level (Laban et al., 2021; Devaraj et al., 2021).

Recently, Sun et al. (2020) proposed a sentence-level model (SUC) that uses an encoding of surrounding sentences as context information to influence the simplification. They use two extra encoders to build a representation of the two preceding and two following sentences, which are attended over in their encoder-decoder generative model. However, when applied to the document-level task, their system was unable to outperform any baseline systems (Sun et al., 2021).

Operation Prediction. Revision-based simplification models learn to predict edit operations to apply at the token-level rather than generating the entire simplification from scratch (Alva-Manchego et al., 2017; Dong et al., 2019; Kumar et al., 2020; Omelianchuk et al., 2021; Dehghan et al., 2022). This has the benefit of providing more control and interpretability over generative approaches, often at the cost of the ability to perform major structural changes. It also allows some systems to leverage non-autoregressive generation strategies, resulting in faster inference times (Malmi et al., 2019; Omelianchuk et al., 2021).

Some works have attempted to predict rewrite operations at the sentence-level. Applying a binary classifier to predict whether simplification should be performed has been found to improve SARI results, reducing conservatism and spurious transformations (Scarton et al., 2020; Garbacea et al., 2021). Others have proposed multi-class systems to predict sentence-level operations that are then used to condition a generative model (Scarton and Specia, 2018; Scarton et al., 2020; Cripwell et al., 2022). These show some capacity for general improvement over end-to-end systems, while also significantly improving performance for specific operations (e.g. splitting in the case of Cripwell et al. (2022)).

At the document-level, there has been limited interest to date. However, there are recent works specifically looking at predicting sentence deletions (Zhong et al., 2020; Zhang et al., 2022). Both of these use features of the discourse structure from surrounding sentences to identify likely deletion candidates.

We bring all of these methods together by proposing a system that uses both sentence and document-level information to predict a multi-class, sentence-level operation plan over an entire document.

3 Problem Formulation

Let C denote an English language document. The aim of document-level simplification is to produce a text S that simplifies the input document C.

As a plan can provide a high-level view of a document, we hypothesize that a document-level simplification model that is based on a plan specifying a simplification operation for each input sentence should fare better than a simplification model that directly simplifies an entire document.

We therefore decompose simplification into a two-stage generation process:

$$p(S \mid C) = p(S \mid C, P)p(P \mid C)$$

where input document $C = c_1 \dots c_n$ is a sequence of complex sentences, $S = s_1 \dots s_k$ is a sequence of simplified sentences and $P = o_1 \dots o_n$ is a sequence of sentence-level simplification operations for C.

[1] Pretrained models, code, and data are available at https://github.com/liamcripwell/plan_simp.

We consider three simplification operations proposed in previous work on sentence simplification (*copy*, *rephrase*, and *split*) to which we add *delete*, an operation that is needed to account for the fact that, contrary to sentence simplification, document-level simplification can require for a sentence present in the input document to be excluded from the resulting simplified document.

Given the input document C, the first-stage model aims to predict the sequence of simplification operations P that should be applied to each individual sentence in that document. The second-stage model generates the output simplified document S conditioned on the input document C and its accompanying simplification plan P.

In this work, we focus on the planning stage, comparing different architectures and demonstrating the impact of planning on three possible document-level simplification models. We leave the exploration of alternative, more complex architectures for the simplification stage to future work.

	Wiki-auto	Newsela-auto
# Doc Pairs	85,123	18,319
# Sent Pairs	461,852	707,776
Avg. $\|C\|$	155.51	868.98
Avg. $\|S\|$	97.72	674.94
Avg. $\|c_i\|$	28.64	22.49
Avg. $\|s_i\|$	21.57	15.84
Avg. n	5.43	38.64
Avg. k	4.53	42.60

Table 1: Statistics of each dataset after preprocessing, where n is # sentences in C and k is # sentences in S.

4 Data

In this section we introduce the datasets used, explain how annotation is performed for each complex sentence and describe other preprocessing steps.

Dataset. For all experiments, we utilise Wiki-auto and Newsela-auto (Jiang et al., 2020), two datasets of English documents paired with their simplification. These datasets were derived from WikiLarge (Zhang and Lapata, 2017) and Newsela (Xu et al., 2015) by aligning the input document with the output simplification at both the sentence and the paragraph level.

WikiLarge gathers three simplification datasets which were automatically-collated from English Wikipedia and Wikipedia simple (Zhu et al., 2010; Woodsend and Lapata, 2011b; Kauchak, 2013).

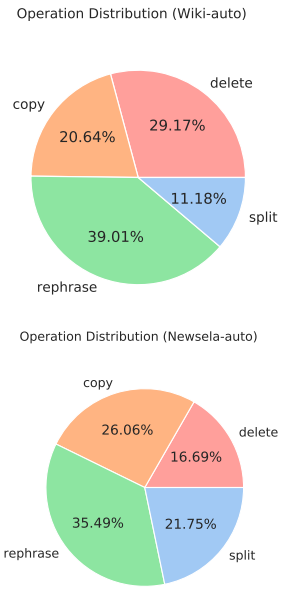

Figure 1: Operation class distributions for Wiki-auto (top) and Newsela-auto (bottom) datasets.

Newsela consists of news articles, each manually rewritten at five different levels of simplification, corresponding to discrete reading levels (0-4) of increasingly simplicity. Aligned pairs are created by pairing every article version with each other version corresponding to a higher reading level. Because of this, there can be up to four aligned document pairs that contain the same document as either the input or the output.

