Computer Science and Information Systems 21(3):899–921 https://doi.org/10.2298/CSIS230912017M

Reaching Quality and Efficiency with a Parameter-Efficient Controllable Sentence Simplification Approach

Antonio Menta and Ana Garcia-Serrano

E.T.S.I. Informática (UNED)
C. de Juan del Rosal, 14, 28040 Madrid, Spain
amenta1@alumno.uned.es
agarcia@lsi.uned.es

Abstract. The task of Automatic Text Simplification (ATS) aims to transform texts to improve their readability and comprehensibility. Current solutions are based on Large Language Models (LLM). These models have high performance but require powerful computing resources and large amounts of data to be fine-tuned when working in specific and technical domains. This prevents most researchers from adapting the models to their area of study. The main contributions of this research are as follows: (1) proposing an accurate solution when powerful resources are not available, using the transfer learning capabilities across different domains with a set of linguistic features using a reduced size pre-trained language model (T5-small) and making it accessible to a broader range of researchers and individuals; (2) the evaluation of our model on two well-known datasets, Turkcorpus and ASSET, and the analysis of the influence of control tokens on the SimpleText corpus, focusing on the domains of Computer Science and Medicine. Finally, a detailed discussion comparing our approach with state-of-the-art models for sentence simplification is included.

Keywords: Text Simplification, Transfer Learning, Language Models.

1. Introduction

The textual information that we consume daily is often presented in a style and vocabulary that varies significantly depending on the source and the particular domain area. For example, the language used on social media platforms is very different from the language used in newspapers, academic journals, or health insurance policies. According to numerous studies conducted by the Organisation for Economic Co-operation and Development (OECD), it has been found that between 15-30% of the adult population can only comprehend basic vocabulary [58]. As a result, a considerable number of adults struggle to understand the content presented in various news articles and books.

This presents a major challenge, as every individual, regardless of their literacy level, has the right to access and comprehend information that is pertinent to their lives. This is particularly important when it comes to information that has a direct impact on their daily activities, decision-making processes, and overall well-being. Furthermore, research conducted across various countries has consistently shown that adults who possess lower levels of literacy are more prone to experiencing health-related issues, less likely to engage in political processes, and generally exhibit lower levels of trust toward others.

Areas such as computer science or medicine, present additional difficulties due to a large number of neologisms and technical words [10]. Several workshops [18,19] and shared tasks [20] have been set up to deepen access to this type of technical documentation. Similarly, tools such as MedSimples [73] can evaluate text and simplify suggestions for a global audience.

Over the last twenty years, several initiatives have been launched to produce easy-to-read versions of texts, with a particular focus on the inclusion of socially disadvantaged groups in society. Unfortunately, the manual production of basic versions is an expensive and time-consuming process. In addition, the content is often outdated by the time it is published simplified version. The aim is to disseminate universally accessible versions of documents on social issues [59] to improve texts for groups with some disabilities such as aphasia [11], autism [21], or dyslexia [51].

In early work, ATS was used as a pre-processing stage for other NLP tasks such as summarization [35], parsing [12], question generation [64] or clarification [74], and more recently, as an improvement of machine translation systems for languages with limited resources [62]. Due to its relationship with machine translation, a natural step was to adapt the same techniques to generate simplified versions as a monolingual translation task [67]. Over time, the task has been divided into different categories: lexical simplification, syntactic simplification, sentence simplification and a hybrid category [3]. Lexical simplification (LS) is based on identifying the most complicated words and replacing them with simpler words [45,48], avoiding unnecessary words or information and cross references and other. Syntactic simplification attempts to reduce grammatical complexity by identifying the most complex syntactic structures and eliminating or replacing them, as for example avoiding the use of relative clauses, which introduce subordinate structures that may be difficult to understand for certain groups of people [53,54]. Sentence Simplification involves operations like sentence splitting, removal of non-essential sentence components, lexical simplification, and more.

The first automatic models were rule-based, focusing on syntactic simplification in English [12]. Subsequently, data-driven approaches were used with supervised systems trained on parallel corpora [7]. Since 2014, neural networks have been widely used for this task. Initially, using word embeddings with systems based on lexical simplification [25] and later, using sequence-sequence architectures for neural machine translation (NMT) [44,47]. In recent years, efforts have shifted towards the use of deep neural networks with large language models for sentence-level simplification systems [58,63].

Within this approach, controllable sentence simplification is a versatile method that lets you fine-tune the output of a simplification model according to certain characteristics. In this work, we continue the research line started by [37] and [57] searching for the optimal feature values for sentence simplification to reach a suitable performance when powerful resources are not available. We select well-known attributes/features and define a new control feature to help the model to produce better lexical simplifications using a controllable sentence simplification approach.

The paper is structured as follows. It begins with a section dedicated to previous work on the task, followed by a description of the controllable sentence simplification approach using neural networks in combination with linguistic attributes. This is followed by a description of the framework developed, the experiments carried out with it, and the datasets used for training and evaluation. Then, a comparison of the results with the state of the art in the task and an analysis of the relevance of each of the features are presented to detail the contribution of the paper. Finally, some conclusions and future work are included.

2. Related Works

Early work on lexical simplification was developed for the English language using word frequencies in dictionaries and word semantic resources such as Wordnet to find simple synonyms for particular words. This approach was replaced by corpus-based methods. One example is Wikipedia. [26] used a dataset of aligned sentence pairs between English Wikipedia and Simple English Wikipedia to identify word substitutions. More recently, approaches based on BERT [15] propose substitution candidates for complex words in a sentence with high semantic similarity. For other languages, such as Spanish, several initiatives have been launched in recent years. For example, the EASIER system [2] evaluates lexical simplification for people with cognitive impairments.

Similar to the work done on lexical simplification using Wikipedia, sentence-level approaches have exploited the creation of parallel corpora between sentences extracted from Wikipedia and Simple English Wikipedia. The success of neural machine translation and the emergence of several large parallel datasets have enabled the use of deep neural network models. Early approaches were based on reinforcement learning [69] or the use of an augmented memory function known as a neural semantic encoder [65].

One of the disadvantages of the neural machine translation encoder-decoder models is that the original text and the simplified version often share many terms. Therefore, a few changes need to be made for simplification. This, combined with the fact that these operations must be learned implicitly, means that a large amount of data is required to learn the rules to be applied. In many cases, this leads to repetition of the original text because the learned patterns are not good enough [70].

