A Tree is All You Need: Unsupervised Sentence Simplification via Dependency Parsing

Vy Vo, Weiqing Wang, and Wray Buntine

Monash University, Clayton, Victoria 3800, Australia
tvoo0019@student.monash.edu

Abstract. Text simplification is a challenging task because it involves more complicated transformations than other text generation tasks such as summarization or paraphrasing alone. One significant obstacle is the lack of large-scale high quality parallel corpora that hinders supervised approaches. But this also motivates unsupervised methods for accelerating research progress. In this paper, we propose a simple yet novel unsupervised sentence simplification system that harnesses parsing structures to produce linguistically effective simplifications. We establish the unsupervised state-of-the-art at 39.13 SARI on TurkCorpus set and perform competitively against supervised baselines on various quality metrics. Our model is capable of introducing substantial modifications to simplify a sentence while maintaining its original semantics and adequate fluency. Furthermore, we demonstrate our framework's extensibility to other languages via a proof-of-concept on Vietnamese data. Codes are available in the Appendices.

Keywords: unsupervised text simplification, dependency parsing, semantics similarity, back translation, Vietnamese simplification

1 Introduction

Text simplification is the task of generating a simpler version of a text so that it can be easily understood, while preserving the main ideas. Simplification is more complicated because it encompasses more transformations than other text rewriting tasks. It involves lexical and structural modifications (as in paraphrasing), length reduction (as in summarization), and can also be regarded as a style transfer problem of transforming a text of expertise into layman language. An overwhelming number of simplification systems rely on parallel corpora to learn complex transformation patterns and automatically generate simplified texts [4,7,10,16,18,21,31,35]. However, the scarcity of high-quality and large-scale datasets is the biggest impediment to progress in supervised text simplification. In recent years, semi-supervised and unsupervised approaches have shown promising results [8,11,14,13,17,20,29,37]. Techniques range from data-driven neural models to linguistically rule-based frameworks, attempting

to achieve hybrid simplifications, or simply targeting a few key operations, or focusing on controllability. Previous studies criticizes these simplification systems for being opaque, suboptimal, and compromising meaning preservation [5,13]. In other words, there is a trade-off of Simplicity vs. Fluency and Adequacy, which to the best of our knowledge, no works have tackled. Thus, we conduct the first study on effective simplification that aims to achieve a balance of these three dimensions. Details of the differences between our approach and existing works can be found in Related Work.

Contribution. Our contribution is summarized as follows:

- We propose a novel method `USDP` for Unsupervised sentence Simplification via deep Dependency Parsing to effectively simplify sentences while adequately preserving semantics and fluency. The model outperforms unsupervised counterparts and stands competitively against supervised systems on SARI and quality metrics.
- We demonstrate that the framework readily extends to other languages: we adapt it for generating simplified Vietnamese sentences as proof-of-concept.
- We provide linguistically-motivated empirical evidence confirming the intuition behind our framework and explaining how simplification arises.

2 Related Work

Earlier works inherit techniques from statistical machine translation [3] to translate a text of the *complex* language to the *simple* language. The translation model is learned through aligned words or phrases in normal-simplified text pairs, referred to as phrase-based simplification [4,10,18,31]. Syntax-based simplification, another line of work, has alignment using syntactic components [38]. The first neural Seq2Seq text simplification system is proposed in [21], based on which Zhang & Lapata [35] and Gou et al. [7] later adopt reinforcement learning algorithm in a similar architecture. Audience Centric Sentence Simplification (`ACCESS`) is a recent supervised state-of-the-art approach [16] that conditions simplifications on various attributes of text complexity. Despite impressive results, the shortage of high-quality datasets pose serious limitations to supervised systems, motivating research in semi-supervised and unsupervised solutions.

Surya et al. [29] propose the first unsupervised neural model for text simplification that minimizes adversarial losses on two separate sets of complex and simple sentences extracted from a parallel Wikipedia corpus. Instead of using aligned data, Zhao et al. [37] introduce a noising mechanism to generate parallel examples from any English datasets, then train denoising autoencoders to reconstruct the original sentences. In the same spirit, Martin et al. [17] use multilingual translation systems to produce various simpler paraphrases from monolingual corpora (e.g., English to French, then French to English), thus eliminating the need of labeled data. This framework is referred to as *back translation*. `DisSim` [20] is another effort focusing on splitting and deletion by applying 35 handcrafted grammar rules over a constituency parse tree.

On the other hand, iterative and decoding-based approaches as in [8,11,27] are considered more effective since not only can they generate hybrid outputs but also allow for quality control explicitly via a scoring function balancing simplicity, fluency and semantics preservation. The algorithm of Kumar et al. [11] iteratively edits a given complex sentence to make it simpler using four operations: removal, extraction, reordering and substitution, while Kariuk & Karamshuk [8] implement beam search with simplicity-aware penalties for sentence simplification without supervision. Schumann et al. [27] design a hill-climbing searching strategy for extractive summarization. They propose a more efficient search space than regular generative models by sampling words in the source text, and the model is explicitly encouraged to generate candidates that maximize semantic similarity while satisfy length constraints.