The types of operations present in different reading level pairings differs significantly, with adjacent level transitions being extremely conservative (no instances of deletion throughout entire dataset). To mitigate any issues arising from this, all models we train with Newsela-auto receive a control-token at the start of the input which specifies the target reading level.

We do not use the D-Wikipedia dataset from Sun et al. (2021) as it does not contain sentence/paragraph alignments and is poorly formatted. In particular, all text is lower-cased and pretokenized in a way that makes it difficult to accurately parse sentences. There are also regular formatting issues at points where references exist in the source article.

Annotating Complex Sentences. Using the pairs (c_i, s_j) of complex and simplified sentences available in Wiki-auto and Newsela-auto, we heuristically assign a silver simplification opera-

tion label to each complex sentence c_i in these two datasets as follows:

Delete: c_i is not aligned to any s_j.

Copy: c_i is aligned to a single s_j with a Levenshtein similarity above 0.92.

Rephrase: c_i is aligned to a single s_j with a Levenshtein similarity below 0.92.

Split: c_i is aligned to multiple s_js.

Preprocessing. Wiki-auto contains many document pairs with wildly different sizes. We therefore clip all complex documents after the last aligned paragraph. Many simple articles resemble a summarization, rather than a simplification of the complex article (lots of deletion, often consisting of about one sentence from each paragraph in C). Because of this, we also remove documents where more than 50% of aligned sentences are labelled as *delete*. Finally, we remove all articles that exceed 1024 tokens (so that we can fit them into a BART baseline generative model).

For Newsela-auto, article pairs are much more even in length as they are manually created to be gradual, direct simplifications of each other. We perform the same length-based filtering to exclude documents that will not fit into a baseline generative model.

Train/Dev/Test Split. For both datasets we use a train/validation/test split of 92.5/2.5/5. This is applied at the document-level so that sentences from the same document will not exist across different sets. For Newsela, this means that all reading level versions of a single article will exist within the same set.

Table 1 and Figure 1 give some statistics and a graphical description of the two datasets after pre-processing.

5 Planning

We present our model and four alternative models we explored for comparison. Training details are given in Appendix A.

5.1 Model (Contextual Classifier)

Given some input document $C = c_1 \dots c_n$ consisting of n complex sentences c_i, the task of the planner is to predict a sequence $\hat{P} = \hat{o}_1 \dots \hat{o}_n$ of n simplification operations with $\hat{o}_i \in \{copy, rephrase, split, delete\}$.

One challenge with this is that the operations have different, sometimes conflicting requirements.

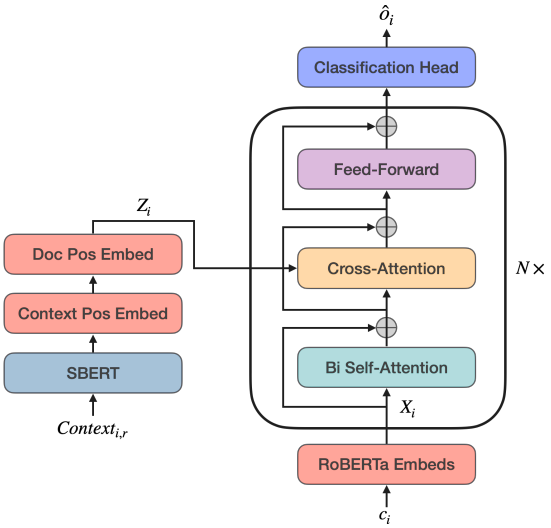

Figure 2: Contextual classifier model architecture.

By construction, splitting is mostly context independent as it is mainly determined by the input sentence's internal structure: a sentence will be split only if it has the appropriate syntactic (e.g., The man who sleeps snores → The man sleeps. He snores.) or discourse (e.g., John went shopping after he left work → John left work. Afterwards he went shopping.) structure. For sentence splitting, context (the other sentences in the input document) has little impact.

In contrast, deletion and to a lesser extent, copy and rephrase are mostly context dependent. Intuitively, a sentence can only be omitted in the simplified text in cases where it is either redundant with, or of minor semantic import relative to, other sentences in the document. That is, while for splitting, internal sentence structure is the key factor, for deletion, it is the semantics of the input sentence and how it relates to that of the other sentences which matters most.

We model these different requirements by using a token level encoder for the target document sentence c_i (the input sentence to be labelled with a simplification operation) and a sentence level representation of the context where each $c_p \in c_1 \dots c_{i-1}, c_{i+1} \dots c_n$ is represented by a sentence level embedding using SBERT. In this way both the internal structural information needed to capture splitting operations and the contextual information required by the other operations are provided. Specifically, we propose a model for planning that combines a classifier with cross-attention over the (dynamic or static) context and two types of positional embeddings. Figure 2 illustrates our model

