Controllable Sentence Simplification in Swedish using Control Prefixes and Mined Paraphrases

Julius Monsen†, Arne Jönsson
Linköping University
julmo634@student.liu.se, arne.jonsson@liu.se

Abstract

Making information accessible to diverse target audiences, including individuals with dyslexia and cognitive disabilities, is crucial. Automatic Text Simplification (ATS) systems aim to facilitate readability and comprehension by reducing linguistic complexity. However, they often lack customizability to specific user needs, and training data for smaller languages can be scarce. This paper addresses ATS in a Swedish context, using methods that provide more control over the simplification. A dataset of Swedish paraphrases is mined from large amounts of text and used to train ATS models utilizing prefix-tuning with control prefixes. We also introduce a novel data-driven method for selecting complexity attributes for controlling the simplification and compare it with previous approaches. Evaluation of the trained models using SARI and BLEU demonstrates significant improvements over the baseline – a fine-tuned Swedish BART model – and compared to previous Swedish ATS results. These findings highlight the effectiveness of employing paraphrase data in conjunction with controllable generation mechanisms for simplification. Additionally, the set of explored attributes yields similar results compared to previously used attributes, indicating their ability to capture important simplification aspects.

Keywords: Natural Language Generation, Simplification, Text Mining

1. Introduction

The goal of Automatic Text Simplification (ATS) is to reduce a text's linguistic complexity to facilitate both readability and understandability whilst preserving the semantic meaning to the largest extent possible (Shardlow, 2014). This task is commonly framed as a monolingual machine translation task where a high-complexity text is translated into a lower-complexity one (Alva-Manchego et al., 2020b). This process may entail adjustments to the text's syntactic structure and/or lexicality. These types of simplifications are often considered the two main forms of simplification (Saggion, 2017).

Much of ATS research has been focused on developing systems that produce generic simplifications without the possibility of tailoring them to meet the requirements of different users. However, in reality, the desired result of a simplification largely depends on the target group and individuals within the target group. Individuals with dyslexia often face challenges with visual decoding of words, particularly lengthy, low-frequency, homophonic, orthographically similar, new, or non-words (Rennes, 2022). In contrast, individuals with intellectual disabilities often have impaired working memory and executive functions, which impact their reading ability and comprehension. Moreover, a typical second language learner may experience difficulties related to vocabulary, unfamiliarity with cultural phenomena, or grammar. Hence, there is no one-size-fits-all solution. Ideally, a system should be able to produce simplifications with different characteristics that cater to different users.

ATS can be carried out using different approaches or combinations thereof. The most successful approach has been to consider the task of simplification as a seq2seq problem and utilize attention-based encoder-decoder architectures, such as the Transformer (Vaswani et al., 2017), to solve it (Alva-Manchego et al., 2020b). Unlike traditional approaches, these models do not require feature extraction and can perform multiple complex text transformations simultaneously. They are also primarily data-driven, meaning they rely on large amounts of parallel data consisting of standard-simple text pairs to learn simplification transformations (Alva-Manchego et al., 2020b). However, such resources are predominantly available in English, which is the case with most language resources, as highlighted by Ruder et al. (2022). In low-resourced languages, building high-performing ATS systems becomes more challenging.

There has been increased interest in developing language-agnostic approaches to address the challenge of ATS in languages where parallel data is not readily available. One such method was proposed by Kajiwara and Komachi (2018), who utilized a monolingual corpus to create pseudo-parallel data. The process involved dividing a corpus into two parts consisting of standard and simplified texts, respectively, and then aligning the two parts using unsupervised alignment algorithms to obtain the best matching standard-simplified pairs. Another more

†Currently affiliated with Örebro University. Email: julius.monsen@oru.se.

recent alternative was proposed by Martin et al. (2022). Instead of using parallel or pseudo-parallel simplification data, they mined paraphrases from large amounts of web-scraped text data by pairing sentences using similarity measures. Moreover, with this approach, there is no need to separate the data into standard and simplified parts, making it more efficient and flexible.

Considering the limited availability of Swedish ATS datasets and the emergence of novel data mining methods, this paper aims to explore the feasibility of utilizing the paraphrase mining approach proposed by Martin et al. (2022) for Swedish ATS. Furthermore, the aim is to train and evaluate Swedish controllable text simplification models[1] that can generate high-quality user-adapted simplifications. More specifically, prefix-tuning (Li and Liang, 2021) with control prefixes (Clive et al., 2022) will be used. Lastly, we set out to explore what text complexity features are the most relevant from a data-driven perspective for controlling text simplification and how they impact model performance.

2. Related Work

There have been some attempts to tackle the problem of lacking parallel simplification data in Swedish. Holmer and Rennes (2023) constructed a pseudo-parallel Swedish simplification dataset following the approach proposed by Kajiwara and Komachi (2018). This work demonstrated that such an approach could be a good option for smaller languages. Holmer and Rennes (2023) trained ATS models with this dataset, establishing strong baselines for ATS in a Swedish context. Nevertheless, there has not been any work employing paraphrase mining techniques and controllable generation mechanisms for Swedish ATS.

Various techniques have been proposed for controlling certain aspects of the simplification generation to increase adaptability and ensure that users with different needs get suitable simplifications (Mallinson and Lapata, 2019; Maddela et al., 2021; Nishihara et al., 2019; Scarton and Specia, 2018; Martin et al., 2020; Clive et al., 2022). The common factor among many of these methods is that they incorporate additional input-dependent information into the data that the model can learn from. One category of information pertains to specific attributes of a text. For example, we may want the generated simplification to have a simple syntactic structure with as few subordinate clauses as possible, or we may want the text to have some specific length. By providing measures of syntactic complexity and text length for both the input and the target, the model can be trained to generate a simplified version of the input text that adheres to these attribute-specific criteria.