Edit-based ATS systems overcome the previous limitation by learning the necessary transformations to convert complex sentences into their simpler counterparts. To do so, they use a predefined set of explicit editing operations (ADD, DELETE, MOVE, REWRITE) to transform only the complex terms during the generation of the simple sentence. Thus, the task can be approached as a sequence tagging problem using a BiLSTM-based architecture [4] or by proposing a programmer-interpreter neural network, inspired by the way humans iteratively simplify a text [16].

Subsequently, [14] added a convolutional graph module with the syntactic information of sentences to aid the detection of complex structures in the simplification process. These models are more interpretable than usual because the words to be replaced and the operation performed on them are known. Also, the search vector space of the models is small, so they are not as dependent on a large dataset as those based on linguistic models.

Figure 1 shows an example of automatic tagging based on word alignment between an original sentence (top) and a simplified sentence (bottom) in the Wikilarge dataset. Words not aligned on the original side receive the label ”D” (DELETE), while words aligned with a different shape receive ”R” (REPLACE), and added words receive an “A” (ADD). Words aligned without an explicit label are given the label “C” (COPY).

Currently, deep learning-based language models have proven to be state-of-the-art in many NLP tasks. Their effectiveness comes from prior training on a self-supervised task, such as a language model or a masked token prediction process. Once the model is

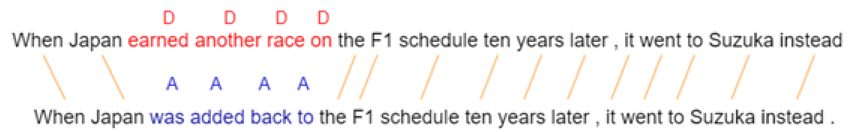

Fig. 1. Example of automatic labelling based on word alignment between an original sentence (top) and a simplified sentence (bottom). Extracted from Wikilarge dataset [69]

trained, its use in other tasks requires a much smaller data set than that needed to train a model from scratch. In particular, pre-trained Bidirectional Encoder Representations from Transformers (BERT) [15] and transfer learning techniques since ULMFit [27], have revolutionized sentence simplification task approaches.

Within this type of model, those based on an encoder-decoder architecture have excelled in tasks such as text summarisation [71] or machine translation. These include BART [34] and T5 [49]. For example, in simpleT5 [42], the authors propose the use of T5 and the union of different datasets to generate sentence simplifications. Another possibility is to include external information in the parallel corpus itself to simplify the model. An example is the BART model and the keywords provided together with the dataset [52]. In short, several works have demonstrated the transfer capabilities of pre-trained language models in the resolution of the sentence simplification task [41,61].

There are also approaches focusing on multiple languages using mT5 [68], a version trained on a massive dataset with more than a hundred languages. One example is the use of the model to simplify Arabic sentences [28]. Other work has used an unsupervised approach for each language using web mining techniques to obtain a dataset based on paraphrasing between sentences [37]. They also propose the addition of external features to the language model to control the simplification generated. This is called "conditional training". This process refers to the process of training a machine learning model to simplify sentences based on specific conditions or constraints. These conditions can include factors such as the intended audience, the level of simplification required, and the type of content being simplified.

For example, a model trained to simplify medical texts may have different conditions and constraints than a model trained with news articles. By incorporating these conditions and constraints into the training process, the model can learn to generate simplified sentences that are more appropriate for the intended use case. This can help improve the accuracy and effectiveness of sentence simplification systems, particularly in specialized domains where specific terminology and language conventions are used.

Conditional training has been applied in various NLP tasks to control text generation using sequence-to-sequence models. These tasks include text summarization [30], sentence compression [23], and text generation with specific linguistic styles [24]. The control of text generation can be achieved in the decoding phase by adjusting the output based on a specific attribute [30], or by modifying the training process through the addition of new features to the input vector, such as vector length [36]. [55] add the value of certain control features to the text to select the appropriate level among grade-level learners. [39] use grammatical attributes as model features to study their impact on simplification. Subsequently, MUSS [37], extend their work by using a massive dataset based on paraphrasing rather than direct simplifications. In [57], they propose to use T5 as a language model to take advantage of its transfer learning capabilities. This model was pre-trained on different NLP tasks (e.g., text summarisation, question answering, machine translation) using a massive dataset such as C4[1].

In the past year, studies have begun to use Large Language Models for text simplification tasks [22]. However, the computational demands of fine-tuning these models for technical domains such as medicine have prevented many researchers from using them. The use of smaller models, but adapted to these complex domains by using a specific dataset, is therefore recommended.

3. Controllable Sentence Simplification

A controllable sentence simplification is a flexible approach that allows the output of a simplification model to be fine-tuned based on specific attributes. These attributes can be defined and customized for different use cases, such as adjusting the level of simplification for different audiences or content types. For example, a simplification model can be configured to simplify text for readers with cognitive disabilities, with output conditioned on text length and lexical complexity. Alternatively, a model can be trained to simplify news articles for readers with limited English proficiency, where output is conditioned on the amount of paraphrasing in the sentence to ensure that meaning is preserved. By incorporating controllable features into the simplification process, models can be tailored to meet specific needs and improve the effectiveness of the simplified output.

The first step in investigating the potential of language models and controllable sentence simplification in this work is to evaluate their transfer learning capabilities across different domains. The evaluation process starts with the general domain, followed by more technical vocabulary such as computer science and medical articles. The language model selected was T5, which has been pre-trained on summarisation and machine translation tasks, making it a good starting point for adding simplification features to the model. This, together with the use of a large dataset from the web, such as C4, makes it suitable for use in different domains. The T5 [49] model was also chosen because the input and output texts are in plain text format, which facilitates the addition of lexical, grammatical, and structural features without the need to modify the model's architecture. In our work, we adopt a simplified approach that involves conditioning the generation process by concatenating plain text control attributes to the source text.

Regarding T5, there are currently 5 pre-trained model versions. Depending on the number of neurons, they vary from T5-small with 60 million to T5-11B with 11 billion. In general, the greater the number of neurons the more information can be stored in the model. One of the main goals of our work is to study text simplification with language models with limited resources for use by as many researchers as possible. For this reason, we have selected the smaller model, T5-small, which can be used on most available hardware.