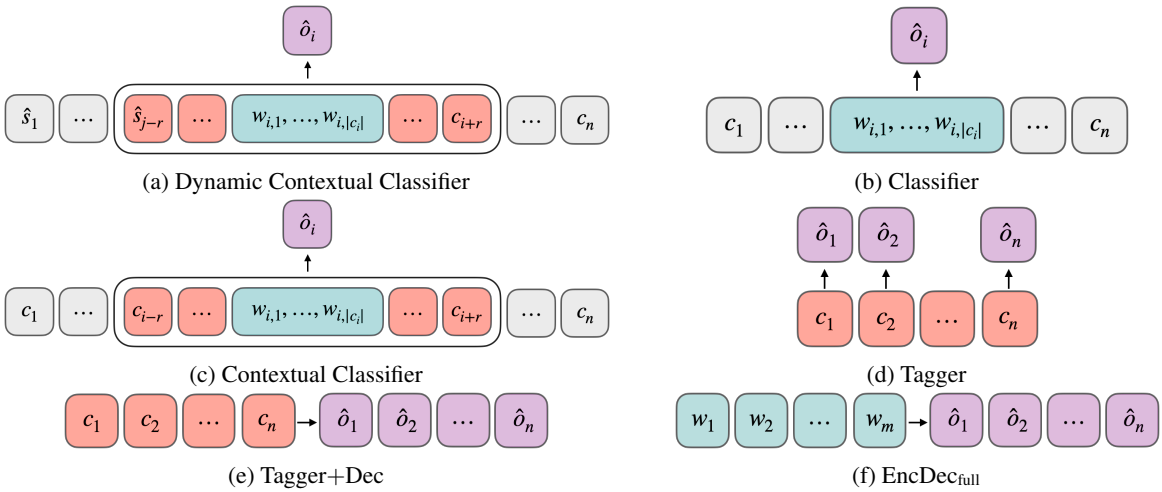

Figure 3: Visualisation of the inputs/outputs of the various models, where $w_{i,t}$ is the tth token in c_i, n is the no. sentences in C and m is the no. tokens in C. Sentence-level representations are shown in red, token-level representations in teal, operation labels in pink, and unused parts of C in grey.

architecture.

Classifier with Cross-attention over the Context. We build upon a RoBERTa classifier architecture to enable conditioning upon the surrounding sentences in the document. We do this by inserting an additional cross-attention layer between the self-attention and the feed-forward layer of each transformer block, allowing the model to attend to a latent representation of the surrounding sentences, Z_i.

Context Representation. To obtain Z_i, we take a fixed window of radius r, extract the r sentences on either side of the target sentence to be simplified and concatenate the representation of each of these sentences. Each context sentence is encoded with the pretrained Sentence-BERT (SBERT) model[2] (Reimers and Gurevych, 2019) and combined with custom learnt positional embeddings.[3]

To better simulate autoregressive inference, we consider a strategy where the left context consists of previously simplified sentences, rather than complex ones. We refer to this as *dynamic context*. At training time, we use the ground truth simplifications

$$Context_{i,r} = Concat(s_{j-r..j-1}, c_{i..i+r}) \quad (1)$$

where $j \in \{1, \ldots, |S|\}$ is the index of the first sentence aligned to c_i in the simple document S. During inference, the simplifications generated at preceding timesteps $\hat{s}_{j-r..j-1}$ are used.

Positional Embeddings. We use custom positional embeddings to encode both information about document, and relative context-window positions. These are each handled by a dedicated embedding layer and added to the representations of the corresponding context sentence.

Document positional embedding indices are simply the document quintile (1-5) that a given sentence falls into. We use quintiles as this will ensure that all indices are encountered within the input document. The context positional embedding indices are the relative distance of a given sentence from the input sentence c_i, adjusted to be within $\mathbb{N}_0$: $ContextPosIdxs = \{p - i + r \mid p \in \{i - r, \ldots, i + r\}\}$.

Initialisation. Given that the cross-attention layers must be trained from scratch, the start of training can see a lot of instability in the model, potentially making it more difficult to model context-independent features of the input sentence. To account for this, we initialise the RoBERTa layers with weights from a context-independent classifier.

5.2 Alternative Models

We compare our model with four alternative models. The different inputs/outputs of the models are illustrated in Figure 3.

Classifier. We fine-tune pretrained `RoBERTa-base` (Liu et al., 2019), which has 12 hidden layers and a hidden size of 768, and

[2]Specifically, all-mpnet-base-v2.

[3]At training time, we backpropogate to the positional embedding layers but keep the SBERT weights frozen.

	Wiki-auto						Newsela-auto					
Model	C	R	S	D	Micro	**Macro**	C	R	S	D	Micro	**Macro**
$\text{EncDec}_{\text{full}}$	26.9	42.2	36.0	51.8	43.2	40.8	26.1	10.8	11.7	9.0	12.2	11.5
Tagger+Dec	29.3	54.5	30.0	51.8	47.7	41.4	72.2	73.9	75.9	79.7	75.0	75.4
Tagger	38.6	54.2	31.7	**58.5**	50.6	45.8	71.4	72.7	74.1	78.4	73.7	74.1
Classifier	42.1	52.9	42.6	49.0	48.4	46.7	77.0	75.6	80.0	78.5	77.4	77.8
Dyn. Context	**44.8**	**57.9**	42.4	54.8	**52.8**	**50.0**	79.3	77.3	82.8	81.4	79.7	80.2
+ docpos	43.7	55.4	**43.6**	56.7	52.3	49.9	**80.0**	**78.1**	**83.6**	**82.0**	**80.3**	**80.8**

Table 2: Planning Accuracy. **Dyn. Context** is the contextual classifier described in Section 5.1 with $r = 13$, dynamic context and weights initialised using the classifier weights (C: Copy, R: Rephrase, S: Split, D: Delete).

add a pooled classification head which takes the final layer [CLS] representation as input. Given an input sentence c_i, the model simply takes the tokenized sentence as input and outputs a prediction score for each operation class. The model is applied from left-to-right on the input document classifying each sentence in turn. Thus, in this approach, while the model has access to the tokens of the sentence to be classified, there is no notion of context which, intuitively, should be detrimental in particular for deletions.

Tagger. We consider a model that frames the problem as a sequence tagging task over the full document, predicting the entirety of $\hat{P}$ at once. Each c_i is encoded using the same SBERT model as the contextual classifier, with the input document C therefore being represented as a sequence of sentence embeddings. In contrast to the classifier, the tagger makes predictions based both on the input sentence to be classified and on the context. However, because the input representation at each index is for an entire sentence we lose some resolution with respect to token-level content. The approach is thus less adapted for splitting.