2.1. Controllable Simplification with Control Tokens

There are controllable simplification approaches utilizing discrete prompting-inspired methods. Martin et al. (2020) proposed ACCESS (AudienCe-CEntric Sentence Simplification), which conditions the generation in a seq2seq model by prepending control tokens (included in the vocabulary), e.g. <*LevSim 0.4*>, to the input sentence. These represent certain attributes of the target sentence in relation to the source sentence. Proxies for four attributes were used to represent specific features of the text simplification process: character length ratio between the source and the target (amount of compression), normalized character-level Levenshtein similarity between the source and the target (amount of paraphrasing), a word frequency-based measure of lexical complexity called WordRank, and the ratio of maximum depth of the dependency tree between the source and the target (syntactic complexity). Conditioning on these control tokens enabled out-of-the-box seq2seq models to outperform their standard counterparts on simplification benchmarks and provide new state-of-the-art results at the time. Moreover, these control parameters are intuitive and easy to interpret and can enable user adaptation.

Martin et al. (2022) applied ACCESS using the same control tokens in a new setting. More specifically, they used ACCESS in combination with BART (Lewis et al., 2020) and fine-tuned the model directly on paraphrases mined from a large corpus of text. This approach coined MUSS (Multilingual Unsupervised Sentence Simplification), yielded impressive state-of-the-art results on several benchmarks. For the paraphrase mining, FAISS (Facebook AI Similarity Search) (Johnson et al., 2017), was used to index sentence embeddings and perform fast Approximate Nearest Neighbour (ANN) search. Methods to identify varied paraphrases of sentences differing in some aspects, such as length and vocabulary, were developed. Furthermore, experiments were conducted using heuristics to make the target sentence simpler than the source sentence. However, surprisingly, mining raw paraphrases led to controllable models with better simplification performance while being more straightforward and requiring fewer prior assumptions.

[1] This work, along with much of the previous ATS research, considers a restricted form of text simplification, namely that of sentence simplification. In this setting, the input is a single source sentence, and the generated output can be composed of one sentence or multiple sentences resulting from a sentence split.

2.2. Controllable Simplification with Control Prefixes

Given a pre-trained language model, the predominant way of adaptation for downstream tasks has historically been to fine-tune all of its parameters. On the other hand, many alternatives have emerged that are parameter-efficient, as highlighted by Liu et al. (2021), some of which freeze most or all pre-trained parameters and fine-tune only a small number of additionally introduced parameters. Prefix-tuning (Li and Liang, 2021) is one such method that introduces continuous prompts (soft prompts) as extra context for the model to condition on during generation. During fine-tuning, these soft prompts, parameterized by the dedicated parameters θ, are learned while the base model parameters are frozen. Such continuous prompts are more expressive than discrete prompts since they are not constrained to embeddings of real tokens. More specifically, prefix-tuning augments the key-value pairs in the self-attention computation in each layer l with a prefix $\mathbf{P}_l \; \forall l \in \{1, ..., L\}$ which is drawn from $\mathbf{P}_\theta = \{\mathbf{P}_1, ..., \mathbf{P}_L\} \in \mathbb{R}^{p \times 2dL}$ parameterized by θ. Here, p is the prefix length (the number of additional key-value pairs in each self-attention computation).

Extending on prefix-tuning, Clive et al. (2022) proposed control prefixes. Unlike standard prefix-tuning, prefix-tuning with control prefixes uses dynamic soft prompts to leverage additional input-dependent information to condition on besides the general static prefix. These dynamic prefixes can provide finer-grained control over generation in conjunction with static ones. It was shown that tuning with control prefixes outperformed prefix-tuning as well as other existing approaches on several generation tasks, including ATS. The controllable attributes used by Clive et al. (2022) were the same as proposed by Martin et al. (2020) (described above) and they also used BART (Lewis et al., 2020) as the base model, with all its parameters frozen.

Formally, in addition to the general prefix $\mathbf{P}_\theta$, a control prefix $\mathbf{C}_\theta$ that changes based on the attribute-level information or guidance G for each input is also trained. Specifically, Clive et al. (2022) trained three sets of distinct prefixes corresponding to the attention classes E (encoder self-attention), Dc (decoder cross-attention) and Dm (masked decoder self-attention), respectively. The general prefix parameterized by θ, is then $\mathbf{P}_\theta = \{\mathbf{P}^E, \mathbf{P}^{Dc}, \mathbf{P}^{Dm}\}$.

Furthermore, assuming an attribute with R possible labels, the control prefix in the l-th layer is $\mathbf{C}_l = \{\mathbf{C}_{l,1}, ..., \mathbf{C}_{l,R}\}$, where $\mathbf{C}_{l,r} \; \forall r \in \{1, ..., R\}$ is the control prefix learned for the r-th attribute label. p_c denotes the control prefix length for this particular attribute and $\mathbf{C}_l$ is drawn from $\mathbf{C}_\theta \in \mathbb{R}^{p_c \times 6dLR}$.