[1] https://www.tensorflow.org/datasets/catalog/c4

3.1. Control Attributes

Different linguistic attributes related to a simplification process are defined by [39], grouping them according to their typology:

- Text compression: Text compression is a process that involves shortening the length of sentences in simplification-oriented corpora. Typically, the length of the simplified sentences is shorter than that of the original sentences, and this compression of information results in a reduction in sentence complexity [38]. Text compression can be achieved through various techniques such as lexical substitution, deletion of non-essential words, and syntactic restructuring.
- Paraphrasing: The level of paraphrasing used in the simplification process is another attribute to highlight. If the level of paraphrasing is low, the meaning of the simplified version will be very similar to the original, and there may be no real simplification of the text. On the other hand, if the level of paraphrasing is high, there is a risk that the information in the sentence may be entirely changed, resulting in a loss of accuracy and relevance. Therefore, it is essential to select a balance between the level of paraphrasing and the level of simplification to ensure that the simplified text is both accurate and understandable.
- Lexical complexity: Lexical complexity is an attribute that indicates the level of complexity of the words used in a sentence. This attribute is based on the task of lexical simplification, which involves replacing complex words with simpler synonyms or alternative words. In technical fields with many specialized terms, substituting these terms with simple synonyms can significantly reduce the complexity of the sentence.
- Syntactic complexity: Based on the syntactic simplification task, syntactic complexity is an attribute that measures the complexity of the sentence structure in the context of syntactic simplification. The difficulty of reading a sentence can vary depending on various factors, such as the verb tense used, the use of the passive voice, or the depth of the sentence dependency tree. For example, using simpler verb tenses such as the present tense can make the sentence structure less complex and more accessible.

In this work, the above conceptual attributes are reflected in several features, the so-called control tokens, to modify the features of each sentence. The control tokens used in this work are character count, word count, Levenshtein-based similarity, dependency tree depth, word rank, and language model Fill-Mask.

Character count (CLR): This feature reflects text compression. That is, it uses the number of characters that each sentence is made of. Previous research has shown that sentence length can be a good indicator of complexity, with longer sentences generally being more complex than shorter ones [38].

Word count (WLR): [39] utilizes a feature based on the hypothesis that sentence simplicity is positively correlated with word frequency. The authors also suggest that word ratio is another important factor to consider in this context. While word frequency is typically associated with familiarity, the use of longer words can present difficulties for readers. Additionally, previous studies have found that simplified sentences tend to incorporate shorter and more frequently used words in comparison to their original versions [17].

Levenshtein-based similarity (LR): This feature is a measure of the changes made between the source and the simplification to determine their degree of paraphrase. It is calculated using Levenshtein's normalized similarity [33]. This is a measure of the similarity between two strings or sequences of characters. It is based on the Levenshtein distance, which is the minimum number of single-character edits (insertions, deletions, or substitutions) required to transform one string into the other.

Dependency Tree Depth (DTDR): This feature is used to add syntactic information to the model. A dependency tree is a hierarchical representation of the grammatical structure of a sentence, where the nodes represent words or groups of words, and the edges represent the syntactic relationships between them. The depth of a dependency tree is an important metric in NLP as it can be used to measure the complexity of a sentence. The deeper the dependency tree, the more complex the sentence. It is calculated by dividing the maximum dependency tree depth value of the simplified sentence by the original sentence [39].

Word Rank (WRR): In the context of lexical simplification, the word rank refers to the frequency with which a particular word appears in a given language or text. Words that have a higher rank are considered to be more common and familiar to readers, while words with a lower rank may be less familiar and more difficult for readers to understand. By considering word rank, lexical simplification tools can help writers identify and replace complex or uncommon words with simpler, more familiar alternatives to improve the overall readability of their text. Word frequency is one of the best indicators of word complexity [46].

Language Model Fill-Mask (LMFMR): This feature is based on masked language modelling. This is a type of language modeling in which certain words or tokens are masked or hidden, and the task is to predict the masked words based on the context of the surrounding words, as can be found in Table 1. We hypothesize that a language model trained on a masking task will be able to predict the set of masked words in a simple sentence earlier than in a complex sentence. In a simple sentence, the words used will be common, so the model will have been trained with them more often than with some complex and less-used words.

Table 1. Illustrates the masking process using COVID-SciBERT as a model and the SimpleText dataset [20], highlighting the dependence of the prediction model on the dataset's domain. This dataset was created to focus on both technological and medical domains simultaneously

Model	Text
Before Masking	Based on the inception - v3 architecture, our system performs better in terms of processing complexity and accuracy than many existing models for imitation learning.
COVID-SciBERT	Based on the [MASK] - v3 architecture, our system performs better in terms of processing complexity and accuracy than many existing models for imitation learning.

For every original and simplified sentence pair, features are computed by dividing the value of the simplified version by the original. This results in a value for each pair, which is then added as a prefix to the original text and displayed in Table 2 with its corresponding feature identifier. During the prediction stage, the feature values can be adjusted to cater to specific audiences, allowing for versatile feature combinations without modifying the model's structure. This input format streamlines the process by enabling pre-calculation and storage of values, which can be added as training progresses.

Table 2. Modified example with calculated features

Model	Text
Original	In the modern era of automation and robotics, autonomous vehicles are currently the focus of academic and industrial research.
Modified	simplify: CLR_0.65 DTDR_0.75 LMFMR_0.65 LR_0.65 WLR_0.8 In the modern era of automation and robotics, autonomous vehicles are currently the focus of academic and industrial research.

In our experiments, the feature values are categorized into fixed-width bins of 0.05 and are limited to a maximum ratio of 1.5. Subsequently, these features are evaluated across two datasets using different models. Selected values for the TurkCorpus dataset [67] included WLR_1.25, CLR_0.85, LR_0.65, DTDR_0.9, WRR_0.6, and LMFR_1.15. For the ASSET dataset [5], we utilized all control tokens, WLR_1.25, CLR_0.8, LR_0.65, DTDR_0.8, WRR_0.65, and LMFR_1.05.

3.2. Developed Framework for Experimentation

Language models are essential tools in natural language processing and the development of these models requires a substantial amount of data, as well as careful selection and tuning of various hyperparameters. This is because the performance of the model depends on the quality and quantity of the training data, as well as the choice of features and the model's architecture. These models can be fine-tuned on specific tasks with a smaller amount of data, enabling researchers to develop state-of-the-art models with limited resources.