Tagger+Dec. We also consider an autoregressive variant of the tagger that better models the dependencies between predicted tags. Here, we include a 1-layer decoder and condition each prediction both on the input document and on the previously predicted operation tags for the earlier sentences. This approach is somewhat similar to Dong et al. (2019); Malmi et al. (2019), except we abstract to the document-level and do not require explicit realisation, as this will be handled downstream by the simplication model.

$\text{EncDec}_{\text{full}}$. Finally, we experiment with an encoder-decoder variant that conditions on a token-level representation of the input, thereby combining a global view and a token-level representation of the input document. We use sentence separator tokens to delimit each sentence in the input document.

5.3 Evaluation Metrics

To evaluate the performance of the various planners we use F1-score, considering each individual prediction at the sentence-level. We report the F1 for each operation class as well as both the micro and macro averages. The micro F1 weights all examples equally, whereas the macro re-weights examples such that each class is represented equally in the final score. Given the class imbalances in the data, we regard macro F1 as our primary metric.

5.4 Results

Table 2 summarizes the results.

Compared to the various baselines, our model consistently shows best results on both datasets. The improvement over the context-free classifier is slightly less on Newsela-auto however. We conjecture that the much larger dataset and additional guidance provided by the reading levels allows the classifier to achieve rather high accuracy without document-level context. We also note that the context-free classifier is markedly outperformed by other models with respect to *delete*, which confirms the intuition that context modeling particularly matters for this operation.

Of the four baselines, $\text{EncDec}_{\text{full}}$ performs worst presumably because the very long input (the whole context is modelled at the token level) challenges the attention mechanism which tends to become blurry as the length of the input increases. This is particularly apparent on the longer Newsela documents.

The tagger models, which both use sentence-level encodings of the complex document, perform

Model	Copy	Rephrase	Split	Delete	Micro	**Macro**
(a) Ablation on Best Model						
Dyn, $r = 13$, +init, +docpos	80.0	78.1	83.6	82.0	80.3	80.8
-docpos	79.3	77.3	82.8	81.4	79.7	80.2
-init	74.9	72.1	77.8	75.2	74.6	75.0
-init, -docpos	75.6	72.0	77.7	77.1	75.1	75.6
(b) Dynamic vs. Static Context						
Stat, $r = 9$	71.3	69.5	75.4	73.3	72.0	72.4
Stat, $r = 13$	72.2	65.3	69.9	68.3	68.5	68.9
Dyn, $r = 9$	73.1	70.1	75.5	75.9	73.1	73.6
Dyn, $r = 13$	75.6	72.0	77.7	77.1	75.1	75.6
(c) With vs without Initialisation						
Dyn, $r = 9$	73.1	70.1	75.5	75.9	73.1	73.6
Dyn, $r = 9$ +init	79.3	78.0	82.7	79.8	79.7	80.0
Dyn, $r = 13$	75.6	72.0	77.7	77.1	75.1	75.6
Dyn, $r = 13$ +init	79.3	77.3	82.8	81.4	79.7	80.2
(d) Window Size						
Stat, $r = 9$	71.3	69.5	75.4	73.3	72.0	72.4
Stat, $r = 13$	72.2	65.3	69.9	68.3	68.5	68.9
Dyn, $r = 9$	73.1	70.1	75.5	75.9	73.1	73.6
Dyn, $r = 13$	75.6	72.0	77.7	77.1	75.1	75.6
Dyn, $r = 9$ +docpos	73.8	72.9	77.2	75.8	74.6	74.9
Dyn, $r = 13$ +docpos	74.9	72.1	77.8	75.2	74.6	75.0
Dyn, $r = 9$ +init +docpos	79.4	78.0	83.1	82.0	80.1	80.6
Dyn, $r = 13$ +init +docpos	80.0	78.1	83.6	82.0	80.3	80.8

Table 3: Ablations on Newsela-auto TestSet.

worse than the classifier. This highlights the importance of having a token-level modeling of the input sentences.

We observe a strong difference in terms of absolute scores between the two datasets. This is likely a result of Wiki-auto being an inferior simplification corpus (discussed in Section 4).

Next, we examine the impact of our modeling choices using ablation (Table 3) and focusing on the higher-quality, Newsela-auto dataset. Our best model is one with dynamic left-context, a context radius of 13, document position embeddings and weight initialisation. We see (Sub-table **a**) that each of these components help improve performance (document position appears less important with a larger context window). Sub-tables **b-d** show that using a dynamic rather than a static context increases results by up to +6.7 Macro F1, while increasing the context radius from 9 to 13 sentences mostly improves performance when dynamic context is used. Using document positional embeddings also generally improves results (Sub-table **d**).

6 Simplification

To assess whether document plans can help improve simplification models, we experiment with two simple document-level simplification models and compare their performance with and without a preceding planning step.

6.1 Simplification Models

All models use the BART model (Lewis et al., 2020) fine-tuned on aligned text pairs. [4]

We consider two variants for document-level simplification: (i) **Doc-BART**, which is finetuned on full document pairs; and (ii) **Sent-BART** which is finetuned on sentence pairs and iteratively applied to each input sentence at test time.

We compare these to various plan-guided (**PG**) systems whereby one of our planners predicts an $\hat{o}_i$ for each c_i and is given as a control-token to a sentence-level BART simplification model. In the case of the dynamic planner, $\hat{o}_i$ is predicted based on the sequence of previously simplified sentences $\hat{s}_{i-r} \ldots \hat{s}_{i-1}$.