If A is a function returning the control prefix for an attribute label indicated by G, then the key matrix $\mathbf{K}_l$ and the value matrix $\mathbf{V}_l$ can be modified according to Equation 1 where $\mathbf{K'}_l, \; \mathbf{V'}_l \in \mathbb{R}^{(p_c+p+m) \times d}$. As for the general prefix $\mathbf{P}_\theta$, there are three sets of control prefixes $\mathbf{C}_\theta = \{\mathbf{C}^E, \mathbf{C}^{Dc}, \mathbf{C}^{Dm}\}$, one for each attention class.

$$\begin{aligned} \mathbf{K'}_l &= [A(G)_{l,K}; \mathbf{P}_{l,K}; \mathbf{K}_l], \\ \mathbf{V'}_l &= [A(G)_{l,V}; \mathbf{P}_{l,V}; \mathbf{V}_l] \end{aligned} \tag{1}$$

Clive et al. (2022) demonstrated that prefix-tuning BART achieved comparable performance to fine-tuning BART and that the further extension with control prefixes yielded significantly better performance than both these approaches. Compared to Martin et al. (2022), control prefixes achieved comparable SARI scores while improving FKGL when evaluated on the ASSET (Alva-Manchego et al., 2020a) dataset.

3. Method

This section presents the method used for mining the Swedish paraphrase dataset, implementing the ATS models as well as training and evaluation procedures. Much of the methodology regarding paraphrase mining used in this study was inspired by the work of Martin et al. (2022), although there are some differences.

3.1. Mining Swedish Paraphrases

In the process of mining paraphrases, mainly two datasets were used. The paraphrases were mined from the Swedish part of the CC-100 corpus (Conneau et al., 2020), containing about 80GB of uncompressed text corresponding to $580,387,314$ paragraphs. It was constructed by processing January–December 2018 Common Crawl snapshots using CC-Net (Wenzek et al., 2020), an open-source repository with tools to download and clean Common Crawl snapshots. Each file in the CC-100 corpus contains documents separated by double newlines. A single newline separates paragraphs within a document. Moreover, $20,091,844$[2] sentences from the Swedish Culturomics Gigaword Corpus (Rødven Eide et al., 2016) was used in the process of creating sentence embeddings and filtering the mined paraphrases.

[2]Deduplicated and filtered based on length – a minimum of 30 characters and three tokens (separated by white space) and a maximum of 300 characters.

3.1.1. Preprocessing

The processing and filtering of sentences[3] was accomplished using a similar approach as Martin et al. (2022). Accordingly, white space, Unicode characters, and punctuation were normalized and sentences with less than 30 or more than 300 characters were discarded. An additional filter was added, ensuring that each sentence had more than three tokens separated by white space. This was to avoid sentences containing a single long token, such as a URL address. The remaining filters, also used by Martin et al. (2022), removed sentences containing more than 10% punctuation and sentences with low language model probability according to a 3-gram language model. A given sentence s was discarded if $p(s)/|s| < -0.7$, where $p(s)$ is the log probability for s and $|s|$ is the sentence length in characters. Martin et al. (2022) used the same threshold function and chose threshold values of -0.6 and -0.8 depending on the language and the training data for the 3-gram model.

The 3-gram language model was trained on all sentences extracted from the Swedish Culturomics Gigaword Corpus (Rødven Eide et al., 2016) using kenLM (Heafield, 2011). Before training, all sentences were tokenized using a `SentencePieceBPETokenizer` from the `tokenizers` library[4], trained on the same data. The filtering described, applied to sentence batches in parallel using 32 CPU cores, resulted in $652,205,685$ sentences.

3.1.2. Embedding Sentences

The next step was to create embeddings for the filtered sentences. KBLab's sentence-BERT model (Rekathati, 2021) was used for this purpose. The original model produced 768 dimensional embeddings. However, as about 650 million sentences were to be embedded, this would require a large amount of memory. Moreover, embedding all sentences would be computationally demanding, even on the available GPUs.

To speed things up and reduce the memory requirements, the sentence-BERT model was modified in two ways. First, the model was distilled using knowledge distillation in which a student model was initialized from a Swedish BERT model (Malmsten et al., 2020), but only 4 out of 12 layers were kept. Then the student model was trained to imitate the teacher model (the Swedish sentence-BERT model) by minimizing MSE between the student model's and the teacher model's embeddings. The monolingual sentence data used here comprised 10 million randomly sampled sentences from the Swedish Culturomics Gigaword Corpus (Rødven Eide et al., 2016). The code used for this knowledge distillation was adapted from the `knowledge_distillation.py` script provided by the `Sentence-Transformers` library (Reimers and Gurevych, 2019). The resulting model was more than twice as fast as the original one, retaining 95.2% of its performance on SweParaphrase (Isbister and Sahlgren, 2020), a gold-standard dataset purposed for evaluating semantic textual similarity. The distilled model's embedding dimension was additionally reduced from 768 to 128 with PCA using the `dimensionality_reduction.py` script also provided by the `Sentence-Transformers` library (Reimers and Gurevych, 2019). This reduced the storage requirement by a factor of 6. However, the trade-off in performance was slightly more noticeable as 94.1% of the original model's performance was retained after this.

3.1.3. Mining Paraphrases

Nearest neighbor search was performed with the embedded sentences using FAISS (Johnson et al., 2017). In Figure 1, the whole data mining procedure is illustrated. The type of FAISS index and the training and search parameters were chosen based on the work by Martin et al. (2022) and the FAISS wiki[5]. An `IVFPQ`-index was trained using 100 million of the sentence embeddings. The number of Voronoi cells was set to $32,768$. Moreover, 8-bit product quantization with eight sub-vectors per vector was carried out. Once the index had been trained, all sentence embeddings were added incrementally to the index.