This motivates our attempt to improve the existing decoding procedure through a linguistics-based unsupervised framework for sentence simplification. We tackle structural and lexical simplification sequentially, rather than simultaneously like previous works, since it would support interpretation and allow more controllability. This sequential approach is also adopted in [13], which leverages `DisSim` together with a self-designed paraphrasing system. We first develop a **stand-alone decoding framework**, similar to [27] for structural simplification, then implement **back translation** for lexical simplification and paraphrasing. We also employ a left-to-right beam search strategy, but discover a much more efficient search space via dependency parsing compared to word-level extraction or constituency parsing. Unlike [27] in which the authors attribute the success of their model to sampling strategy, we argue that similarity control plays a more critical role in our approach. Another key difference of our approach is that we make full use of pre-training, i.e., neither fine tuning nor end-to-end training is required, thereby making it highly reproducible and robust to OOD examples.

3 Method

We propose a two-phase pipeline that tackles syntactic and lexical simplification one by one. The first phase consists of an independent left-to-right decoder, integrated with a novel and efficient sampling strategy that makes use of dependency relations among words in a sentence. This phase mainly focuses on deletions, though we also induce chunking (i.e., breaking a sentence into meaningful phrases) during the process. In the second phase, we back translate the generated English outputs from phase 1 to generate effective paraphrases and lexical simplifications.

3.1 Structural Simplification

Search Objective. Given an input sentence $c = (c_1, c_2, ..., c_n)$, we aim to generate a shorter sentence $s = (s_1, s_2, ..., s_m)$ expressing the same meaning as c. Whereas the previous works perform deletions by imposing length constraints, we go beyond length reduction and strictly define which parts of a sentence to

keep and which to remove. The goal is to eliminate redundant details – those if removed do not significantly alter the meaning of the entire sentence. We quantify the importance of a token by measuring changes in semantic similarity scores when omitting it. This motivates our search objective function as follows

$$f(s) = f_{sim}(c, s) + \alpha f_{flu}(s) + f_{depth}(s) \tag{1}$$

where α is the relative weight on Fluency score. Note that, the detailed reason why the weights of f_{sim} and f_{depth} are the same is given below under **Search Algorithm**. The decoding objective is a linear combination of individual scoring functions with each score normalized within the range $[0, 1]$. Details of each score function are described below.

Semantic Similarity: We calculate cosine similarity between sentence embeddings of c and generated hypothesis $s_{1:t}$ at each time step t, so $\mathrm{f}_{sim}(c, s_{1:t}) = cos(e_c, e_{s_{1:t}})$. In our work, we use the pre-trained sentence-BERT model (SBERT) [23] to derive semantically meaningful sentence embeddings to calculate cosine similarity. Other unsupervised works, in constrast, use IDF weighted average of unigram embeddings [11,27].

Fluency: Our fluency scorer quantifies the grammatical accuracy of a sequence based on a constituent-based 4-gram language model. The fluency score is

$$f_{flu}(s_{1:t}) = \frac{1}{|s_{1:t}|} \sum_{u=1}^{t} \log p(pos_u \,|\, pos_{1:u-1})$$

where pos_t indicates the part-of-speech of token s_t The language model is pretrained on a massive unlabeled corpus. Because English constituents are bounded, constituent-based language model is a reusable light-weight solution compared to regular vocabulary-based language models.

Tree Depth Constraint: Dependency tree depth is a popular metric of syntactic complexity in various literature in linguistics [6,26,34]. Deeper trees indicate more complicated structures, and it is recently shown in `ACCESS` [16] that controlling maximum depth of dependency tree yields the best simplification results. Thus, f_{depth} further scores candidate sentences by the **inverse maximum tree depth** reached at the generated token. This constraint prevents the decoder from going too deep, thereby producing a structurally simpler output.

Search Space. Deletion is a form of extractive summarization by nature, motivating us to adopt the word-extraction method proposed by [27]. They suggest candidates be selected from tokens in the input sentence, instead of the corpus vocabulary. Specifically at each step, a new candidate is sampled from words that are in the input sentence but not in the current summary. We argue that this is not necessary and propose a more efficient approach. Figure 1 illustrates a hierarchical view of a dependency tree. We observe that each token exists in

dependency relations with its parent and children. In other words, its meaning and function are directly determined by these nodes in the dependency tree. At a time step, we only consider tokens that are the direct children of the previously generated token, resulting in a smaller search space. We additionally arrange the words in the same order as the input before scoring hypotheses since word order plays a critical role both grammatically and semantically. This approach not only guarantees global fluency but also allows us to achieve optimal solution with a small beam size. We refer to this novel strategy as **Family Sampling**.