Training details are given in Appendix B.

6.2 Evaluation

To measure meaning preservation and fluency, we use BARTScore (Yuan et al., 2021), a state-of-the-

[4] We use the pretrained *facebook/bart-base* model from https://huggingface.co/facebook/bart-base.

System	BARTScore ↑				SMART ↑			FKGL ↓	SARI ↑	Length	
	Faith. ($s \rightarrow h$)	P ($r \rightarrow h$)	R ($h \rightarrow r$)	F1	P	R	F1			Tokens	Sents
Input	-0.93	-2.47	-1.99	-2.23	63.2	62.7	62.8	8.44	20.52	866.9	38.6
Reference	-1.99	-0.93	-0.93	-0.93	100	100	100	4.93	99.99	671.5	42.6
Doc-BART	-2.48	-2.68	-2.76	-2.72	61.9	43.9	50.6	10.01	47.07	600.8	20.7
Sent-BART	**-1.86**	-1.63	-1.56	-1.60	78.9	80.1	79.3	5.03	73.02	666.4	42.6
PG_{Tag}	-1.95	-2.22	-2.18	-2.20	62.0	62.6	61.6	5.07	56.13	657.4	41.8
$PG_{Tag+Dec}$	-1.94	-2.22	-2.18	-2.20	62.2	62.5	61.6	5.09	56.06	654.2	41.4
PG_{Clf}	-1.91	-1.68	**-1.53**	-1.60	77.8	**81.2**	79.3	**4.95**	73.83	688.8	44.5
PG_{Dyn}	-1.91	**-1.60**	-1.54	**-1.57**	**80.2**	81.0	**80.5**	4.98	**75.00**	667.2	42.6
PG_{Oracle}	-1.93	**-1.39**	**-1.40**	**-1.40**	**85.5**	**85.0**	**85.3**	**4.91**	**80.74**	655.6	42.1

Table 4: Results of document simplification systems on Newsela-auto. For BARTScore, s is the source, h is the hypothesis, and r is the reference.

art summarization metric that has proved effective on many other text generation tasks. We also compute SMART (Amplayo et al., 2022), a new metric that considers sentences as the primary unit of comparison. It was shown to be highly effective for document summarization and does not use any neural model, making it very fast to compute (we use the SMARTL+CHRF version). We cannot use other model-based metrics, such as BERTScore or QuestEval, as these do not support texts longer than 512 tokens.

To assess simplicity, we use the Flesch-Kincaid grade level (FKGL), a document-level metric used to measure text readability, which has been found to have the highest correlation with simplicity measures of human-written simplifications (Scialom et al., 2021). We also report the popular SARI (Xu et al., 2016). The EASSE python library (Alva-Manchego et al., 2019a) is used for calculation of FKGL and SARI. We include results for other popular metrics in Appendix D.

At test time we generate sequences using beam search with a beam size of 5 and a maximum length of 1024 tokens. We enforce a minimum length for Doc-BART, which is tuned on the validation set.

We do not conduct a human evaluation as we intend the focus of this work to be on the planning component and include simplification results only to confirm its efficacy. We leave a more in-depth investigation of the interaction between planning and document-level simplification to future work.

6.3 Results

Results can be seen in Table 4.

PG_{Dyn} achieves the highest results of all systems. Using the silver operation labels (PG_{Oracle}) leads to a substantial further increase in performance across every metric, highlighting the impact of planning and pointing to the possibility of further improvements to be made.

Using either PG_{Dyn} or PG_{Clf} yields generally better results than Sent-BART. Both systems achieve better FKGL and SARI, suggesting greater output simplicity. Sent-BART achieves much higher source-oriented BARTScore (faithfulness) than even the references, suggesting some conservativity in its transformations.

PG_{Clf} achieves slightly higher recall BARTScore than PG_{Dyn}, while also generating the longest outputs, both in terms of tokens and sentences. This suggests it is less effective at identifying sentences for deletion, confirming our hypothesis that context is key for deletion. We can see here that the rank order of SMART matches that of BARTScore, suggesting it is similarly suited for simplification.

Both PG_{Tag} and $PG_{Tag+Dec}$ perform quite badly relative to the other PG systems and Sent-BART. However, Doc-BART is by far the worst performing system, presumably a result of it failing to properly handle the long document lengths.

7 Conclusion

In this paper we present an approach to document simplification that decomposes the task into a two-stage process of planning and generation. We propose a planning system that is able to take document context and structure into account to produce a coherent high-level simplification plan. By using this plan to guide a sentence-level simplification model, we are able to outperform end-to-end systems in terms of both meaning preservation and simplicity.

We leave for future work the development of dedicated simplification models that can leverage a document-level plan while also considering contextual information directly during generation.

8 Acknowledgements

We thank the anonymous reviewers for their feedback. We gratefully acknowledge the support of the French National Research Agency (Gardent; award ANR-20-CHIA-0003, XNLG "Multilingual, Multi-Source Text Generation").

Experiments presented in this paper were carried out using the Grid'5000 testbed, supported by a scientific interest group hosted by Inria and including CNRS, RENATER and several Universities as well as other organizations (see https://www.grid5000.fr).

9 Limitations

All models we propose are only trained with English datasets and therefore do not extent to other languages. There are also additional high-level simplification operations for which our planning framework does not offer support, such as sentence reordering, insertion, and fusion.