Subsequently, all embeddings in the index were used as queries for ANN search. The top eight nearest neighbors (excluding the query embedding itself) were retrieved from the index for each query embedding. Here, the 16 cells with the nearest centroids were searched exhaustively. These paraphrases were then filtered based on L2 distance to the query embedding and density (relative distance compared to the other seven retrieved neighbors). Martin et al. (2022) used thresholds of 0.05 and 0.6, respectively. However, at this stage, these were set to 0.1 and 1.2 to include more paraphrases and allow for experimentation and more fine-grained filtering during postprocessing. Furthermore, an additional filter was applied enforcing a case-insensitive character-level Levenshtein distance greater or equal to 20%[6]. Almost identical pairs were thus discarded. All in all, this yielded $5,066,787$ paraphrase pairs.

[3] Paragraphs tokenized using NLTK (Bird et al., 2009)

[4] https://github.com/huggingface/tokenizers

[5] https://github.com/facebookresearch/faiss/wiki

[6] Martin et al. (2022) used two more filters only relevant for sequences with multiple sentences.

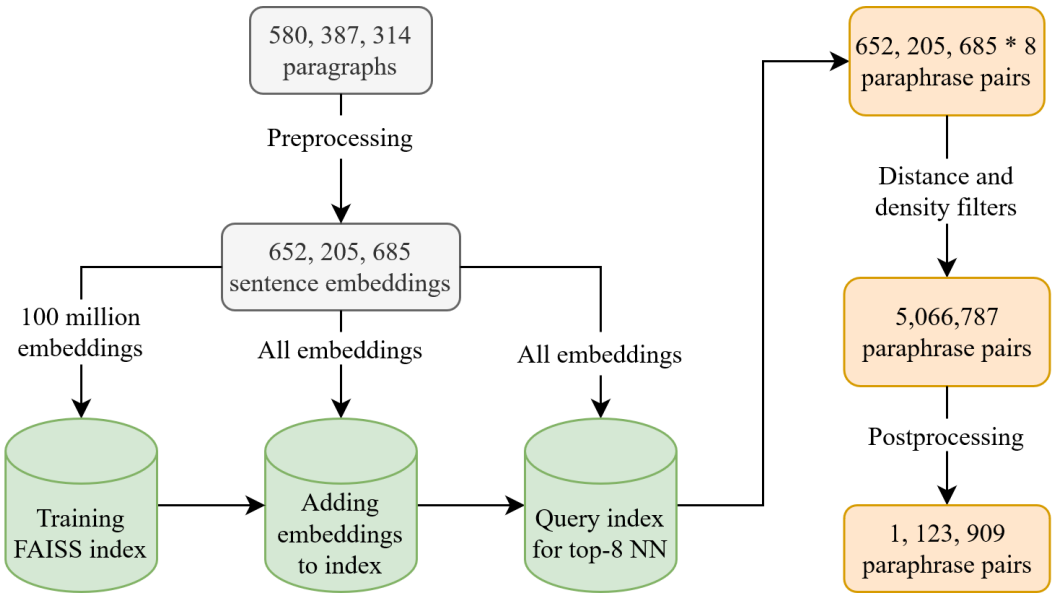

Figure 1: An overview of the paraphrase mining process.

3.1.4. Postprocessing

Before using the paraphrase pairs for training simplification models, instances containing identical pairs or where any of the sentences had words present in the Swedish version of LDNOOBW (List of Dirty, Naughty, Obscene, and Otherwise Bad Words)[7] were discarded. Due to this deduplication and filtering, $1,835,426$ pairs were removed.

Finally, a more fine-grained analysis was conducted to find appropriate filtering thresholds for L2 distance and density. The aim was to have approximately 1 million sentence pairs in the final dataset, similar to the datasets produced by Martin et al. (2022). They observed that performance drastically improved when increasing the number of mined pairs, indicating that efficient mining at scale is critical to performance. Based on this observation and analysis of L2 distances and densities, thresholds of 0.08 and 0.96 were chosen for L2 distance and density, respectively. After this filtering, $1,123,909$ sentence pairs remained. Furthermore, to give the model some sense of which of the sentences in a pair was simpler, the shortest sentence was used as the target and the longer one as the source, as a crude heuristic. The mean length of source and target sentences was 11.60 and 9.72 words, respectively. $10,000$ of these pairs were randomly set aside as validation data.

3.2. Data and Control Attributes

In addition to the mined paraphrase dataset described above, hereafter referred to as the Mined SWEdish Paraphrases (MSWEP) dataset[8], the PK18 dataset (Lindberg and Kindberg, 2018), a gold-standard dataset for Swedish text simplification, was used for evaluating the trained models.

The PK18 dataset consists of $1,005$ text pairs in Swedish. The texts were collected from four Swedish organizations and public institutions. Each text pair constitutes a standard text and a simplified version of it. The simplifications were manually written by experts working with making information more accessible. These were then aligned manually with the standard counterparts. Both standard and simplified texts consist of one to five sentences. As Holmer and Rennes (2023) points out, PK18 is currently the largest and most suitable alternative for evaluating ATS systems in the Swedish language. As this work is limited to investigating simplification on the sentence level, only pairs with one sentence in respective versions were used, similar to Holmer and Rennes (2023). The remaining subset then consisted of 467 sentence pairs. These were used for evaluating the Swedish simplification models.