The training dataset is created by incorporating selected feature values that capture the essential aspects of the sentence, such as its syntactic structure and lexical complexity. The feature values are fine-tuned using the validation dataset, which is different from the training set. The feature values vary across datasets due to distinct creation criteria, which reflect the underlying differences in the data distribution and the task requirements. The test dataset is used to evaluate the model's performance and compare it with prior work. This is an essential step, as it provides a benchmark for the model's performance and enables researchers to identify areas for improvement. The test set is typically held out from the training and validation sets and should be representative of the data distribution to ensure reliable evaluation.

We have created a Python framework for testing different models and datasets. The code is available on our GitHub repository[2] . The advantage of using this framework is that it facilitates the reproducibility of the experiments performed and allows the comparison between the results obtained with different hyperparameters. The model was trained using Pytorch-lightning and HuggingFace [62] on an Nvidia GeForce GTX 1070 Ti GPU with 8 GB of memory.

4. Benchmarking

Selecting the correct pre-trained language model is crucial for achieving accurate and effective sentence simplifications. From all available T5 pre-trained models in the HuggingFace community, we have trained with the smallest, T5-small (60 million parameters) to find a suitable solution when no powerful resources are available. Other pre-trained described in the original paper [49] are: T5-Base (220 million parameters), T5-Large (770 million parameters), T5-3B (3 billion parameters), and T5-11B (11 billion parameters).

All tests have been performed with the same hyperparameters such as 5 epochs (total number of iterations of all the training data), a batch size (the number of training examples utilized in one iteration) of 6 for both training and validation, a maximum number of 256 tokens, a learning rate (the rate at which an algorithm updates or learns the values of the parameters) of 3e-4 and a weight_decay (regularization technique applied to the weights of the neural network) of 0.1. A beam search size (decision-making layer to choose the best output given target variables) of 8 and a seed value of 22 have also been used.

These hyperparameter values are the result of an initial experimental exploration with the selected model and the training dataset. After the model was trained on the training dataset, we took advantage of the advanced capabilities of Optuna [1], an automatic hyperparameter optimization software framework, specifically designed for machine learning.

With Optuna, we conducted a meticulous hyperparameter search, performing 500 experiments to ensure the best possible results for our work. To ensure accuracy, we limited the maximum and minimum values of each control feature to 1.5 and 0.5, respectively, while using an incremental search value of 0.05. Finally, the ultimate results were obtained using the test dataset, ensuring that our model was both well-trained and well-optimized for maximum performance.

4.1. Training Dataset

Most available datasets are small, limiting their use with deep neural networks. Therefore, we chose the Wikilarge dataset [69]. This is the largest and most widely used dataset for text simplification. It consists of 296,402 automatically aligned sentence pairs extracted from the English version of Wikipedia and Wikipedia Simple. The dataset is not a parallel corpus, but a comparable corpus, because the articles in Wikipedia Simple could have been written independently of those with the same title in Wikipedia [59].

Several papers have questioned the quality of the Wikilarge dataset [8,60,66]. We use Sentence-BERT [50] to evaluate its quality. This model is a modification of the BERT architecture that uses siamese and triplet networks [56] to obtain a sentence-level embedding with a semantic representation of the sentence, that is, a transformation of the

[2] `https://github.com/Hisarlik/Simplification_experiments`

sentence into a vector of numbers, where sentences with similar meanings have similar numerical values [29]. This representation allows comparison with other sentences using cosine similarity. The model can be used as part of the Sentence Transformers toolkit[3] which enables the creation of high-quality, semantically meaningful sentence embeddings. The toolkit employs transformer-based neural networks to encode sentences as dense vectors, which can then be used for a variety of NLP tasks.

Other semantic similarity representations can be used to measure the similarity between words based on their meaning. Latent semantic analysis or graph-based models are some of the popular representations used for this purpose. These representations employ different techniques to capture the semantic relationships between concepts. Ontology-based similarity uses a structured knowledge base to measure the similarity formally and explicitly, which makes it useful for reasoning and inference. This measure has been explored in domains such as medicine. For example, the HESML library [32] uses an ontology-based semantic similarity measure to match related words.

In our work, we aim to improve the semantic quality of the training data used for automatic text simplification. To achieve this, we decided to select several threshold values between 0 and 1, which would allow us to generate different training datasets. For each pair of original and simplified text, we calculated the semantic difference and removed pairs with a similarity value lower than the threshold.

Table 3 summarizes the number of training documents with a similarity value higher than each of the selected thresholds. By finding the ideal threshold value, we were able to ensure that the original and simplified texts were semantically related, resulting in a reduction of the training data, but with higher semantic quality. Table 4 provides some examples of pairs that were eliminated during this process. Overall, our approach allowed us to improve the effectiveness and accuracy of automatic text simplification by enhancing the quality of the training data used.

Table 3. Pair sentence similarity using Sentence-BERT

Similarity	Pair sentences total
TOTAL	296.402
≥ 0.3	257.720
≥ 0.5	228.612
≥ 0.7	187.890
≥ 0.85	131.816
≥ 0.9	103.321

Next, we compare results obtained by selecting different similarity values between pairs of sentences in the training set. Deep learning-based models, studied in the previous section, typically perform better on larger datasets. However, we also want to test the quality of Wikilarge's automatic simplifications. For this purpose, we trained different models using each of the selected thresholds: 0.3, 0.5, 0.7, 0.85, and 0.9 similarities. We evaluated the models on the TurkCorpus dataset, using the best hyperparameters found during the validation stage and obtaining results using the test data (Table 5).

[3] https://www.sbert.net/

Table 4. Similarity ≤ 0.3 Sample deleted from the original corpus

Model	Text
Original	He is credited with many recording innovations.
Simplified	He was born in Waukesha, Wisconsin, and died in White Plains, New York.
Similarity Score	0.242623
Original	Content-control software determines what content will be available.
Simplified	In the USA, there is a filter used with TV called the V-chip.
Similarity	0.133548

Table 5. Pair sentence similarity threshold results on the TurkCorpus dataset

Similarity threshold	SARI	FKGL
ALL	42.58	5.38
≥ 0.3	42.65	5.74
≥ 0.5	42.89	5.77
≥ 0.7	42.92	6.37
≥ 0.85	43.25	6.38
≥ 0.9	43.10	6.27

Turkcorpus is a dataset in which each sentence has 8 simplifications created by Amazon Mechanical Turk workers. Each labeller was asked to paraphrase the original sentence, preserving as much meaning as possible. It contains 2000 sentence pairs for validation and 359 for testing. We also used ASSET, which uses the same data to crowdsource simplifications with a series of rewrite transformations (10 simplifications per original sentence).