Furthermore, the Newsela dataset which we use in our experiments requires a license to use, meaning that researchers cannot fully reproduce our work without first obtaining said license from Newsela Inc. Due to this constraint and the low-quality alignments observed within the Wiki-auto dataset, we strongly encourage any work towards producing new open-access datasets for the document-level simplification task.

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, Carolina Scarton, and Lucia Specia. 2019a. EASSE: Easier automatic sentence simplification evaluation. 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): System Demonstrations*, pages 49–54, Hong Kong, China. Association for Computational Linguistics.

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

Reinald Kim Amplayo, Peter J. Liu, Yao Zhao, and Shashi Narayan. 2022. Smart: Sentences as basic units for text evaluation.

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*.

Liam Cripwell, Joël Legrand, and Claire Gardent. 2021. Discourse-based sentence splitting. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 261–273, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Liam Cripwell, Joël Legrand, and Claire Gardent. 2022. Controllable sentence simplification via operation classification. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 2091–2103, Seattle, United States. Association for Computational Linguistics.

Mohammad Dehghan, Dhruv Kumar, and Lukasz Golab. 2022. GRS: Combining generation and revision in unsupervised sentence simplification. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 949–960, Dublin, Ireland. 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 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4972–4984, Online. Association for Computational Linguistics.

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 57th Annual Meeting of the Association for Computational Linguistics*, pages 3393–3402, Florence, Italy. Association for Computational Linguistics.

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

Sian Gooding. 2022. On the ethical considerations of text simplification. In *Ninth Workshop on Speech*

and Language Processing for Assistive Technologies (SLPAT-2022), pages 50–57, Dublin, Ireland. Association for Computational Linguistics.

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.

David Kauchak. 2013. Improving text simplification language modeling using unsimplified text data. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1537–1546, Sofia, Bulgaria. Association for Computational Linguistics.

Dhruv Kumar, Lili Mou, Lukasz Golab, and Olga Vechtomova. 2020. Iterative edit-based unsupervised sentence simplification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7918–7928, Online. Association for Computational Linguistics.

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, Online. Association for Computational Linguistics.

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

Junyi Jessy Li and Ani Nenkova. 2015. Detecting content-heavy sentences: A cross-language case study. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1271–1281, Lisbon, Portugal. Association for Computational Linguistics.

Zhe Lin, Yitao Cai, and Xiaojun Wan. 2021. Towards document-level paraphrase generation with sentence rewriting and reordering. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1033–1044, Punta Cana, Dominican Republic. Association for Computational Linguistics.

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

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.

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, Hong Kong, China. Association for Computational Linguistics.

Louis Martin, Éric 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, Marseille, France. European Language Resources Association.

Kshitij Mishra, Ankush Soni, Rahul Sharma, and Dipti Sharma. 2014. Exploring the effects of sentence simplification on Hindi to English machine translation system. In *Proceedings of the Workshop on Automatic Text Simplification - Methods and Applications in the Multilingual Society (ATS-MA 2014)*, pages 21–29, Dublin, Ireland. Association for Computational Linguistics and Dublin City University.

Makoto Miwa, Rune Sætre, Yusuke Miyao, and Jun'ichi Tsujii. 2010. Entity-focused sentence simplification for relation extraction. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 788–796, Beijing, China. Coling 2010 Organizing Committee.

Christina Niklaus, Bernhard Bermeitinger, Siegfried Handschuh, and André Freitas. 2016. A sentence simplification system for improving relation extraction. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: System Demonstrations*, pages 170–174, Osaka, Japan. The COLING 2016 Organizing Committee.

Sergiu Nisioi, Sanja Štajner, Simone Paolo Ponzetto, and Liviu P. Dinu. 2017. Exploring neural text simplification models. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 85–91, Vancouver, Canada. Association for Computational Linguistics.

Kostiantyn Omelianchuk, Vipul Raheja, and Oleksandr Skurzhanskyi. 2021. Text Simplification by Tagging. In *Proceedings of the 16th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 11–25, Online. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. 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 3982–3992, Hong Kong, China. Association for Computational Linguistics.

C. Scarton, P. Madhyastha, and L. Specia. 2020. Deciding when, how and for whom to simplify. © 2020 The Author(s) and IOS Press. This is an Open Access article distributed under the terms of the Creative Commons Attribution-NonCommercial Licence (http://creativecommons.org/licenses/by-nc/4.0/).

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, Melbourne, Australia. Association for Computational Linguistics.

Thomas Scialom, Louis Martin, Jacopo Staiano, Éric Villemonte de la Clergerie, and Benoît Sagot. 2021. Rethinking automatic evaluation in sentence simplification.

Advaith Siddharthan. 2003. Preserving discourse structure when simplifying text. In *Proceedings of the 9th European Workshop on Natural Language Generation (ENLG-2003) at EACL 2003*, Budapest, Hungary. Association for Computational Linguistics.

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, Online. Association for Computational Linguistics.

Sanja Štajner and Maja Popovic. 2016. Can text simplification help machine translation? In *Proceedings of the 19th Annual Conference of the European Association for Machine Translation*, pages 230–242.

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

Renliang Sun, Zhe Lin, and Xiaojun Wan. 2020. On the helpfulness of document context to sentence simplification. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1411–1423, Barcelona, Spain (Online). International Committee on Computational Linguistics.

K. Woodsend and Mirella Lapata. 2011a. Wikisimple: Automatic simplification of wikipedia articles. In *AAAI*.

Kristian Woodsend and Mirella Lapata. 2011b. 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.

Weizhe Yuan, Graham Neubig, and Pengfei Liu. 2021. Bartscore: Evaluating generated text as text generation. In *Advances in Neural Information Processing Systems*, volume 34, pages 27263–27277. Curran Associates, Inc.