3.2.1. Selecting Control Attributes

As previously mentioned, Martin et al. (2020) selected control attributes manually to represent specific grammatical attributes of the text simplification process. In this work, a more data-driven approach was applied. Using a subset of text complexity features from SCREAM (Swedish Compound REAdability Metric) (Falkenjack et al., 2017) especially suitable for sentence-level analysis, features were

[7] https://github.com/LDNOOBW

[8] Publicly available at https://huggingface.co/datasets/jumonsen/MSWEP

initially computed for the training dataset. This was done by annotating sentences with linguistic information, such as part-of-speech and dependency relations, using a Swedish `spaCy` (Honnibal et al., 2020) model (`sv_core_news_lg`) and then passing the annotated sentences to SCREAM. In addition to the SCREAM features, normalized character-level Levenshtein similarity was also included in the initial feature set, as it was used by Martin et al. (2020). Thus, 22 features were extracted. All features were computed as the ratio between the target and source sentences. All ratios were capped at a maximum of 2, as done by Martin et al. (2020).

Once ratios had been computed for all features, a selection process began. Highly correlated features (a correlation coefficient above 0.8) were removed. More specifically, if two features had a high correlation, the one with the lowest number of unique values was removed while the other was kept. Generally, it was hypothesized that features with more unique values would work better as control attributes since the variety would induce more nuance in the controlled element and thus provide end users with greater possibilities of adapting the output. Of the remaining features, the four features with the highest number of unique values were selected. Table 1 shows the final features extracted as well as those corresponding to the original features proposed by Martin et al. (2020).

Top-4	Original
avg_word_len	tot_token_len
avg_dep_dist_dep	avg_sent_depth
n_swevoc_tot	lev_sim
tot_token_len	n_swevoc_tot

Table 1: Selected control attributes based on the subset of SCREAM with the addition of character-level Levenshtein similarity (lev_sim). The right column shows the reference control attributes used by Martin et al. (2020) with the n_swevoc_tot feature as a replacement for the English WordRank feature. This lexical measure counts the number of words belonging to different SweVoc (Heimann Mühlenbock and Johansson Kokkinakis, 2012) word lists. avg_word_len is the average length of words in a sentence, tot_token_len the combined length of all tokens in a sentence, avg_dep_dist_dep the average dependency distance between all words within a sentence, and avg_sent_depth is the depth of the dependency tree, given a single sentence.

3.3. Implementation

In total, six models were trained to perform text simplification in Swedish. KBLab's BART model `KBLab/bart-base-swedish-cased` and the multilingual mBART model `facebook/mbart-large-50` were used as base models. The reason for also including the multilingual mBART model was due to the smaller size of the Swedish BART model. These models and their corresponding tokenizers were loaded through the `transformers` library (Wolf et al., 2020) and wrapped in a `LightningModule` with additional data processing and training functionality using the Lightning framework[9].

3.3.1. Prefix Module

The prefix module was built based on the `PeftModelForSeq2SeqLM` class from the `peft` library (Mangrulkar et al., 2022). This class implements functionality for prefix-tuning but differs from the implementation by Clive et al. (2022) in several aspects. First, it uses a single shared prefix for cross-attention and decoder attention and no prefix for self-attention in the encoder. Secondly, it lacks functionality for control prefixes. Therefore, modifications were made so that cross-attention and decoder attention had separate prefix encoders. The self-attention computation in the encoder was ignored since the models seemed to achieve comparable performance to those reported by Clive et al. (2022) without it. Then additional functionality was added to accommodate control prefixes based on conditional information provided with the input data. Prefixes were passed to the attention computations through the `past_key_values` parameter in the forward pass.

3.3.2. Training

The training was conducted using 16-bit automatic mixed precision and AdamW for optimization paired with a linear learning rate scheduler. The BART models were first tuned without control prefixes to provide strong baselines. In these cases, token and positional embeddings were frozen, similar to the work of Clive et al. (2022). Additionally, these models were regularized using a weight decay of 0.1. In the remaining training runs during which control prefixes were utilized, all base model parameters were frozen. During training, each sentence pair was tokenized, padded, and truncated to a length of 128. Examples were then sampled[10] so that the inputs were sorted according to the number of tokens in the source sentence to minimize the amount of padding.

Regarding prefix-tuning with control prefixes, the hyperparameters were chosen mainly based on previous research by Clive et al. (2022). Accordingly, the total prefix length was set to 100. The size

[9] https://lightning.ai/

[10] With the sampler from https://github.com/sarthusarth/SortishSampler.

of control prefixes was set to 1 similar to Clive et al. (2022). Since four control attributes were used, the static prefix had a length of 96. Similar to Martin et al. (2020), the control attribute ratios were discretized into bins of fixed width of 0.05, resulting in 40 bins in total.

All models were trained for 10 epochs using $3,000$ warmup steps and a batch size of 64 across four Tesla V100-SXM2-32GB GPUs. The checkpoint achieving the highest SARI score on the validation set was chosen as the final model. The generation parameters were the same as those used by Clive et al. (2022), i.e. `num_beams=6`, `length_penalty=1`, `min_new_tokens=3`, `max_new_tokens=100` and `no_repeat_ngram_size=3`. The same generation parameters were also used in the evaluation.

3.3.3. Evaluation

All models were evaluated by measuring SARI (Xu et al., 2016) and BLEU (Papineni et al., 2002). These metrics were implemented using `EASSE` (Easier Automatic Sentence Simplification Evaluation) (Alva-Manchego et al., 2019), which improves upon the original version of SARI[11]. The models were evaluated on the PK18 dataset. For the control prefix models, oracle control ratios were used during training. Ratios were then fixed during inference on the test set, similar to Clive et al. (2022). The fixed ratios were set to values to maximize SARI on the validation sets. Nevertheless, in practice, with actual users involved, they can be customized.