Search Algorithm. We integrate our novel family sampling strategy with a regular beam search algorithm that keeps top k hypotheses with highest scores derived from Equation 1. To begin with, we condition the sequence on the main subject of the sentence, i.e. the subject of the `ROOT` verb. This does not affect the rest of the sentence, but contributes to simplification by directly introducing the main verb and subject. For each search step, a candidate token s_t is sampled from the family of direct child nodes of token s_{t-1}, excluding those having been previously generated. We score each candidate according to (1) and select k hypotheses with the highest scores for the next generation step. Our search only terminates when it completes a branch of a predefined length λ and satisfies the minimum similarity threshold τ. Note in (1) that we treat f_{sim} and f_{depth} equally so that we only need to control trade-off between compression and semantics via λ and τ. The goal is to find the shortest most similar sub-sequence, and we wish to preserve as much semantics as possible by keeping tokens that add the most semantic value to the sequence. As mentioned in [27], given input length n, target output length m and corpus vocabulary size V, auto-regressive or edit-based generation has search space of V^m, while ours is restricted to C_n^m. Regarding time complexity, our algorithm is bounded by $O(d \times k \times max\_ch(s_{t_1}))$ with parsing tree depth d, beam size k and $max\_ch(s_{t_1}))$ being the maximum number of direct children of a token s_{t-1} where $ch(s_{t_1}) \leq n$ and mostly $\ll n$.

Chunking: A branch in the tree is found to correspond to a meaning chunk in the sentence, so we add a `<SEP>` token inducing a chunk whenever a sampling set is empty. Humans naturally perform simplification in this manner by chunking a complex sentence into understandably simpler structures, and we find that most of these chunks are prepositional and adjective phrases. After a `<SEP>`, we reverse the tree and restart family sampling with the token nearest to `ROOT`.

3.2 Back Translation

Phase 2 aims to enhance the quality of our simplifications through paraphrasing and lexical simplification using a back translation framework, which consists of two reliable off-the-shelf machine translation systems English-X and X-English where X is any other language. We first translate the structurally simpler sentences in English to X, then have the outputs back translated to English. This technique is applied in style transfer tasks [22,36] to disentangle the content

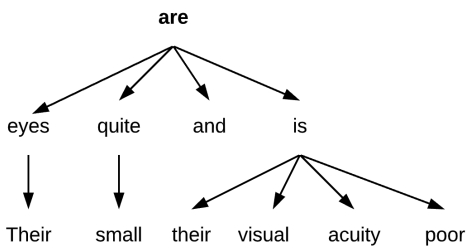

Fig. 1. A dependency tree of the sentence *Their eyes are quite small, and their visual acuity is poor.* The `ROOT` verb is *are* and the main `subj` is *Their eyes.*

and stylistic characteristics of the text. Thus, we rely on back translation to strip the complex style off a text while keep meaning unchanged. Not only does it contribute to additional simplifications via paraphrasing and lexical substitution, but we also find it particularly useful to correct subpotimality in the decoder's output. Another advantage of this technique is that one can further collect various paraphrases by exploiting multiple languages.

4 Experiments

4.1 Data

We evaluate our English model on TurkCorpus [33] and PWKP [35]. TurkCorpus is a standard dataset for evaluating sentence simplification works, which contains 2000 sentences for validation and 359 for testing, each of which has 8 simplification references collected through crowd-sourcing. TurkCorpus is originally extracted from *WikiLarge* corpus compiled from [35]. PWKP is the test set of *WikiSmall* - another dataset constructed from main-simple Wikipedia articles. PWKP provides 100 test sentences with 1-to-1 aligned reference. *Newsela* [32] is another commonly used dataset in text simplification works, which is unfortunately unavailable due to restricted access rights. We only use the test sets for evaluation and comparison, and to prove the robustness of our system, we further use *CNN / Daily Mail* dataset [28] for training the fluency model, leaving both evaluation corpora untouched. For the Vietnamese model, we train the fluency model on the public Vietnamese news corpus *CP_Vietnamese-UNC* of 41947 sentences), then generate simplifications of 200 sentences extracted from Vietnamese law corpus *CP_Vietnamese-VLC* in an unsupervised manner. Both datasets are open sourced by Undertheseа NLP [1].

4.2 System Details

For simplicity we utilize SpaCy[2] integrated with Berkeley Neural Parser [9] for constituent and dependency parsing. Since SpaCy does not currently support

[1] `https://github.com/undertheseanlp/resources`
[2] `https://spacy.io`

Vietnamese, we parse Vietnamese texts through VnCoreNLP[3] [30]. VnCoreNLP is built upon Vietnamese Treebank [19], which contains 10200 constituent trees formatted similarly to Penn Treebank [15]. We obtain sentence embeddings from SBERT model `paraphrase-mpnet-base-v2` [24] for English, and the multilingual version `distiluse-base-multilingual-cased-v2` for Vietnamese [23]. Our fluency model is a 4-gram language model with Kneser-Ney smoothing built on sequences of constituents and implemented via NLTK[4] package. Regarding the back translation framework, we experiment with free Google Translate service[5] - a robust neural translation system that support two-way translation **English-German** and **German-English**. Though the system can be run with any available target languages, we choose German for our main model because it is a high-resource language. For the Vietnamese model, we simply use English as the target language. While our search algorithm can automatically guarantee language idiomaticity, grammatical accuracy is an issue since the model tends to prefer content words to function words to maximize semantic similarity. Thus, we strictly force Fluency to be twice as important, i.e., $\alpha = 2$, while set equal weights to Similarity and Depth constraint. Our base model is evaluated at $\lambda = 0.5$ and $\tau = 0.95$, reported as `USDP-Base`. Additionally, in order to validate our effectiveness more comparably, we approximate τ to same level of competing unsupervised methods, respectively at 0.90 for TurkCorpus (`USDP-Match`^{a}) and 0.75 for PWKP (`USDP-Match`^{b}). Across all experiments, beam size is fixed at 5, and in order to understand the effect of back translation, we also evaluate the quality of simplifications before and after phase 2.