In addition, to analyze the output of the models we have used a newly created dataset for the SimpleText lab in CLEF 2022 [20]. The lab focused on the challenges encountered for scientific text simplification in two domains as Computer Science and Medicine. For the former, 13 topics were selected from the Guardian newspaper headlines on the topic. For the second, the texts came from medical articles retrieved from Google Scholar and Pubmed. The texts were simplified by a master's student in translation or by an expert, either a subject matter expert or a professional translator. Each example was rewritten several times until it was clear to a non-expert. The dataset consists of 648 original-simplified sentence pairs for the validation and 116,763 for the test without the simplified version.

The results reported in the following subsections have been obtained by training the model with Wikilarge with the documents that have a similarity greater than 0.85 as explained in the training data section. From this model, we have selected the feature values in the validation dataset and the result using the TurkCorpus test set. The model used the following features: WLR_1.25, CLR_0.85, LR_0.65, DTDR_0.9, WRR_0.6, and LMFR_1.15.

For the ASSET dataset, all control tokens are used: WLR_1.25, CLR_0.8, LR_0.65, DTDR_0.8, WRR_0.65, and LMFR_1.05. Overall, by utilizing these datasets and selecting the optimal feature values, we were able to test and validate the effectiveness of our approach to automatic text simplification. These results could help improve the accessibility and readability of texts for a wider public.

4.2. Evaluation Metrics

In the field of automatic text simplification, it is essential to have reliable metrics for evaluating the effectiveness and accuracy of different models and approaches. Following [5,55,70], SARI is chosen as the main metric to evaluate the final results [67]. SARI stands for System Output Against Reference and Input, and it is a widely used automatic metric for evaluating sentence simplification.

SARI works by comparing the system output with the ground truth and the input sentence. It averages the F1 scores of additions, keeping, and deletions operations. The addition score measures the ability of the system to add new information to the output sentence while preserving the meaning of the input. The keeping score measures the ability of the system to retain the information from the input sentence in the output. Finally, the deletion score measures the ability of the system to remove redundant or unnecessary information from the input sentence.

In addition to using SARI as the primary metric for evaluating the effectiveness and accuracy of our models, we also incorporated the FKGL metric [31] in our study. FKGL, or Flesch-Kincaid Grade Level, is a widely used readability metric that measures the complexity of text based on the number of words per sentence and the number of syllables per word. Unlike SARI, which measures the semantic quality of the simplifications created, FKGL only reflects the simplicity of the text and does not consider other attributes such as the preservation of meaning or grammatical features. It is calculated using the following formula:

$$FKGL = 0.39\frac{TotalWords}{TotalSentences} + 11.8\frac{TotalSyllables}{TotalWords} - 15.59 \tag{1}$$

FKGL score is expressed as a grade level, which indicates the level of education required to understand the text. In our work, we utilized the Python EASSE library [6] to calculate the SARI and FKGL values for our models. The EASSE library is specifically designed for simplification metrics and provides a range of functions for evaluating the effectiveness and accuracy of different models and approaches to automatic text simplification.

In the succeeding section, we present the results of other approaches from their respective papers since reproducing them on our hardware was challenging. Nonetheless, we were able to replicate the results of the t5-small model [57].

4.3. Discussion

Table 6 and Table 7 present a comparative analysis of our best model against other state-of-the-art models for sentence simplification using the TurkCorpus and ASSET datasets, respectively. The models selected for comparison are:

- DMASS-DCSS [72]: This model is based on a sequence-sequence architecture with an attention mechanism. The model combines deep learning and rule-based methods. The authors propose a two-stage model that first employs a transformer architecture (DMASS) to generate multiple simplification candidates and then ranks them using a rule-based system (DCSS) to select the best paraphrase [74].

– ACCESS [39]: This sequence-sequence model incorporates an explicit attribute control mechanism for simplification. The attributes used in this model reflect text compression, paraphrasing, and lexical, and syntactic complexity.
– MUSS [37]: This BART-based model uses the attributes of the ACCESS model to perform sentence simplification. The model was trained using sentence-level paraphrase data rather than traditional simplification data. It also uses unsupervised pre-training and adjustable generation mechanisms, allowing flexible control over attributes such as length and lexical complexity during the inference process.
– Control prefixes [13]: This BART-based model employs a technique known as prefix-tuning to incorporate the representation of attributes in different layers of the model. Prefix-tuning is a powerful lightweight technique for adapting a large pre-trained language model to a downstream application. These attributes guide the generation of text toward a particular task, in this case, text simplification.
– T5-base [57]: In this paper, the authors explore the use of T5 fine-tuning for text simplification, combined with a controllable mechanism to regulate system output, which can help generate adapted text for different audiences. It includes a new feature, #words, to assist the model in replacing long words with shorter ones.
– GPT3.5 and chatGPT [22]: This approach uses OpenAI's GPT3.5 (text-davinci-003) and ChatGPT with different prompts to guide the model to a particular desired output. The model demonstrates the zero-shot transfer capabilities of LLMs. The author also uses a one-shot method with multiple combinations of examples on the model.

Table 6. Comparison of state-of-the-art models with the TurkCorpus dataset

Model	Type	Data	SARI	FKGL
DMASS-DCSS	Transformers	WikiLarge	40.45	8.04
GPT3.5 / ChatGPT	LLM	Single Shot Example	41.82	6.97
ACCESS	Transformers	WikiLarge	41.87	7.22
Control prefixes	BART	WikiLarge	42.32	7.73
MUSS – Supervised	BART	Wikilarge	42.62	6.98
T5-small (Sheang & Saggion, 2021)	T5-small	Wikilarge	40.83	6.78
T5-base (Sheang & Saggion, 2021)	T5-base	Wikilarge	**43.31**	6.17
T5-small (This work)	T5-small	Wikilarge	43.25	6.38

Upon comparing our methodology to other studies utilizing TurkCorpus, our model demonstrated nearly identical performance to the current state-of-the-art, achieving a SARI score of 43.25. This score was marginally lower than the T5-base model's 43.31 [57]. The unexpectedly low score of the LLM-based model in TurkCorpus, compared to the score it obtained in another evaluation dataset (ASSET), is noteworthy. The authors indicates that LLMs sentence simplification methods possess a penchant for excising segments of intricate sentences.

