Unsupervised Sentence Simplification via Dependency Parsing

Anonymous ACL submission

Abstract

Text simplification is the task of rewriting a text so that it is readable and easily understood. In this paper, we propose a simple yet novel unsupervised sentence simplification system that harnesses parsing structures to produce linguistically effective simplifications. This means our model is capable of introducing substantial modifications to simplify a sentence while maintaining its original semantics and adequate fluency. 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. Furthermore, we demonstrate our framework's extensibility to other languages via a proof-of-concept on Vietnamese data. Code for reproduction is anonymously published at https://anonymous.4open.science/r/USDP-744B/.

1 Introduction

Text simplification (TS) contributes to promoting social inclusion by making information more accessible to people with reading comprehension problems such as second-language learners, non-experts or those suffering from cognitive impairment (Siddharthan, 2014; Stajner, 2021). Simplification may involve **splitting** and/or **reordering** parts of the text to produce simpler syntactic structures. **Lexical transformation** can also be performed through the substitution of easier and more familiar vocabularies. Another way is to simplify a text is **deleting** redundant details producing an output shorter in length. On the other hand, the output can become longer with extra information added to provide explanations for difficult concepts i.e., **elaboration**. Regardless of the operations, the simple variants must preserve the key meaning of the original texts.

Previous studies criticize existing systems for being opaque, suboptimal and semantic compromising (Garbacea et al., 2021; Maddela et al., 2021; Stajner, 2021). Table 1 illustrates how the meaning is sensitive to minor changes to the sentence, thereby easily sacrificed. Generally, TS works must be mindful of the trade-off of **Simplicity** vs. **Fluency** and **Adequacy** (semantics similarity). Although some works explicitly incorporate all three properties in the training objectives (Laban et al., 2021; Zhang and Lapata, 2017; Kumar et al., 2020), their evaluations do not clearly explain whether or not simplicity is induced as the cost of fluency and adequacy. By exploiting deep dependency parsing, we contribute a novel unsupervised strategy `USDP` that strictly enforces grammatically fluent simplification while ensuring the most important ideas are retained. We shed light on how our model achieves the balance of these three properties through both automatic metrics and human judgement, which at the same time substantiate our superiority over unsupervised counterparts and competitive performance against supervised systems.

Original sentence	**Simple variant**
In Ethiopia, HIV disclosure is low	In Ethiopia , HIV is low
Mustafa Shahbaz , 26 , was shopping for books about science	Mustafa Shahbaz, 26 years old , was a group of books about science
Healthy diet linked to lower risk of chronic lung disease	Healthy diet linked to lung disease

Table 1: Examples of logic errors produced by `ACCESS` (Martin et al., 2020) and `DMLMTL` (Guo et al., 2018), taken from (Garbacea et al., 2021).

Whereas most models are restricted to the language of the data they are trained on, we demonstrate that our framework readily extends to other languages and adapt it to simplify Vietnamese texts as proof-of-concept. We also address interpretability by providing linguistically-motivated empirical evidence confirming the intuition behind our frame-

work.

2 Related Work

Supervised TS. Earlier works inherit techniques from statistical machine translation (Brown et al., 1990) 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 (Coster and Kauchak, 2011; Koehn et al., 2007; Narayan and Gardent, 2014; Wubben et al., 2012). Meanwhile, in syntax-based simplification (Zhu et al., 2010), the alignment units are syntactic components. The first neural sequence-to-sequence text simplification system is proposed in (Nisioi et al., 2017). Utilizing the same architecture, other works (Guo et al., 2018; Zhang and Lapata, 2017) further train reinforcement-learning-based models, based on a reward function that is a weighted sum of three component rewards: simplicity, relevance and fluency. Audience Centric Sentence Simplification ACCESS is a recent supervised state-of-the-art approach (Martin et al., 2020) that condition the simplified outputs on different attributes of text complexity. These models rely on parallel corpora to implicitly learn hybrid transformation patterns. Despite impressive results, the scarcity of high-quality and large-scale datasets heavily impedes progress in supervised TS (Alva-Manchego et al., 2020). The attention has thus been shifted towards semi-supervised and unsupervised approaches.