4.3 Competing Models

We benchmark our system against both supervised and unsupervised models. Supervised systems include `PBMT-R` [31], `SBMT-SARI` [33], `Dress` / `Dress-Ls` [35] and recent state-of-the-art `ACCESS` [16]. We also consider semi-supervised `BTTS / BTRLTS / BTTS100` [37] and unsupervised counterparts `UNTS / UNTS10K` [29] and `RM+EX / RM+EX+LS / RM+EX+RO / RM+EX+LS+RO` [11].

5 Results

5.1 Automatic Evaluation

We use EASSE package [2] to compute standardized simplification metrics and perform evaluation on publicly accessible outputs of competing systems. These include Compression ratio (**CR**), Exact copies (**CP**), Split ratio (**%SP**), Additions proportion (**%A**) and Deletions proportion (**%D**), all of which are evaluated against the source sentences. We exclude BLUE and FKGL since BLEU is previously reported to be a poor estimate of simplicity [1,31,33] and FKGL only

[3] `https://github.com/vncorenlp/VnCoreNLP`
[4] `https://www.nltk.org/`
[5] `https://translate.google.com/`

applies to text of at least 200 words. We contribute to enhancing the current simplification evaluation suite by adding measures of semantic similarity and fluency. Similarity score is again based on cosine similarity between sentence embedding vectors (**SIM**), and we adopt a "referee" language model for scoring fluency (**FL**). This is to assure fair comparison among systems since ours has a different fluency scoring scheme. We use **pseudo-log-likelihood scores** (PPLs) proposed in [25], which is shown to promote linguistic fluency rather than pure likeliness in conventional log probabilities. We also evaluate the reference sentences on these quality metrics, and benchmark the outputs against them through SARI (average **SARI** and component **Add**, **Keep**, **Del** scores). This however is only on TurkCorpus set since PWKP only provides 1 reference. Table 1, 2 and 3 present results of automatic evaluation respectively on TurkCorpus, PWKP and CP_Vietnamese-VLC, both before and after Back translation (**BT**) implemented.

TurkCorpus. We establish the unsupervised state-of-the-art SARI on TurkCorpus, +1.65 point improvement over the closest baseline and only behind two supervised methods: `ACCESS` and `SBMT-SARI`. In addition to the competitive performance on Compression ratio and Split ratio, we outperform the current semi-supervised and unsupervised across all other quality metrics. Our simplifications have the highest fluency at **-2.47** and similarity score at **0.95** while achieve remarkably high percentages of additions at **16%** and deletions at **21-25%** at the same level of some supervised methods. Our raw outputs from phase 1 alone gains fairly high proportions of deletions as lowering τ. Note that this number generally takes both deletions and substitutions into account, but in this phase, it reflects our model's effectiveness in performing deletions since substitutions are not implemented until phase 2. Our main focus in structural simplification is not on sentence splitting, rather chunking it into meaningful sub-sequences. Thus, our system fails to perform standard splitting operation, but in realistic settings, splitting is an incredibly challenging task when done without parallel examples or extensive grammar rules.

PWKP. As Kumar et al. [11] do not experiment on PWKP, we run their codes to evaluate `RM+EX+LS+RO` model on PWKP set for comparison. Overall, our model and `RM+EX+LS+RO` produce more diverse simplifications than the supervised systems, measured by remarkable proportion of additions and deletions. Interestingly, the quality of unsupervised outputs is also closer to that of references, in which `RM+EX+LS+RO` achieves consistently better performance. This may be because the model is accompanied by a pretrained Word2Vec on WikiLarge data, which has a relatively same distribution as PWKP as both are Wikipedia-based. Meanwhile, none of our variants see any similar examples of any kind beforehand. Given such a high level of modification, we again have the highest similarity score at **0.96**, and when we try to match the similarity level as in `USDP-Match`^{b}, we achieve more compression (**54%**) and deletions (**56%**) while

Table 1. Results on TurkCorpus. ↑ Higher is better. ↓ Lower is better.