Our approach achieved promising results when evaluated on the ASSET dataset. Specifically, we obtained a SARI value of 44.56 by employing all control tokens. This result was

Table 7. Comparison of state-of-the-art models with the ASSET dataset

Model	Type	Data	SARI	FKGL
ACCESS	Transformers	WikiLarge	40.13	7.29
Control prefixes	BART	WikiLarge	43.58	5.97
MUSS – Supervised	BART	WikiLarge	44.15	6.05
T5-small (Sheang & Saggion, 2021)	T5-small	WikiLarge	39.12	6.99
T5-base (Sheang & Saggion, 2021)	T5-base	Wikilarge	45.04	5.88
GPT3.5 / ChatGPT	LLM	Single Shot Example	**47.94**	7.00
T5-small (This work)	T5-small	Wikilarge	44.56	5.76

the third-best among all models evaluated. The result of the chatGPT-based model is high compared to the other approaches, 47.94. Notably, our approach is the smallest of the models evaluated, containing almost four times fewer parameters than the T5 base. Despite this, our approach still achieves very competitive results, suggesting that it is very resource efficient. Our language model utilizes a smaller version of T5, with only 60 million parameters, compared to T5-base, which has 220 million parameters. This demonstrates the effectiveness of our approach in reducing the number of parameters required to achieve optimal results.

Our approach achieves this by eliminating the most semantically distant sentence pairs from WikiLarge and incorporating the LMFMR feature into the model. The same applies to the other work based on BART: the smallest BART model has 140 million parameters, and the largest model goes up to 400 million parameters. So, we get better results with less than half as many parameters as the BART base version.

5. Parameter-Efficient Controllable Sentence Simplification Results

Parameter-efficient controllable sentence simplification plays an important role in improving the readability and comprehension of complex texts for the wider public. By using a reduced number of parameters, these models aim to achieve meaningful and controllable text simplification without compromising overall performance. This results in a more resource-efficient approach. It requires less computing power and memory while preserving the integrity and meaning of the original content. By incorporating controllable features, these models can be fine-tuned to meet specific requirements, allowing for greater flexibility and adaptability. As a result, they can be applied to various tasks beyond text simplification, such as summarization, paraphrasing and translation.

5.1. Ablation Studies of Control Tokens

In this section, we study the influence of the control tokens we have used in the training. Each model has been trained with the reduced version of Wikilarge (a minimum similarity threshold of 0.85 between original and simplified sentences) to improve the quality of the results. A list of the most relevant training models according to the number of control tokens used can be found in Table 8 and Table 9.

To analyze the results, we began with the T5-small model without any additional tokens. Using this configuration, the model achieved a SARI value of 33.17 in TurkCorpus and 32.85 in the ASSET corpus. Next, we added each token individually and found that the best improvement with a single token was achieved with Work Rank feature (+6.96) in TurkCorpus and with Levenshtein-based similarity (+7.56) in ASSET. It's also worth noting that adding this feature resulted in an improvement of (+6.38) in TurkCorpus and adding Word Rank resulted in an improvement of (+7.03) in ASSET. These results are consistent with those published in [39]. Both features have the most significant impact on improving the Wikilarge TurkCorpus and ASSET based models.

After analyzing the TurkCorpus dataset, we discovered that Character Count feature enhanced the model's performance by (+4.25), Word Count by (+3.94), and Dependency Tree Depth by (+3.55). These results indicate that the control tokens had a similar impact on the model's performance. In comparison, the ASSET dataset showed that Word Count improved the model's performance by (+5.6), Dependency Tree Depth by (+2.96), and Character Count by (+2.94). This suggests that using only one control token can result in a significant improvement in the quality of simplifications. However, when analyzing the Language Model Fill-Mask control token on its own, the improvement achieved was only +1.3 in TurkCorpus and +0.21 in ASSET. This indicates that this feature, on its own, does not lead to a significant SARI improvement.

The incorporation of additional control tokens results in a progressive improvement in the quality of the generated output, and the best model for this work is achieved when using all control tokens. Both datasets exhibit this behaviour, with TurkCorpus achieving a SARI value of 43.25 with our model, representing a significant improvement of (+10.08) over the base model T5-small. Similarly, with ASSET, the best model obtains a SARI value of 44.56 using all the tokens.

When we analyzed the values of the control tokens for both the TurkCorpus and the ASSET datasets, we found that the values were either the same or very similar. This suggests that the control tokens have a stable effect on the performance of the model across different datasets. Furthermore, an analysis of the impact of the Language Model Fill-Mask control token used in conjunction with the other tokens reveals a slight improvement compared to models that do not use it. This denotes that the feature can be consistently incorporated into the simplification process.

5.2. Analysis of the Influence of the Control Tokens

The aim of including control tokens in the language model is to improve the quality of the output generated by the model. This section focuses on investigating whether the presence of control tokens affects the output of the model. The SimpleText corpus [20], which contains texts from the fields of computer science and medicine, was selected for this analysis.

The corpus is characterized by a large number of technical terms, making it challenging to simplify the texts for an audience without technical expertise in Computer Science or Medicine. The evaluation set was used to derive the control token values, and the sentences for the analysis were generated using the test set. The obtained control values were +WLR0.85+CLR0.75+LR0.65+DTDR0.95+WRR0.5+LMFR1.1, and the SARI value for the test set was 40.09.