Semi-Supervised TS. Instead of using aligned data, Zhao et al. 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. 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, and mining cross-lingual data is recently shown useful for TS task (Mallinson et al., 2020).

Unsupervised TS. Surya et al. 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. DisSim (Niklaus et al., 2019) is another effort focusing on splitting and deletion by applying 35 hand-crafted grammar rules over a constituency parse tree. Recent works tend to favor edit and decoding-based approach. This line of work is advantageous since not only can the system generate hybrid outputs without relying on aligned datasets, but also allow for quality control explicitly via a scoring function balancing simplicity, fluency and semantics preservation. The algorithm of Kumar et al. iteratively edits a given complex sentence to make it simpler using four operations: removal, extraction, reordering and substitution, while Kariuk and Karamshuk implement beam search with simplicity-aware penalties for sentence simplification without supervision. In the same setting as (Zhang and Lapata, 2017), KiS (Laban et al., 2021) revisits reinforcement learning and tackles simplification for paragraphs in the unsupervised manner. However, the method involves end-to-end training on multiple Transformer-based models, which is computationally expensive and makes it challenging to extend to new settings. In contrast, we propose a light-weight solution making 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 out-of-distribution examples. Although still focusing on sentence simplification, in **Discussion** section, we provide directions on how our framework can be flexibly adapted for various purposes, including paragraph-level simplification.

Specifically, we improve on the existing decoding procedure through a linguistics-based unsupervised framework for sentence simplification. We perform structural and lexical simplification sequentially, rather than simultaneously like previous works, since it would support interpretation and allow for more controllability. This sequential approach is also adopted in (Maddela et al., 2021), which leverages DisSim together with a self-designed paraphrasing system. We first develop a **stand-alone decoding framework** for structural simplification, then adopt **back translation** for lexical simplification and paraphrasing. After studying prior works, we find that **splitting** and **elaboration** are difficult to implement without labeled data or heavily injected grammar rules, while our goal is to maximize the capacity of TS system when deprived of external knowledge. An interesting discovery is that back translation in phase 2 is a convenient technique, in that it can also perform **reordering** (as part of the rewriting process), if

such an operation is necessary to produce a familiar structure. Thus, in the first phase, we choose to focus on **deletion** - the operation that easily leads to poor adequacy if not properly done.

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 operating in a much more efficient search space induced by making use of dependency relations among words in a sentence. Though this phase is mainly about deletion, the system can also perform **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. Figure 1 illustrates a running example for how our procedure works.

3.1 Structural Simplification

3.1.1 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 from its sentence. This motivates our search objective function as follows

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

where α is the relative weight on Fluency score. The reason why the weights of f_{sim} and f_{depth} are the same is given below under **System Details** section. 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 $f_{sim}(c, s_{1:t}) = cos(e_c, e_{s_{1:t}})$. In our work, we use the pre-trained sentence-BERT model (SBERT) (Reimers and Gurevych, 2020a) to derive semantically meaningful sentence embeddings to calculate cosine similarity. Other unsupervised works, in constrast, use IDF weighted average of unigram embeddings (Kumar et al., 2020; Schumann et al., 2020).

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 pre-trained 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 (Genzel and Charniak, 2003; Sampson, 1997; Xu and Reitter, 2016). Deeper trees indicate more complicated structures, and it is recently shown in ACCESS (Martin et al., 2020) 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.

3.1.2 Search Space

Deletion is a form of extractive summarization by nature, motivating us to adopt the word-extraction method proposed by (Schumann et al., 2020). 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**.