TurkCorpus	CR↓	CP↓	%SP↑	%A↑	%D↑	FL↑	SIM↑	SARI↑	Add↑	Keep↑	Del↑
Reference	0.95	1.07	0.16	0.14	0.18	-2.63	0.95	-	-	-	-
Supervised											
Dress	0.75	0.22	0.99	0.04	**0.27**	-2.66	0.91	36.84	2.5	65.65	42.36
Dress-Ls	0.77	0.26	0.99	0.04	**0.26**	-2.63	0.92	36.97	2.35	67.23	41.33
PBMT-R	0.95	0.11	1.03	0.10	0.11	-2.59	**0.96**	**38.04**	5.04	**73.77**	35.32
ACCESS	0.94	**0.04**	1.20	**0.16**	0.16	-2.52	**0.95**	**41.38**	6.58	72.79	44.78
SBMT-SARI	0.94	0.10	1.02	**0.16**	0.13	-2.65	**0.96**	**39.56**	5.46	72.44	40.76
Semi-Supervised											
BTTS100	0.92	0.45	1.02	0.03	0.10	**-2.46**	**0.97**	34.48	1.51	**74.44**	27.48
BTTS	0.92	0.20	1.17	0.08	0.14	-2.66	**0.96**	36.38	1.9	71.03	36.22
BTRLTS	0.92	0.19	1.16	0.08	0.15	-2.70	**0.96**	36.49	2.14	70.31	37.03
Unsupervised											
UNTS	0.85	0.21	1.00	0.06	0.17	-2.70	0.89	36.29	0.83	69.44	38.61
UNTS_10K	0.88	0.19	1.01	0.07	0.14	-3.10	0.92	37.15	1.12	71.34	38.99
RM+EX	0.83	0.44	1.00	0.01	0.15	-2.58	0.94	35.88	0.84	73.14	33.65
RM+EX+LS	0.82	0.16	1.00	0.06	0.21	-2.91	0.90	37.48	1.59	68.20	42.65
RM+EX+RO	0.86	0.36	1.01	0.02	0.14	-2.61	0.94	36.07	0.99	72.36	34.86
RM+EX+LS+RO	0.85	0.13	1.01	0.08	0.20	-2.92	0.90	37.27	1.68	67.00	43.12
Our system											
USDP-Base											
With BT	0.92	**0.04**	1.01	**0.16**	**0.21**	**-2.47**	**0.95**	**39.13**	**6.77**	64.44	**46.19**
Without BT	0.95	0.15	1.00	0.07	0.09	-2.877	**0.98**	34.13	0.87	71.34	30.18
USDP-Match[a]											
With BT	0.88	**0.04**	1.01	**0.15**	**0.25**	-2.55	0.94	**38.33**	**6.28**	62.13	**46.58**
Without BT	0.89	0.13	1.00	0.07	0.15	-3.053	**0.96**	34.44	0.94	68.57	33.82

preserving slightly higher semantics than RM+EX+LS+RO. The fluency scores of simplifications from all automated systems remain far behind human outputs.

CP_Vietnamese-VLC. As a proof-of-concept for our approach in another language, we only conduct evaluation on USDP-Base. We do not report PPLs since that model has only been shown to work on English data. Instead, we report normalized character-level Levenshtein similarity [12] (**LevSIM**) which demonstrates the structures of the output sentences do not deviate significantly from the original. Results in Table 3 are consistent with what we have achieved on English corpora, proving the potential to apply the framework to other languages.

5.2 Human Evaluation

Human judgement is critical to assess text generation. We randomly select 50 sentences from TurkCorpus test set, and have 5 volunteers (2 native and 3 non-native speakers with adequate English proficiency) examine the simplified outputs from ACCESS (supervised state-of-the-art), RM+EX+LS (closest and best performing unsupervised variant) and our method USDP-Base. In a similar setup to

Table 2. Results on PWKP dataset. ↑ Higher is better. ↓ Lower is better.

PWKP	CR↓	CP↓	%SP↑	%A↑	%D↑	FL↑	SIM↑
Reference	0.81	0.03	**1.31**	**0.17**	0.32	**-1.39**	0.91
Supervised							
Dress	0.62	0.11	1.01	0.02	0.39	-2.18	0.87
Dress-Ls	0.63	0.13	1.01	0.01	0.37	-2.10	0.88
PBMT-R	0.96	0.14	1.01	0.06	0.07	-2.05	**0.97**
Unsupervised							
RM+EX+LS+RO	0.61	**0.01**	**1.21**	**0.17**	**0.52**	-2.68	0.81
Our system							
USDP-Base							
With BT	0.87	0.03	1.00	**0.16**	0.28	-2.05	**0.96**
Without BT	0.88	0.08	1.00	0.06	0.15	-2.64	**0.95**
USDP-Match[b]							
With BT	**0.54**	**0.00**	1.00	0.11	**0.56**	-2.38	0.84
Without BT	**0.53**	0.03	1.00	0.03	0.49	-3.36	0.85

Table 3. Results of USDP-Base on CP_Vietnamese-VLC

CP_Vietnamese-VLC	CR↓	CP↓	%SP↑	%A↑	%D↑	LevSIM↑	SIM↑
With BT	0.89	0.00	1.06	0.11	0.20	0.86	0.91
Without BT	0.91	0.00	0.99	0.06	0.12	0.91	0.94

the previous studies [11,13,37], the volunteers are asked to use a five-point Likert scale and provide ratings for each simplification version on 3 dimensions: **Fluency** *(Is the output sentence grammatical and well formed?)*, **Adequacy** *(How much meaning from the original sentence is preserved?)* and **Simplicity** *(Is the output simper than the original sentence?)*. We also have 50 Vietnamese simplifications from CP_Vietnamese-VLC outputs assessed by 4 native Vietnamese speakers on the same quality dimensions. We simply report the average ratings in Table 4, substantiating that we surpass both ACCESS and RM+EX+LS on all dimensions, and our simplified sentences in Vietnamese are perceived to have adequate quality.