Table 8. Ablation study on several trained models with different control tokens values on Wikilarge dataset (similarity≥ 0.85). SARI values were obtained on TurkCorpus

Model	SARI	FKGL
T5-small (No tokens)	33.17	10.01
+ LMFMR1.2	34.47	9.11
+DTDR0.8	36.72	9.01
+WLR0.75	37.11	8.10
+CLR0.75	37.42	7.97
+LR0.85	39.55	8.09
+WRR0.65	40.13	8.40
+WLR1.15+CLR0.75+DTDR0.8+WRR0.55+LMFR0.8	42.37	6.39
+WLR1.3+LR0.6+DTDR0.85+WRR0.75+LMFR1.2	42.62	6.78
+CLR0.95+LR0.7+DTDR0.9+WRR0.5+LMFR0.85	42.66	7.06
+WLR1.15+CLR0.9+DTDR1.45+LR0.65+LMFR1.25	42.80	6.99
+WLR1.2+CLR0.85+LR0.75+WRR0.75+LMFR1.1	43.04	6.57
+WLR1.25+CLR0.85+LR0.7+DTDR0.95+WRR0.6	43.04	6.85
+WLR1.25+CLR0.85+LR0.65+DTDR0.9+WRR0.6+LMFR1.15	43.25	6.38

Table 9. Ablation study on several trained models with different control tokens values on Wikilarge dataset (similarity≥ 0.85). SARI values were obtained on ASSET test datasets

Model	SARI	FKGL
T5-small (No tokens)	32.85	9.61
+ LMFMR1.3	33.06	9.51
+DTDR0.8	35.81	8.90
+WLR0.7	38.45	7.96
+CLR0.85	35.79	9.14
+LR0.8	40.41	7.56
+WRR0.65	39.88	8.20
+WLR1.1+CLR0.9+DTDR0.8+WRR0.8+LMFR0.95	42.25	6.50
+WLR1.05+LR0.6+DTDR0.8+WRR0.75+LMFR1.3	44.30	6.22
+CLR0.95+LR0.6+DTDR0.8+WRR0.75+LMFR1.25	44.19	6.57
+WLR1.3+CLR0.9+DTDR0.8+LR0.6+LMFR1.0	43.55	5.78
+WLR1.15+CLR0.85+LR0.65+WRR0.8+LMFR0.95	44.36	6.26
+WLR1.0+CLR0.9+LR0.65+DTDR0.8+WRR0.55	44.41	6.11
+WLR1.25+CLR0.8+LR0.65+DTDR0.8+WRR0.65+LMFR1.05	44.56	5.76

When the examples were analysed, it was found that the T5-small model often repeated the original sentence, making changes only by deleting words it considered less important. For example, it may delete information contained in parenthesis and acronyms, such as changing ”A mock PDA-based survey (PBS)” to ”A mock PDA-based survey”. In a few cases, it can change verbs, such as changing ”are emerging to become more prevalent” to ”are becoming more prevalent”. In contrast, the output generated by our model shows several features provided by the control tokens used. In all examples, the generated text is shorter than the original, indicating sentence compression. We have also included syntactic modification and paraphrasing.

One example is as follows: ”In particular, the ubiquitous access to remote resources is one of the most interesting characteristics achievable by using mobile devices such as Personal Digital Assistants, cellular phones, and tablets.”. This sentence commences with an introductory phrase, ”In particular” and has a noun phrase ”the ubiquitous access to remote resources” as its subject. In the simplified sentence the main clause is ”Using mobile devices such as Personal Digital Assistants, cellular phones, and tablets” which directly emphasizes the role of these devices in accessing remote resources. The simplification also changes the focus. The first sentence focuses on the ”interesting characteristics” achieved by using mobile devices, while the second sentence emphasizes that using these devices is ”one of the best ways” to access remote resources. This shifts the focus from the characteristics of the technology itself to the benefits it brings to the user.

One of the most frequently used features in the model is lexical substitution, where words are replaced with simpler alternatives. For instance, the word ”prevalent” is replaced with ”popular”. Similarly, ”browsed” is substituted with ”used”. Another interesting substitution is ”ambulatory setting” for ”hospital”.

Another example is the sentence: “In distributed computing paradigm, mobile surrogate systems migrates from one host in a network to another”. The term “paradigm” can be confusing for some people. Additionally, the use of the word ”migrates” implies a more technical process than is necessary to convey the idea. The simplified version removes the first term and uses the more commonly understood term “move” instead of “migrates”. This simplification clarifies the idea by removing unnecessary technical language.

6. Conclusions and Future Works

The approach we have presented utilizes a Deep Learning model known as T5 that has been pre-trained on a large corpus of text. To make the text simplification process more effective, we have incorporated several features into the model. By incorporating these features, our model can transform the complex text into a more straightforward and accessible form while still retaining the original meaning and context. Overall, our approach demonstrates the potential of leveraging pre-trained language models and transfer learning for text simplification tasks.

Our approach allows us to control the simplification result by selecting specific values for each of the previously trained features. This increases the flexibility of the system, as it is possible to generate different simplified versions, controlling, for example, the length of the text or its lexical richness. In addition, the architecture supports new features without having to modify any previous components, making it an ideal approach for experimentation. In order to train our text simplification model, we have used Wikilarge as our primary source of data. However, we recognize that while Wikilarge is a valuable resource, it is a large and diverse dataset that may contain noise and irrelevant data. Therefore, we have created a smaller dataset with higher semantic quality. This dataset contains high-quality text that has been selected to ensure that it is relevant and suitable for text simplification. By using a smaller dataset with higher semantic quality, we aim to achieve more accurate and reliable results from our model.

To validate and test the effectiveness of our text simplification approach, we have used two different datasets: Turkcorpus and ASSET. These datasets were selected for their diversity in text complexity and their suitability for text simplification tasks. By testing our model on these datasets, we aimed to evaluate its performance and effectiveness in simplifying complex text.

Our model has demonstrated considerable success in simplifying text and achieved results that are on par with state-of-the-art approaches. It has obtained a SARI value of 43.25 on the TurkCorpus corpus and a value of 44.56 on the ASSET corpus, which are notable accomplishments in the field. Remarkably, these results were achieved using a smaller model size, with only 60 million parameters, as compared to other studies in the field. This suggests that our model is not only accurate but also more efficient and resource-friendly than some of the existing approaches, making it a promising solution for various applications requiring text simplification.

Using language models with reduced parameters is crucial for several reasons. Firstly, it ensures that these models are accessible to a wider range of users, including researchers and developers who may not have access to high-end hardware or computational resources. This democratization of AI technology allows more individuals and organizations to take advantage of language models in their work.

Moving forward, we plan to extend our approach to an additional area of interest, specifically Digital Humanities (DH). This interdisciplinary field involves the application of computational methods and tools to the study of human culture, history, and literature. The use of text simplification in historical texts is important because it can make these texts more accessible to a wider public, including students, researchers, and the general public. By simplifying complex linguistic structures and archaic vocabulary, more people can learn about and appreciate the rich cultural heritage contained in these works.