3.1.3 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 (Schumann et al., 2020), 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 (Prabhumoye et al., 2018; Zhang et al., 2018) to disentangle the content 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 (Xu et al., 2016) and PWKP (Zhang and Lapata, 2017). 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 (Zhang and Lapata, 2017). 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* (Xu et al., 2015) 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 (See et al., 2017) 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 Underthesea

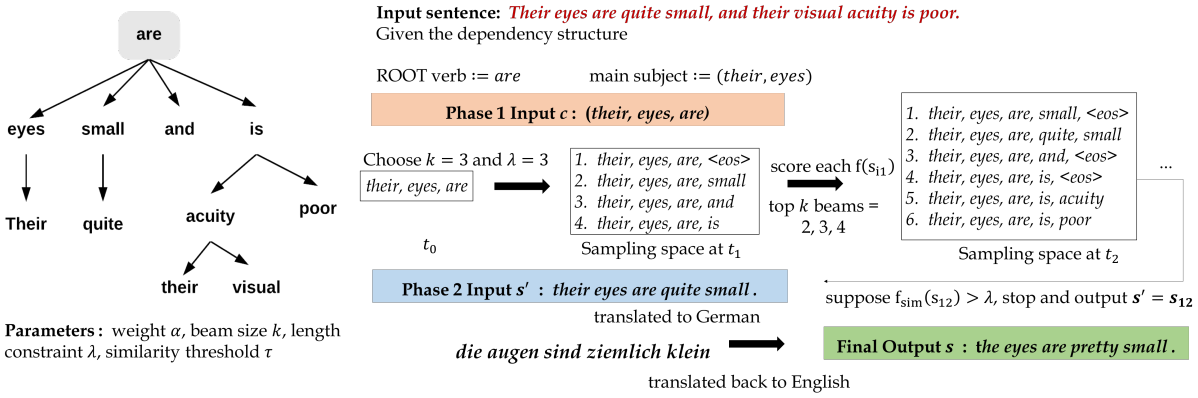

Figure 1: At each search step, a candidate token is sampled from the direct children of previously generated token. Score each candidate according to the objective function and select k hypotheses with highest scores for the next generation step. The decoder terminates once there is a sequence reaching length λ with at least similarity τ.

NLP [1].

4.2 System Details

We utilize SpaCy[2] and Berkeley Neural Parser (Kitaev and Klein, 2018) for constituent and dependency parsing. Since SpaCy does not currently support Vietnamese, we parse Vietnamese texts through VnCoreNLP[3] (Vu et al., 2018). VnCoreNLP is built upon Vietnamese Treebank (Nguyen et al., 2009), which contains 10200 constituent trees formatted similarly to Penn Treebank (Marcus et al., 1993). We obtain sentence embeddings from SBERT model `paraphrase-mpnet-base-v2` (Reimers and Gurevych, 2020b) for English, and the multilingual version `distiluse-base-multilingual-cased-v2` for Vietnamese (Reimers and Gurevych, 2020a). 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.

[1] `github.com/undertheseanlp/resources`
[2] `spacy.io`
[3] `github.com/vncorenlp/VnCoreNLP`
[4] `nltk.org`
[5] `translate.google.com`

4.3 Competing Models

We benchmark our system against both supervised and unsupervised models. Supervised systems include `PBMT-R` (Wubben et al., 2012), `SBMT-SARI` (Xu et al., 2016), `Dress` / `Dress-Ls` (Zhang and Lapata, 2017) and recent state-of-the-art `ACCESS` (Martin et al., 2020). We also consider semi-supervised `BTTS` / `BTRLTS` / `BTTS100` (Zhao et al., 2020) and unsupervised counterparts `UNTS` / `UNTS10K` (Surya et al., 2019) and `RM+EX` / `RM+EX+LS` / `RM+EX+RO` / `RM+EX+LS+RO` (Kumar et al., 2020).