Bohan Zhang, Prafulla Kumar Choubey, and Ruihong Huang. 2022. Predicting sentence deletions for text simplification using a functional discourse structure. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 255–261, Dublin, Ireland. Association for Computational Linguistics.

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.

Yang Zhong, Chao Jiang, Wei Xu, and Junyi Jessy Li. 2020. Discourse level factors for sentence deletion in text simplification. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):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, Beijing, China. Coling 2010 Organizing Committee.

A Training Details for the Planning Models

Each model was trained with a learning rate of $1e^{-5}$, a batch size of 32 and a dropout rate of 0.1. We ran experiments on a computing grid with 2× Nvidia A40 GPUs (45GB memory).

For the contextual classifier, we test with r values of 9 and 13, subject to findings in Appendix C. All layers in common with the standard RoBERTa architecture are initialised with the `RoBERTa-base` pretrained weights. All added positional embedding layers are also initialised with the pretrained weights from the `RoBERTa-base` positional embedding layer. All other layers are randomly initialised.

B Training Details for the Simplification Models

For all generative models, we used a learning rate of $3e^{-5}$, a batch size of 16, and performed dropout with a rate of 0.1 and early stopping. The network has 6 layers in each of the encoder and decoder, with a hidden size of 768. All models were trained on a computing grid using 2× Nvidia A40 GPUs (45GB memory) in under 24 hours.

C Context Window Size

To determine the optimal context window size for the contextual planner we ran a series of experiments with varying values of the radius, r. We used 100,000 random examples from the Newsela-auto (non-adjacent reading levels) training set and trained a model with each of the configurations for 5 epochs. Results can bee seen in Figure 4.

The *deletion* operation is most affected by the inclusion of context, with performance rapidly rising as r grows to 13. The *rephrase* operation appears to slowly degrade in performance as r increases, while the other two operations show no obvious pattern. We also observe that $r = 9$ produces the highest macro F1.

D Extra Evaluation Results

For clarity, we provide scores for a wider range of simplification evaluation metrics that were not included in the main body of the paper in Table 5. These mostly include popular metrics used for sentence simplification that we do not believe adapt as well to the document-level setting, do not provide further insight into system differences, or have not recieved much support in the literature. Specifially, we include BLEU (Papineni et al., 2002), and full operation scores for both SARI and D-SARI (Sun et al., 2021). For D-SARI, we apply the document-level penalties on top of the base EASSE implementation of SARI.

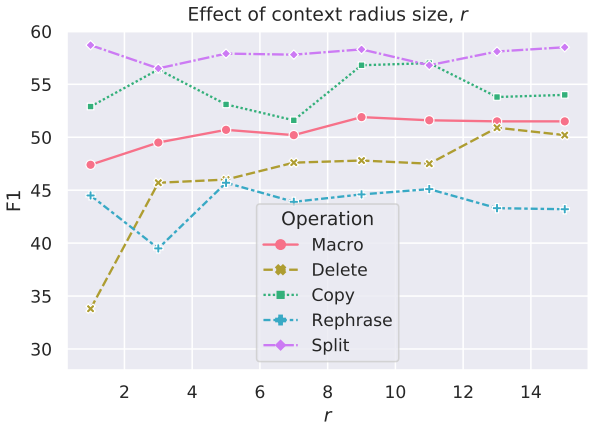

Figure 4: Effect of context window size on F1 scores.

We can see that the main SARI differences between the context-free planner and Sent-BART is that Sent-BART achieves higher keep, while the planner achieves higher add. This suggests that Sent-BART is likely more conservative in edits. Further, as the planner does not have access to contextual content, it is likely failing to consistently copy/delete the correct parts of the text.

E Example Planner Predictions

Figure 5 shows example snippets of planner model outputs. We have selected representative extracts that highlight the strengths and weaknesses of the main models. We do not include outputs from Tagger as they are virtually identical to Tagger+Dec in most cases and therefore do not provide further insight.

F Example Simplification

Figure 6 shows system output examples for the simplification models. We only show texts from Wiki-auto as they are easier to showcase due to their shorter length, as well as their being licensing restrictions for Newsela content.

System	BLEU ↑	D-SARI ↑	add	keep	delete	SARI ↑	add	keep	delete
Input	46.2	8.76	0.0	26.29	0.0	20.52	0.0	61.56	0.0
Reference	100.0	99.98	99.99	99.97	99.99	99.99	100	99.97	99.99
Doc-BART	31.13	30.60	16.54	25.01	50.24	47.07	20.41	55.40	65.40
Sent-BART	70.74	66.27	53.89	71.95	72.95	73.02	55.91	83.66	79.48
PG_{Tag}	48.08	42.96	31.70	44.01	53.17	56.13	35.61	65.61	67.18
$PG_{Tag+Dec}$	48.12	43.31	31.57	44.68	53.69	56.06	35.54	65.54	67.11
PG_{Clf}	70.84	62.97	56.31	65.15	67.47	73.83	57.62	83.56	80.32
PG_{Dyn}	**72.41**	**67.42**	**56.83**	**71.82**	**73.61**	**75.00**	**58.88**	**84.75**	**81.36**
PG_{Oracle}	**78.97**	**77.02**	**63.44**	**83.92**	**83.70**	**80.74**	**65.22**	**89.94**	**87.05**

Table 5: Extra results for document simplification experiments on Newsela-auto.