Table 4. Human Evaluation Results on TurkCorpus (English) and CP_Vietnamese-VLC (Vietnamese) datasets.

Model	Fluency	Adequacy	Simplicity
English			
USDP-Base	**4.32**	**3.93**	**3.22**
ACCESS	4.16	3.46	3.18
RM+EX+LS	3.59	3.12	2.86
Vietnamese			
USDP-Base	3.33	3.48	3.04

5.3 Controllability

Table 5 displays output results on different values of τ and λ. Adjusting similarity threshold τ has more impact on the output quality than length ratio λ. This is simply because our algorithm must satisfy a pre-defined τ before termination, regardless of length constraints. This shows that lowering similarity threshold encourages the model to produce shorter sentences, by inducing more deletions and compression. However, this does not occur at the cost of semantics preservation. Even when τ is set to 0.70, the output sentences still have very high similarity scores. Little content is lost since we not only reduce length but also seek to maximize semantics preservation. This is done by only extracting important tokens, most of which turn out to be content words. This behavior is discussed in detail in the following section.

Table 5. Effects of threshold values on simplification quality of 100 sentences from TurkCorpus. Both are evaluated on `USDP-Base`.

Value	**CR↓**	**%D↑**	**SARI↑**	**SIM↑**
Effect of λ at $\tau = 0.95$				
0.25	0.949	0.084	33.599	0.976
0.50	0.949	0.084	33.599	0.976
0.75	0.960	0.071	33.472	0.977
1.00	0.991	0.037	32.864	0.981
Effect of τ at $\lambda = 0.5$				
0.70	0.810	0.221	33.626	0.919
0.80	0.841	0.194	33.766	0.938
0.90	0.913	0.122	33.800	0.967
0.95	0.949	0.084	33.599	0.976

6 Discussion

The main contribution of our paper lies in our intuition underlying family sampling that leads to meaningful deletions. At local search steps, we ensure each added token brings about significant improvement in semantics. We conduct a syntactic investigation to understand this behavior better. In figure **A** (Fig. 2), we examine the correlation between the tokens' part-of-speech and changes in similarity. We randomly sample 1 million English sentences from all the data, consecutively removing each token from its original sentence and tracking how much reduction in semantics similarity it causes. **Content words** such as `NOUN`, `PRON`, `VERB`, `ADJ` and `ADV` each contributes more than 6% improvement in semantics, compared to **function words** such as `CONJ` or `DET` with less than 4%. Hence, the decoder tends to favor content-related parts, and eliminate supporting prepositional or adverbial phrases. We also examine the effect of tree depth (equivalent to sentence length) on similarity changes, which is reported in [27]

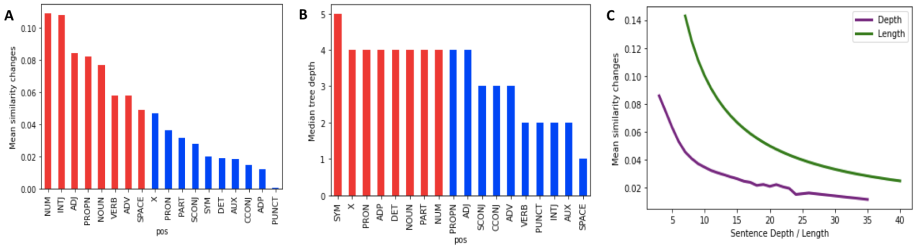

Fig. 2. Figure **A** explores the effect of grammatical functionality of words on similarity. **B** shows the uniform distribution of part-of-speeches across the sentence. **C** illustrates the allocation of important words in the sentence.

as a problem of position bias. We find that **this is not a major issue** to our work. Though a large portion of content is allocated towards the beginning of the sentence (corresponding to the top of a parse tree), the distributions of content words and function words are almost uniform sentence-wise, which is depicted in figures **B** and **C** accordingly. Thus, we rule out position bias and instead attribute this to human nature of writing. In the second phase, back translation further produces meaningful diversity (i.e. significantly reducing exact copies), which altogether contributes to effective simplification. We observe that back translation does more paraphrasing than simple lexical substitution. Therefore, sometimes the output sentences must be longer to be re-written in a simpler way, resulting in slightly less compression.

7 Conclusion

We implement the novel **family sampling** strategy on top of the regular beam-search-based decoding for sentence simplification. We directly tackle data scarcity issue by proposing an unsupervised framework that effectively generates hybrid outputs in a simple architecture, and achieves state-of-the-art results. Our proof-of-concept of Vietnamese simplification demonstrates it has plentiful rooms for improvement, and that the framework can also be applied to other languages.