5 Results

5.1 Automatic Evaluation

We use EASSE package (Alva-Manchego et al., 2019) to compute standardized simplification met-

rics 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 (Alva-Manchego et al., 2020; Wubben et al., 2012; Xu et al., 2016) and FKGL only 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 (Salazar et al., 2020), 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 2, 3 and 4 present results of automatic evaluation respectively on TurkCorpus, PWKP and CP_Vietnamese-VLC, both before and after Back translation (**BT**) implemented.

5.1.1 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. Figure 2 and 3 visualize how well our base model balances simplicity with adequacy and fluency compared to other methods. Examples of system outputs are provided in Appendix B.

5.1.2 PWKP

As Kumar et al. (Kumar et al., 2020) 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 preserving slightly higher semantics than `RM+EX+LS+RO`. The fluency scores of simplifications from all automated systems remain far behind human outputs.

5.1.3 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 (Levenshtein, 1966) (**LevSIM**) which demonstrates the structures of the output sentences do not deviate significantly from the original. Results in Table 5 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

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	-2.88	**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	-3.05	**0.96**	34.44	0.94	68.57	33.82

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

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 the previous studies (Kumar et al., 2020; Maddela et al., 2021; Zhao et al., 2020), 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 5, 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.

5.3 Controllability

Table 6 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

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

Table 4: Results of USDP-Base on CP_Vietnamese-VLC

Model	Fluent	Adequate	Simple
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

Table 5: Human Evaluation Results on TurkCorpus (English) and CP_Vietnamese-VLC (Vietnamese) datasets.

Value	CR↓	%D↑	SARI↑	SIM↑
Effect of λ at $\tau = 0.95$				
0.25	0.95	0.08	33.60	0.98
0.50	0.95	0.08	33.60	0.98
0.75	0.96	0.07	33.47	0.98
1.00	0.99	0.04	32.86	0.98
Effect of τ at $\lambda = 0.5$				
0.70	0.81	0.22	33.63	0.92
0.80	0.84	0.19	33.77	0.94
0.90	0.91	0.12	33.80	0.97
0.95	0.95	0.08	33.60	0.98

Table 6: Effects of threshold values on simplification quality of 100 sentences from TurkCorpus. Both are evaluated on USDP-Base.

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.

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. 4), 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 (Schumann et al., 2020) 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

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

Linguistics, pages 4668–4679, Online. Association for Computational Linguistics.

Fernando Alva-Manchego, Louis Martin, Carolina Scarton, and Lucia Specia. 2019. 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.

Peter F. Brown, John Cocke, Stephen A. Della Pietra, Vincent J. Della Pietra, Fredrick Jelinek, John D. Lafferty, Robert L. Mercer, and Paul S. Roossin. 1990. A statistical approach to machine translation. *Computational Linguistics*, 16(2):79–85.

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

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.

Dmitriy Genzel and Eugene Charniak. 2003. 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*, pages 65–72.

Han Guo, Ramakanth Pasunuru, and Mohit Bansal. 2018. Dynamic multi-level multi-task learning for sentence simplification. In *Proceedings of the 27th International Conference on Computational Linguistics (COLING 2018)*.

Oleg Kariuk and Dima Karamshuk. 2020. Cut: Controllable unsupervised text simplification. *arXiv preprint arXiv:2012.01936*.

Nikita Kitaev and Dan Klein. 2018. Constituency parsing with a self-attentive encoder. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2676–2686, Melbourne, Australia. Association for Computational Linguistics.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. 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*, pages 177–180, Prague, Czech Republic. 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. Association for Computational Linguistics.

Philippe Laban, Tobias Schnabel, Paul N. 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*, volume 1.

Vladimir Iosifovich Levenshtein. 1966. Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics Doklady*, 10(8):707–710. Doklady Akademii Nauk SSSR, V163 No4 845-848 1965.

Mounica Maddela, Fernando Alva-Manchego, and Wei Xu. 2021. Controllable text simplification with explicit paraphrasing. In *Proceedings of the North American Association for Computational Linguistics (NAACL)*.