4. Results

The evaluation results for the models trained on the MSWEP dataset and evaluated on PK18 are shown in Table 2. For comparison purposes, previous results on PK18 from work by Holmer and Rennes (2023) are also presented, including scores for gold references. These were computed by considering the original sentence as the system output and the gold standard simplified sentence as the reference.

As shown in Table 2, the baseline models demonstrate strong performance, with SARI scores of 31.72 and 33.30, and BLEU scores of 18.65 and 17.47 for kbBART$_{BASE}$ and mBART$_{LARGE}$ respectively. This is similar to the performance of the best model from the work by Holmer and Rennes (2023). Furthermore, the Swedish prefix-tuned kbBART$_{BASE}$ model with the original control attributes significantly outperformed the baseline model with a SARI score of 37.60. mBART$_{LARGE}$ trained with the same control attributes performed slightly worse with a SARI score of 35.43, which was still well above the baseline model.

The prefix-tuned models in which the top-4 attributes from the SCREAM subset were used, yielded similar results. The kbBART$_{BASE}$ model gave a SARI score of 37.65, which was the highest for all models. On the other hand, BLEU was slightly worse than the kbBART$_{BASE}$ model with the other control attributes. The BART$_{LARGE}$ model also performed similarly to the corresponding model with the original attributes. It achieved a SARI score of 35.18, which is marginally lower.

Examples are provided in Table 3 to illustrate the simplifications produced by the models. These were produced using source sentences from the test set and the best performing prefix-tuned kbBART$_{BASE}$ model from Table 2. Control attributes were fixed with the same values as used during testing.

5. Discussion

The prefix-tuned models significantly outperformed the baseline model and ATS models from previous work on the Swedish PK18 test set. The prefix-tuned models achieved SARI scores above 37.60 for kbBART$_{BASE}$ and above 35.18 for mBART$_{LARGE}$. It is also worth noting that BLEU scores were higher for all models compared to previous work, especially for the baseline models. This could be due to the mined paraphrase dataset having more lexically similar sentence pairs than the pseudo-parallel corpus compiled by Holmer and Rennes (2023). Without the control mechanism, the paraphrase data did not seem to improve on simplification, as demonstrated by the fact that the baseline model achieved a lower SARI score of 31.72 or a very similar SARI score of 33.30 to the model trained with the pseudo-parallel. To this, it is worth adding that Holmer and Rennes (2023), who utilized the methodology as proposed by Kajiwara and Komachi (2018), used a significantly smaller amount of sentences initially from the CC-100 corpus (Conneau et al., 2020) (about 60 million resulting in $442,152$ sentence pairs). These differences might be reflected in the results. It is then reasonable that the baseline models without a controllable generation got a high BLEU score since they were not trained to simplify text but rather to paraphrase text. This points to the importance of the controllable generation mechanism for performing simplifications.

The multilingual baseline model performed better than the monolingual Swedish fine-tuned model. However, the prefix-tuned multilingual models performed worse regarding SARI scores than the monolingual models. This was surprising since it has been shown that base model size matters

[11] See https://github.com/feralvam/easse.

Model	Control Attributes	SARI	BLEU
Gold reference (Holmer and Rennes, 2023)	-	22.81	12.80
Best model from Holmer and Rennes (2023)	-	33.24	9.53
kbBART$_{BASE}$ fine-tuned (baseline)	-	31.72	**18.65**
mBART$_{LARGE}$ fine-tuned (baseline)	-	33.30	17.47
kbBART$_{BASE}$ prefix-tuned	Original	37.60	13.48
mBART$_{LARGE}$ prefix-tuned	Original	35.43	10.80
kbBART$_{BASE}$ prefix-tuned	Top-4	**37.65**	11.16
mBART$_{LARGE}$ prefix-tuned	Top-4	35.18	10.88

Table 2: Evaluation results for Swedish models on the PK18 test set.

Source	System output
Example 1	
FUB arbetar enligt tidigare antagen handlingsplan för FUB:s arbete med kognitiv och kommunikativ tillgänglighet med att sprida kunskap och medvetenhet till den egna rörelsen om vad kognitiv och kommunikativ tillgänglighet är.	FUB arbetar med att sprida kunskap om vad kognitiv och kommunikativ tillgänglighet är.
FUB works, according to the previously adopted action plan for FUB's work with cognitive and communicative accessibility, with spreading knowledge and awareness within the organization about what cognitive and communicative accessibility is.	*FUB works with spreading knowledge about what cognitive and communicative accessibility is.*
Example 2	
Socialdepartementet uppmärksammade den framgångssaga som BPSD-registret var för personer med demens och ville därför bygga ett liknande arbetsverktyg anpassat för personer som har insatser genom LSS.	Socialdepartementet har tagit fram ett arbetsverktyg för personer med demens.
The Ministry of Social Affairs drew attention to the success story that the BPSD register was for people with dementia and therefore wanted to build a similar tool adapted for people who have interventions through LSS.	*The Ministry of Social Affairs has developed a tool for people with dementia.*
Example 3	
Kansliet kommer också arbeta vidare på att hitta goda exempel och nya sätt att arbeta på genom att se hur andra organisationer arbetar.	Kansliet jobbar vidare med att hitta nya sätt att arbeta på.
The office will also continue to work on finding good examples and new ways of working by studying how other organizations work.	*The office continues to work on finding new ways of working.*
Example 4	
Resultatet skulle förhoppningsvis vara att vi sku-lle kunna få rådgivning om hur vi ska agera i lokalföreningar och länsförbund.	Vi skulle kunna få hjälp med hur vi ska agera i lokalföreningar och länsförbund.
The result would hopefully be that we could get counseling on how to act in local and county associations.	*We could get help with how to act in local and county associations.*

Table 3: Output examples from the best simplification model. English translations are denoted in italics.

when it comes to prefix-tuning (Lester et al., 2021). Nevertheless, it seemed that the multilingual model had more difficulty adopting the controllable generation mechanism. The reason for this is unclear but could be because the base model is not tuned on Swedish data beforehand, and the prefixes lack the capacity to steer the model towards Swedish to the same extent. An alternative that could have improved the performance of these models would have been to first fine-tune it on Swedish paraphrase data as a warm-up and then do the prefix-tuning with control prefixes.