References

1. Alva-Manchego, F., Martin, L., Bordes, A., Scarton, C., Sagot, B., Specia, L.: ASSET: A dataset for tuning and evaluation of sentence simplification models with multiple rewriting transformations. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. pp. 4668–4679. Association for Computational Linguistics, Online (Jul 2020). https://doi.org/10.18653/v1/2020.acl-main.424, https://aclanthology.org/2020.acl-main.424
2. Alva-Manchego, F., Martin, L., Scarton, C., Specia, L.: 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. pp. 49–54. Association for Computational Linguistics, Hong Kong, China (Nov 2019), https://www.aclweb.org/anthology/D19-3009
3. Brown, P.F., Cocke, J., Della Pietra, S.A., Della Pietra, V.J., Jelinek, F., Lafferty, J.D., Mercer, R.L., Roossin, P.S.: A statistical approach to machine translation. Computational Linguistics **16**(2), 79–85 (1990), https://aclanthology.org/J90-2002
4. Coster, W., Kauchak, D.: Simple English Wikipedia: A new text simplification task. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies. pp. 665–669. Association for Computational Linguistics, Portland, Oregon, USA (Jun 2011), https://aclanthology.org/P11-2117
5. Garbacea, C., Guo, M., Carton, S., Mei, Q.: 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). pp. 1086–1097. Association for Computational Linguistics, Online (Aug 2021). https://doi.org/10.18653/v1/2021.acl-long.88, https://aclanthology.org/2021.acl-long.88
6. Genzel, D., Charniak, E.: Variation of entropy and parse trees of sentences as a function of the sentence number. In: Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing. pp. 65–72 (2003), https://aclanthology.org/W03-1009
7. Guo, H., Pasunuru, R., Bansal, M.: Dynamic multi-level multi-task learning for sentence simplification. In: Proceedings of the 27th International Conference on Computational Linguistics (COLING 2018) (2018)
8. Kariuk, O., Karamshuk, D.: Cut: Controllable unsupervised text simplification. arXiv preprint arXiv:2012.01936 (2020)
9. Kitaev, N., Klein, D.: Constituency parsing with a self-attentive encoder. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 2676–2686. Association for Computational Linguistics, Melbourne, Australia (Jul 2018). https://doi.org/10.18653/v1/P18-1249, https://aclanthology.org/P18-1249
10. Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., Herbst, E.: Moses: Open source toolkit for statistical machine translation. In: Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions. pp. 177–180. Association for Computational Linguistics, Prague, Czech Republic (Jun 2007), https://aclanthology.org/P07-2045
11. Kumar, D., Mou, L., Golab, L., Vechtomova, O.: Iterative edit-based unsupervised sentence simplification. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. pp. 7918–7928. Association for Computational Linguistics (2020)
12. Levenshtein, V.I.: Binary codes capable of correcting deletions, insertions and reversals. Soviet Physics Doklady **10**(8), 707–710 (Feb 1966), doklady Akademii Nauk SSSR, V163 No4 845-848 1965
13. Maddela, M., Alva-Manchego, F., Xu, W.: Controllable text simplification with explicit paraphrasing. In: Proceedings of the North American Association for Computational Linguistics (NAACL) (2021)

14. Mallinson, J., Sennrich, R., Lapata, M.: Zero-shot crosslingual sentence simplification. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 5109–5126. Association for Computational Linguistics, Online (Nov 2020). https://doi.org/10.18653/v1/2020.emnlp-main.415, https://aclanthology.org/2020.emnlp-main.415
15. Marcus, M.P., Santorini, B., Marcinkiewicz, M.A.: Building a large annotated corpus of English: The Penn Treebank. Computational Linguistics **19**(2), 313–330 (1993), https://aclanthology.org/J93-2004
16. Martin, L., de la Clergerie, É., Sagot, B., Bordes, A.: Controllable sentence simplification. In: Proceedings of the 12th Language Resources and Evaluation Conference. pp. 4689–4698. European Language Resources Association, Marseille, France (May 2020), https://aclanthology.org/2020.lrec-1.577
17. Martin, L., Fan, A., de la Clergerie, É., Bordes, A., Sagot, B.: Muss: Multilingual unsupervised sentence simplification by mining paraphrases. arXiv preprint arXiv:2005.00352 (2021)
18. Narayan, S., Gardent, C.: Hybrid simplification using deep semantics and machine translation. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 435–445. Association for Computational Linguistics, Baltimore, Maryland (Jun 2014). https://doi.org/10.3115/v1/P14-1041, https://aclanthology.org/P14-1041
19. Nguyen, P.T., Vu, X.L., Nguyen, T.M.H., Nguyen, V.H., Le, H.P.: Building a large syntactically-annotated corpus of Vietnamese. In: Proceedings of the Third Linguistic Annotation Workshop (LAW III). pp. 182–185. Association for Computational Linguistics, Suntec, Singapore (Aug 2009), https://aclanthology.org/W09-3035
20. Niklaus, C., Cetto, M., Freitas, A., Handschuh, S.: Transforming complex sentences into a semantic hierarchy. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. pp. 3415–3427. Association for Computational Linguistics, Florence, Italy (Jul 2019). https://doi.org/10.18653/v1/P19-1333, https://aclanthology.org/P19-1333
21. Nisioi, S., Štajner, S., Ponzetto, S.P., Dinu, L.P.: Exploring neural text simplification models. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). pp. 85–91. Association for Computational Linguistics, Vancouver, Canada (Jul 2017). https://doi.org/10.18653/v1/P17-2014, https://aclanthology.org/P17-2014
22. Prabhumoye, S., Tsvetkov, Y., Salakhutdinov, R., Black, A.W.: Style transfer through back-translation. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 866–876. Association for Computational Linguistics, Melbourne, Australia (Jul 2018). https://doi.org/10.18653/v1/P18-1080, https://aclanthology.org/P18-1080
23. Reimers, N., Gurevych, I.: Making monolingual sentence embeddings multilingual using knowledge distillation. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 4512–4525. Association for Computational Linguistics, Online (Nov 2020). https://doi.org/10.18653/v1/2020.emnlp-main.365, https://aclanthology.org/2020.emnlp-main.365
24. Reimers, N., Gurevych, I.: Making monolingual sentence embeddings multilingual using knowledge distillation. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics (11 2020), https://arxiv.org/abs/2004.09813