Jonathan Mallinson, Rico Sennrich, and Mirella Lapata. 2020. Zero-shot crosslingual sentence simplification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5109–5126, Online. Association for Computational Linguistics.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

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.

Louis Martin, Angela Fan, Éric de la Clergerie, Antoine Bordes, and Benoît Sagot. 2021. Muss: Multilingual unsupervised sentence simplification by mining paraphrases. *arXiv preprint arXiv:2005.00352*.

Shashi Narayan and Claire Gardent. 2014. 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)*, pages 435–445, Baltimore, Maryland. Association for Computational Linguistics.

Phuong-Thai Nguyen, Xuan-Luong Vu, Thi-Minh-Huyen Nguyen, Van-Hiep Nguyen, and Hong-Phuong Le. 2009. Building a large syntactically-annotated corpus of Vietnamese. In *Proceedings of the Third Linguistic Annotation Workshop (LAW III)*, pages 182–185, Suntec, Singapore. Association for Computational Linguistics.

Christina Niklaus, Matthias Cetto, André Freitas, and Siegfried Handschuh. 2019. Transforming complex sentences into a semantic hierarchy. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3415–3427, Florence, Italy. Association for Computational Linguistics.

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

Shrimai Prabhumoye, Yulia Tsvetkov, Ruslan Salakhutdinov, and Alan W Black. 2018. Style transfer through back-translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 866–876, Melbourne, Australia. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2020a. Making monolingual sentence embeddings multilingual using knowledge distillation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4512–4525, Online. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2020b. 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.

Julian Salazar, Davis Liang, Toan Q. Nguyen, and Katrin Kirchhoff. 2020. Masked language model scoring. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2699–2712, Online. Association for Computational Linguistics.

Geoffrey Sampson. 1997. Depth in english grammar. *Journal of Linguistics*, 33(1):131–151.

Raphael Schumann, Lili Mou, Yao Lu, Olga Vechtomova, and Katja Markert. 2020. Discrete optimization for unsupervised sentence summarization with word-level extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5032–5042.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. 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)*, pages 1073–1083, Vancouver, Canada. Association for Computational Linguistics.

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

Sanja Stajner. 2021. Automatic text simplification for social good: Progress and challenges. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2637–2652, Online. Association for Computational Linguistics.

Sai Surya, Abhijit Mishra, Anirban Laha, Parag Jain, and Karthik Sankaranarayanan. 2019. Unsupervised neural text simplification. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2058–2068, Florence, Italy. Association for Computational Linguistics.

Thanh Vu, Dat Quoc Nguyen, Dai Quoc Nguyen, Mark Dras, and Mark Johnson. 2018. 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*, pages 56–60, New Orleans, Louisiana. Association for Computational Linguistics.

Sander Wubben, Antal van den Bosch, and Emiel Krahmer. 2012. Sentence simplification by monolingual machine translation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1015–1024, Jeju Island, Korea. 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.

Yang Xu and David Reitter. 2016. Convergence of syntactic complexity in conversation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 443–448, Berlin, Germany. 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.

Zhirui Zhang, Shuo Ren, Shujie Liu, Jianyong Wang, Peng Chen, Mu Li, Ming Zhou, and Enhong Chen. 2018. Style transfer as unsupervised machine translation. *arXiv preprint arXiv:1808.07894*.

Yanbin Zhao, Lu Chen, Zhi Chen, and Kai Yu. 2020. Semi-supervised text simplification with back-translation and asymmetric denoising autoencoders. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 05:9668–9675.