There were minimal differences regarding the sets of control attributes for Swedish. For kbBART$_{BASE}$, the top-4 SCREAM features yielded a 0.05 increase in SARI compared to the original attributes while mBART$_{LARGE}$ with the original attributes, provided a 0.25 increase in SARI compared to the top-4 selected SCREAM features. It is not easy to draw any conclusions regarding these ob-

servations, but what can be noted is that the selected SCREAM feature seems to provide as good results as the other features.

By looking at output examples produced by the best model, we observed that it could perform various types of simplification transformations. For example, it could perform lexical simplifications and syntactic simplifications such as deletion of clauses, and rearrangement of sentences, to make a text simpler. Although some simplifications were performed impressively, the model was imperfect. In some rare cases, the model produced output that had little to do with the input. Additionally, the model sometimes over-deleted important information, including negations, and struggled with preserving named entities. As observed by Martin et al. (2022) and Holmer and Rennes (2023), pairing sentences with semantic similarity measures sometimes provides sentences that are similar but contain different named entities, such as names, places, dates, and times. It might affect the training and increase the risk of hallucinations of named entities during inference. This could explain the above-mentioned issue.

Regarding the effects of choosing different control ratios, we observed that a change in one attribute ratio did not necessarily affect the output that much. Instead, the combination of lowering all attribute ratios seemed to produce the best simplification. The degree to which all features are involved is somewhat unclear. Training separate models with single control attributes could have been done to distinguish the effects of individual control attributes. Moreover, it was also observed that the attribute controlling length had the most impact generally, consistent with the observation made by Martin et al. (2020). Since the effect of control attribute settings is not evident in all cases, it might confuse real users. It would be desirable to change controls with the guarantee that the output will be adjusted accordingly. Nevertheless, the models showcase that control for Swedish simplification can be achieved using control prefixes to a certain extent.

When it comes to the evaluation of ATS systems, previous research (e.g. Sulem et al., 2018; Alva-Manchego et al., 2020b), has illustrated that automatic measures such as BLEU and SARI have certain flaws when judging the quality of simplifications. This is especially true for BLEU. SARI is, despite its flaws, considered the best option for evaluating ATS automatically. Furthermore, the PK18 test set only has one reference sentence, which might have affected the reliability of the evaluation results to some degree. Human evaluations with different user groups could have provided more reliable estimations of the quality of the produced simplifications.

6. Conclusion

The results presented in this paper highlighted the potential and effectiveness of using paraphrase data in combination with controllable generation mechanisms as opposed to creating parallel or pseudo-parallel corpora. The controllable generation involved prefix-tuning with control prefixes. This approach is especially helpful in languages, such as Swedish, where parallel data is not readily available. The reason why the model learns to perform simplification transformation is most likely because there are aspects of simplification inherent in the paraphrase data. With the controllable generation mechanisms, these are distinguished and can be adapted to simplify sentences.

Another contribution of this paper was the Mined SWEdish Paraphrase (MSWEP) dataset, containing $1,123,909$ paraphrase pairs that were mined from large amounts of text data. This was done using a similar methodology as proposed by Martin et al. (2022). This dataset is made publicly available and could be used for training Swedish ATS models or for other tasks that may benefit from paraphrase data.

The selected features from SCREAM seemed to provide good guidance for controlling the simplification as they positively impacted model performance compared to a fine-tuned baseline model. The models performed well above previously reported results for Swedish. This result highlighted the effectiveness of using paraphrase data paired with controllable generation to carry out the task of ATS. However, the gains were most prevalent with a smaller monolingual model in contrast to a larger multilingual model. Furthermore, the best-performing model showed capabilities in carrying out multiple simplification transformations.

Despite showing promising results, there is still room for improvement in future studies. Exploration of controllable text simplification applied to texts with multiple sentences is one such important direction to consider since most work has been carried out on single sentences. Moreover, exploring methods to ensure that named entities are preserved from the input to the output could be a fruitful direction to pursue to provide higher-quality simplifications. Finally, involving human users is crucial since they are the ones who will end up using tools based on models such as the ones presented in this paper. Therefore, carrying out studies with a human evaluation of ATS systems with controllability is an important future research direction.

7. Ethical Considerations

In general, when training machine learning models on large amounts of text data, there is always

a risk that the model reflects biases pertained in the training data. Acknowledging this is crucial when deploying models for practical use, as users might be affected if exposed to such biases or other harmful model behavior. Pre-trained language models, such as those used in this research, have, for example, been shown to exhibit certain social biases (Liang et al., 2021). A central part of this work was also the creation of the MSWEP, the mined paraphrase dataset. The Common Crawl snapshots contain a large portion of low-quality data, including offensive and inappropriate language. For this reason, careful attention must be paid to ensure that such content is removed. The filtering employed to create the CC-100 corpus (Conneau et al., 2020) and the efforts made to ensure high-quality data likely mitigated this to a large extent. Nevertheless, it is possible that some instances of this kind slipped through the filters and remained in the dataset. The effect this might have had is probably negligent, but essential to be aware of.