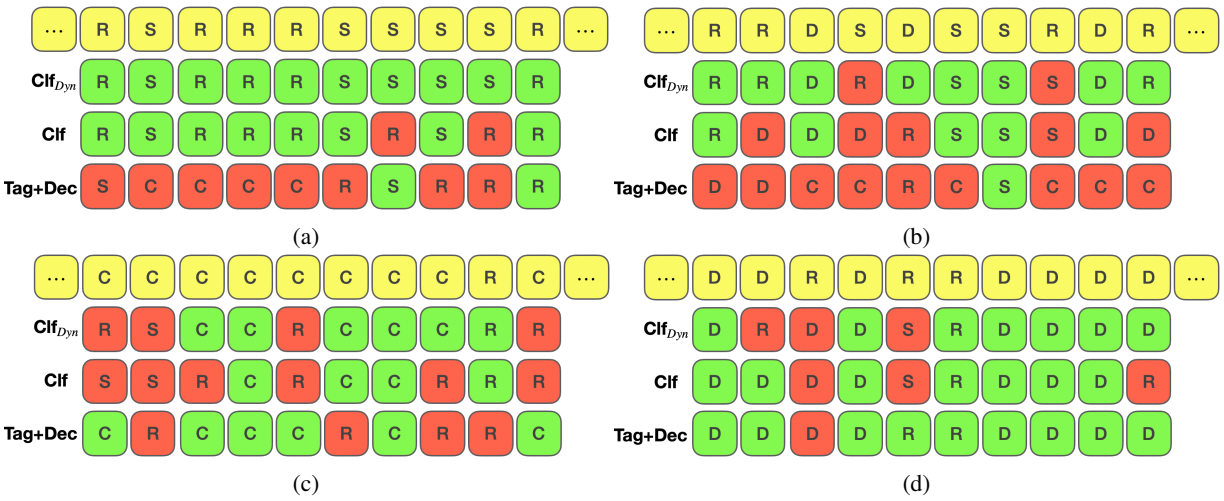

Figure 5: Example planning results for various models. Subfigures show representative snippets from Newsela-auto test-set documents. The silver labels are shown above in yellow, and system outputs are shown on the rows below with correct predictions in green and incorrect predictions in red. Clf_{Dyn} is our best performing model, the contextual classifier with dynamic context. Figure 5a shows a case where there are lots of context-agnostic operations (rephrase, split) resulting in poor performance from Tagger+Dec. Figure 5b shows a varied snippet where Clf_{Dyn} appears to be the best at identifying both rephrase and split, as well as delete. Figures 5c and 5d show that Tagger+Dec is capable of performing well in situations demanding a lot of context-dependent operations (copy, delete).

System	Output
Complex	Silvano "Nano" Campeggi (January 23, 1923 – August 29, 2018) was an Italian artist who designed and produced the artwork for the posters of many classic Hollywood films. His iconic images are associated with the golden era of Hollywood and Campeggi is now generally regarded as the most important graphic artist and poster designer in the history of American cinema. In the following decades, Campeggi designed and produced the poster and advertising graphics for over 3000 films, working not only under contract with the MGM studios, but also with Warner Brothers, Paramount, Universal, Columbia Pictures, United Artists, RKO, Twentieth-Century Fox and several other movie studios. Sixty-four of the films he illustrated won Oscars, including "Casablanca", "Ben-Hur", "Singin' in the Rain", "An American in Paris", "West Side Story", "Exodus", "Breakfast at Tiffany's", and "Gigi". Campeggi died on 29 August 2018, at the age of 95.
Simple	Silvano "Nano" Campeggi (January 23, 1923 – August 29, 2018) was an Italian artist. He designed and produced the artwork for the posters of many classic Hollywood movies. Campeggi was known for his poster designs for "Casablanca", "Singin' in the Rain", and "Breakfast at Tiffany's". Campeggi died on August 29, 2018 in Florence from respiratory failure at the age of 95.
Doc-BART	Silvano "Nano" Campeggi (January 23, 1923 – August 29, 2018) was an Italian artist. He designed and produced the artwork for the posters of many classic Hollywood movies, including "Casablanca", "Ben-Hur", "Singin' in the Rain", "An American in Paris", "West Side Story", "Exodus", "Breakfast at Tiffany's", "Gigi", and "".
Sent-BART	Silvano "Nano" Campeggi (23 January 1923 – 29 August 2018) was an Italian artist. He designed and produced the artwork for the posters of many classic Hollywood movies. Campeggi is generally regarded as the most important graphic artist and poster designer in the history of American cinema. Campeggi designed and produced the poster and advertising graphics for over 3000 movies, working not only under contract with the MGM studios, but also with Warner Brothers, Paramount, Universal, Columbia Pictures, United Artists, RKO, Twentieth Century Fox and several other movie studios. Sixty-four of the movies he illustrated won Oscars, including "Casablanca", "Ben-Hur", "Singin' in the Rain", "An American in Paris", "West Side Story", "Exodus", "Breakfast at Tiffany's", and "Gigi". Campeggi died on 29 August 2018, at the age of 95.
PG_{Dyn}	Silvano "Nano" Campeggi (January 23, 1923 – August 29, 2018) was an Italian artist. He designed and produced the artwork for the posters of many classic Hollywood movies. Sixty-four of the movies he illustrated won Oscars, including "Casablanca", "Ben-Hur", "Singin' in the Rain", "An American in Paris", "West Side Story", "Exodus", "Breakfast at Tiffany's", and "Gigi". Campeggi died on 29 August 2018 at the age of 95.

Table 6: Simplification outputs for a specific document pair example. Although Newsela-auto is the focus of our simplification experiments, we can only include example documents from Wiki-auto due to licensing constraints.