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 Additional Visualization

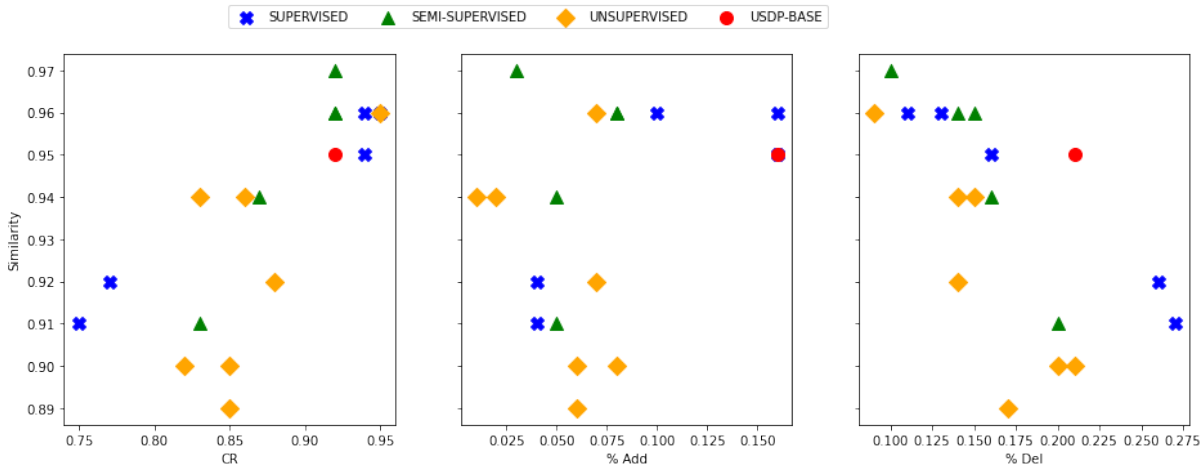

Figure 2: Visualization of systems' capacity to balance **Adequacy** with **Simplicity** on TurkCorpus

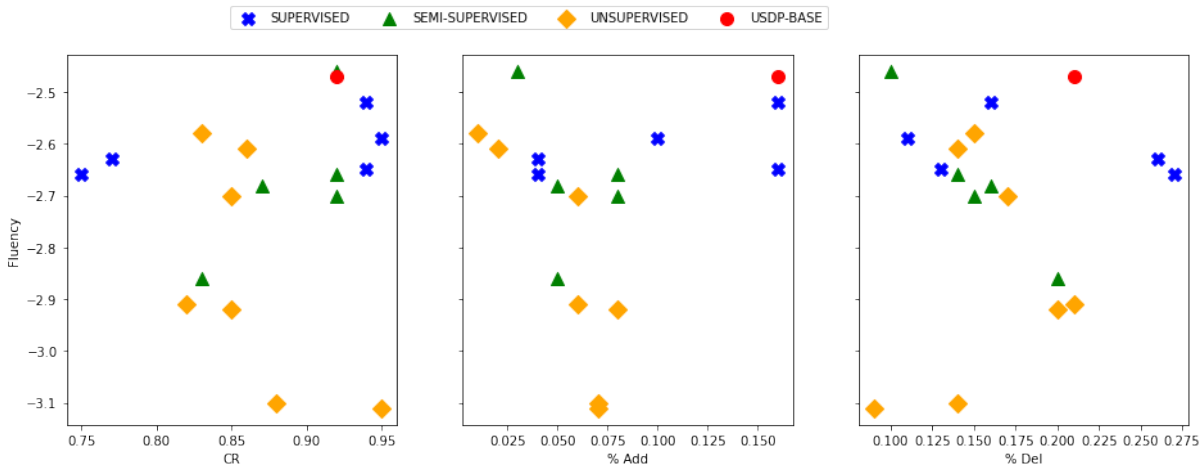

Figure 3: Visualization of systems' capacity to balance **Fluency** with **Simplicity** on TurkCorpus