25. Salazar, J., Liang, D., Nguyen, T.Q., Kirchhoff, K.: Masked language model scoring. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. pp. 2699–2712. Association for Computational Linguistics, Online (Jul 2020). https://doi.org/10.18653/v1/2020.acl-main.240, https://aclanthology.org/2020.acl-main.240
26. Sampson, G.: Depth in english grammar. Journal of Linguistics **33**(1), 131–151 (1997)
27. Schumann, R., Mou, L., Lu, Y., Vechtomova, O., Markert, K.: Discrete optimization for unsupervised sentence summarization with word-level extraction. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. pp. 5032–5042 (2020)
28. See, A., Liu, P.J., Manning, C.D.: Get to the point: Summarization with pointer-generator networks. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 1073–1083. Association for Computational Linguistics, Vancouver, Canada (Jul 2017). https://doi.org/10.18653/v1/P17-1099, https://aclanthology.org/P17-1099
29. Surya, S., Mishra, A., Laha, A., Jain, P., Sankaranarayanan, K.: Unsupervised neural text simplification. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. pp. 2058–2068. Association for Computational Linguistics, Florence, Italy (Jul 2019). https://doi.org/10.18653/v1/P19-1198, https://www.aclweb.org/anthology/P19-1198
30. Vu, T., Nguyen, D.Q., Nguyen, D.Q., Dras, M., Johnson, M.: VnCoreNLP: A Vietnamese natural language processing toolkit. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations. pp. 56–60. Association for Computational Linguistics, New Orleans, Louisiana (Jun 2018). https://doi.org/10.18653/v1/N18-5012, https://aclanthology.org/N18-5012
31. Wubben, S., van den Bosch, A., Krahmer, E.: Sentence simplification by monolingual machine translation. In: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 1015–1024. Association for Computational Linguistics, Jeju Island, Korea (Jul 2012), https://aclanthology.org/P12-1107
32. Xu, W., Callison-Burch, C., Napoles, C.: Problems in current text simplification research: New data can help. Transactions of the Association for Computational Linguistics **3**, 283–297 (2015). https://doi.org/10.1162/tacl_a_00139, https://aclanthology.org/Q15-1021
33. Xu, W., Napoles, C., Pavlick, E., Chen, Q., Callison-Burch, C.: Optimizing statistical machine translation for text simplification. Transactions of the Association for Computational Linguistics **4**, 401–415 (2016). https://doi.org/10.1162/tacl_a_00107, https://aclanthology.org/Q16-1029
34. Xu, Y., Reitter, D.: Convergence of syntactic complexity in conversation. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). pp. 443–448. Association for Computational Linguistics, Berlin, Germany (Aug 2016). https://doi.org/10.18653/v1/P16-2072, https://aclanthology.org/P16-2072
35. Zhang, X., Lapata, M.: Sentence simplification with deep reinforcement learning. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. pp. 584–594. Association for Computational Linguistics, Copenhagen, Denmark (Sep 2017). https://doi.org/10.18653/v1/D17-1062, https://aclanthology.org/D17-1062

36. Zhang, Z., Ren, S., Liu, S., Wang, J., Chen, P., Li, M., Zhou, M., Chen, E.: Style transfer as unsupervised machine translation. arXiv preprint arXiv:1808.07894 (2018)
37. Zhao, Y., Chen, L., Chen, Z., Yu, K.: Semi-supervised text simplification with back-translation and asymmetric denoising autoencoders. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34, pp. 05:9668–9675 (2020)
38. Zhu, Z., Bernhard, D., Gurevych, I.: A monolingual tree-based translation model for sentence simplification. In: Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010). pp. 1353–1361. Coling 2010 Organizing Committee, Beijing, China (Aug 2010), https://aclanthology.org/C10-1152