Another ethical aspect to consider is the impact of training large language models on the environment, as pointed out by Bender et al. (2021). Although this research used relatively efficient ways of adapting already pre-trained models, it is nevertheless worth considering since the pre-training can have a significant environmental impact. In today's landscape of large language models, BART is also relatively small, with about 400 million parameters for the large version. In contrast, models such as GPT-4 have hundreds of billions or even trillions of parameters requiring pre-training with a significantly larger environmental impact. With this in mind, efforts to develop parameter-efficient and environmentally friendly model adaptation methods are important.

8. Acknowledgements

The first author wishes to express gratitude to Örebro University and the Counterfactual Commonsense Project for their generous support in funding and enabling participation in the conference.

9. Bibliographical References

Fernando Alva-Manchego, Louis Martin, Antoine Bordes, Carolina Scarton, Benoit Sagot, and Lucia Specia. 2020a. 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.

Fernando Alva-Manchego, Carolina Scarton, and Lucia Specia. 2020b. Data-driven sentence simplification: Survey and benchmark. *Computational Linguistics*, 46(1):135–187.

Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, page 610–623, New York, NY, USA. Association for Computing Machinery.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit*. "O'Reilly Media, Inc.".

Jordan Clive, Kris Cao, and Marek Rei. 2022. Control prefixes for parameter-efficient text generation. In *Proceedings of the 2nd Workshop on Natural Language Generation, Evaluation, and Metrics (GEM)*, pages 363–382, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Johan Falkenjack, Evelina Rennes, Daniel Fahlborg, Vida Johansson, and Arne Jönsson. 2017. Services for text simplification and analysis. In *Proceedings of the 21st Nordic Conference on Computational Linguistics*, pages 309–313, Gothenburg, Sweden. Association for Computational Linguistics.

Kenneth Heafield. 2011. KenLM: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 187–197, Edinburgh, Scotland. Association for Computational Linguistics.

Katarina Heimann Mühlenbock and Sofie Johansson Kokkinakis. 2012. Swevoc - a swedish vocabulary resource for call. In *Proceedings of the Swedish Language Technology Conference 2012 workshop on NLP for CALL*, pages 28 – 34.

Daniel Holmer and Evelina Rennes. 2023. Constructing pseudo-parallel Swedish sentence corpora for automatic text simplification. In *Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 113–123, Tórshavn, Faroe Islands. University of Tartu Library.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength natural language processing in python.

Tim Isbister and Magnus Sahlgren. 2020. Why not simply translate? a first Swedish evaluation benchmark for semantic similarity.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2017. Billion-scale similarity search with gpus.

Tomoyuki Kajiwara and Mamoru Komachi. 2018. Text simplification without simplified corpora. *Journal of Natural Language Processing*, 25(2):223–249.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

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

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. 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 4582–4597, Online. Association for Computational Linguistics.

Paul Pu Liang, Chiyu Wu, Louis-Philippe Morency, and Ruslan Salakhutdinov. 2021. Towards understanding and mitigating social biases in language models.

Maja Lindberg and Erik Kindberg. 2018. Proffskorpus - korpusdokumentation. Internal report.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing.

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

Jonathan Mallinson and Mirella Lapata. 2019. Controllable sentence simplification: Employing syntactic and lexical constraints.

Martin Malmsten, Love Börjeson, and Chris Haffenden. 2020. Playing with words at the national library of sweden – making a swedish bert.

Sourab Mangrulkar, Sylvain Gugger, Lysandre Debut, Younes Belkada, and Sayak Paul. 2022. Peft: State-of-the-art parameter-efficient fine-tuning methods.

Louis Martin, Éric de la Clergerie, Benoit Sagot, and Antoine Bordes. 2020. Controllable sentence simplification. In *Proceedings of the Twelfth 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 Benoit Sagot. 2022. Muss: Multilingual unsupervised sentence simplification by mining paraphrases. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 1651–1664, Marseille, France. European Language Resources Association.

Daiki Nishihara, Tomoyuki Kajiwara, and Yuki Arase. 2019. Controllable text simplification with lexical constraint loss. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 260–266, Florence, Italy. Association for Computational Linguistics.

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

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Faton Rekathati. 2021. The kblab blog: Introducing a swedish sentence transformer.

Evelina Rennes. 2022. *Automatic Adaptation of Swedish Text for Increased Inclusion*. Ph.D. thesis, Linköping University.

Sebastian Ruder, Ivan Vulić, and Anders Søgaard. 2022. Square one bias in NLP: Towards a multi-dimensional exploration of the research manifold. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2340–2354, Dublin, Ireland. Association for Computational Linguistics.

Stian Rødven Eide, Nina Tahmasebi, and Lars Borin. 2016. The swedish culturomics gigaword corpus: A one billion word swedish reference dataset for nlp. In *From Digitization to Knowledge 2016: Resources and Methods for Semantic Processing of Digital Works/Texts*.

Horacio Saggion. 2017. *Automatic Text Simplification*. Synthesis Lectures on Human Language Technologies. Springer Cham.

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

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

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

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Guillaume Wenzek, Marie-Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco Guzmán, Armand Joulin, and Edouard Grave. 2020. CCNet: Extracting high quality monolingual datasets from web crawl data. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4003–4012, Marseille, France. European Language Resources Association.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

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.