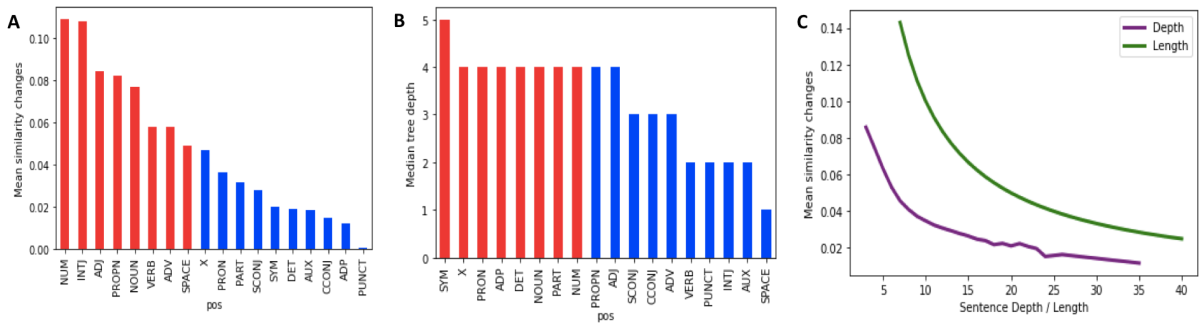

Figure 4: Plot **A** explores the effect of grammatical functionality of words on similarity. **B** shows the uniform distribution of part-of-speeches across the sentence. **C** illustrates the allocation of important words in the sentence.

B Qualitative Evaluation

Example of deleting prepositional/adjective phrases	
Original sentence	*Jeddah is the **principal** gateway to Mecca, Islam's holiest city, which able-bodied Muslims **are required to visit at least once** in their lifetime.*
System outputs	
`RM+EX+LS`	*Jeddah is the principal gateway to Mecca, Islam's Holiest city, which Able-Bodied Muslims are required to visit at least once in their lifetime.*
`USDP-Base`	
With BT	*Jeddah is the **main** gateway to Mecca, the holiest city in Islam that Muslim people **had to visit** during their lifetime.*
Without BT	*Jeddah is the principal gateway to Mecca , Islam 's holiest city , which able - bodied Muslims required - visit in their lifetime.*
Example of summarization	
Original sentence	*At four-and-a-half years old he was left to fend for himself on the streets of northern Italy for the next four years, **living in various orphanages and roving through towns with groups of other homeless children.***
System outputs	
`RM+ES+LS`	*At Four-And-A-Half years old he was left to fend for himself on the walls of northern Italy for the next four years.*
`USDP-Base`	
With BT	*At the age of four and a half, he had to support himself on the streets of northern Italy for the next four years, **wandering through the cities living in various orphanages.***
Without BT	*At four - and - a - half years old he was left - to fend for himself on the streets of northern Italy for the next four years , living in various orphanages - roving through towns..*
Example of chunking	
Original sentence	*In late 2004, Suleman made headlines by cutting Howard Stern's radio show from four Citadel stations, citing Stern's frequent discussions regarding his upcoming move to Sirius Satellite Radio.*
System outputs	
`RM+ES+LS`	*In late 2004, Suleman made headlines by cutting Howard Stern'S radio show from four Citadel trains, reporting Stern'S serious questions.*
`USDP-Base`	
With BT	*In late 2004, Suleman made headlines - **by cutting Howard Stern's radio show from four Citadel stations** - **citing Stern's discussions** - **regarding the upcoming move**.*
Without BT	*In late 2004, Suleman made headlines - by cutting Howard Stern's radio show from four Citadel stations - citing Stern's discussions - regarding upcoming move.*
Example of simplistic paraphrasing	
Original sentence	*Fearing that Drek will destroy the galaxy, Clank asks Ratchet to **help him find the famous superhero Captain Qwark, in an effort to stop Drek.***
System outputs	
`RM+ES+LS`	*Fearing that Drek will bring the universe, Clank asks Ratchet to help him find the famous Superhero captain Qwark, in an attempt to get Drek.*
`USDP-Base`	
With BT	*Fearing Drek might destroy the galaxy, Clank asks Ratchet to **find the superhero in order to stop Drek**.*
Without BT	*Fearing that Drek will destroy the galaxy , Clank asks Ratchet - help find the superhero - in effort stop Drek.*

Table 7: Qualitative results on TurkCorpus.