Over time, languages have experienced significant evolution, with some words and syntactic structures becoming outdated and supplanted by others. Additionally, the meanings of many words have shifted, taking on new nuances that differ from their original usage. As a result, modern readers may find it challenging to understand a literary piece from three centuries ago, due to elements such as extended sentences and the employment of particular words. Consequently, contemporary readers may struggle to comprehend a literary work from three centuries ago due to factors such as lengthy sentences and the utilization of specific words [40]. To make the simplifications, we pretend to use the same feature control tokens but using the mT5 architecture [68]. This model has been shown to give good results in text simplification approaches in Spanish [58].

Acknowledgments. This work has been partially funded by the Spanish Research Agency (AEI) through the CLARA-NLP project with sub-projects in three domains: Digital Humanities [40] (PID2020-116001RB-C32), Medicine [9] (PID2020-116001RA-C33) and Finance [43] (PID2020-116001RB-C31).

References

1. Akiba, T., Sano, S., Yanase, T., Ohta, T., Koyama, M.: Optuna: A next-generation hyperparameter optimization framework (2019)
2. Alarcon, R., Moreno, L., Martínez, P., Macías, J.A.: Easier system. evaluating a spanish lexical simplification proposal with people with cognitive impairments. International Journal of Human–Computer Interaction 0(0), 1–15 (2022)
3. Althunayyan, S., Azmi, A.: Automated text simplification: A survey. ACM Computing Surveys 54, Article no. 43 (03 2021)
4. Alva-Manchego, F., Bingel, J., Paetzold, G., Scarton, C., Specia, L.: Learning how to simplify from explicit labeling of complex-simplified text pairs. In: Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers). pp. 295–305. Asian Federation of Natural Language Processing, Taipei, Taiwan (Nov 2017)
5. Alva-Manchego, F., Martin, L., Bordes, A., Scarton, C., Sagot, B., Specia, L.: ASSET: A dataset for tuning and evaluation of sentence simplification models with multiple rewriting transformations. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. pp. 4668–4679. Association for Computational Linguistics, Online (Jul 2020)
6. Alva-Manchego, F., Martin, L., Scarton, C., Specia, L.: EASSE: Easier automatic sentence simplification evaluation. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations. pp. 49–54. Association for Computational Linguistics, Hong Kong, China (Nov 2019)
7. Alva-Manchego, F., Scarton, C., Specia, L.: Data-driven sentence simplification: Survey and benchmark. Computational Linguistics 46(1), 135–187 (2020)
8. Amancio, M., Specia, L.: An analysis of crowdsourced text simplifications. In: Proceedings of the 3rd Workshop on Predicting and Improving Text Readability for Target Reader Populations (PITR). pp. 123–130. Association for Computational Linguistics, Gothenburg, Sweden (Apr 2014)
9. Campillos-Llanos, L., Terroba, A., Zakhir, S., Valverde-Mateos, A., Capllonch-Carrión, A.: Building a comparable corpus and a benchmark for spanish medical text simplification. Procesamiento del Lenguaje Natural 69, 189–196 (2022)
10. Campillos-Llanos, L., Valverde-Mateos, A., Capllonch-Carrión, A., Moreno-Sandoval, A.: A clinical trials corpus annotated with umls entities to enhance the access to evidence-based medicine (2021)
11. Carroll, J., Minnen, G., Canning, Y., Devlin, S., Tait, J.: Practical simplification of english newspaper text to assist aphasic readers. Proceedings of AAAI'98 Workshop on Integrating Artificial Intelligence and Assistive Technology pp. 7–10 (1998)
12. Chandrasekar, R., Doran, C., Srinivas, B.: Motivations and methods for text simplification. In: COLING 1996 Volume 2: The 16th International Conference on Computational Linguistics (1996)
13. Clive, J., Cao, K., Rei, M.: Control prefixes for parameter-efficient text generation. In: Proceedings of the 2nd Workshop on Natural Language Generation, Evaluation, and Metrics (GEM). pp. 363–382. Association for Computational Linguistics, Abu Dhabi, United Arab Emirates (Hybrid) (Dec 2022)
14. Cumbicus-Pineda, O.M., Gonzalez-Dios, I., Soroa, A.: A syntax-aware edit-based system for text simplification. In: Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021). pp. 324–334. INCOMA Ltd., Held Online (Sep 2021)
15. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding (2019)

16. Dong, Y., Li, Z., Rezagholizadeh, M., Cheung, J.C.K.: EditNTS: An neural programmer-interpreter model for sentence simplification through explicit editing. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. pp. 3393–3402. Association for Computational Linguistics, Florence, Italy (Jul 2019)
17. Drndarević, B., Saggion, H.: Towards automatic lexical simplification in Spanish: An empirical study. In: Proceedings of the First Workshop on Predicting and Improving Text Readability for target reader populations. pp. 8–16. Association for Computational Linguistics, Montréal, Canada (Jun 2012)
18. Ermakova, L., Bellot, P., Braslavski, P., Kamps, J., Mothe, J., Nurbakova, D., Ovchinnikova, I., San-Juan, E.: Overview of simpletext clef 2021.workshop and pilot tasks. In Conference and Labs of the Evaluation Forum (2021)
19. Ermakova, L., Mothe, J., SanJuan, E.: Overview of the simpletext workshop at inforsid-2021: Scientific text simplification and popularization. In Conference and Labs of the Evaluation Forum (2021)
20. Ermakova, L., Ovchinnikov, I., Kamps, J., Nurbakova, D., Araújo, S., Hannachi, R.: Overview of the clef 2022 simpletext task 3: Query biased simplification of scientific texts. Proceedings of the Working Notes of CLEF 2022 - Conference and Labs of the Evaluation, Bologna, Italy pp. 2792–2804 (2022)
21. Evans, R.J., Orasan, C., Dornescu, I.: An evaluation of syntactic simplification rules for people with autism. In: PITR@EACL (2014)
22. Feng, Y., Qiang, J., Li, Y., Yuan, Y., Zhu, Y.: Sentence simplification via large language models. ArXiv abs/2302.11957 (2023)
23. Févry, T., Phang, J.: Unsupervised sentence compression using denoising auto-encoders. In: Proceedings of the 22nd Conference on Computational Natural Language Learning. pp. 413–422. Association for Computational Linguistics, Brussels, Belgium (Oct 2018)
24. Ficler, J., Goldberg, Y.: Controlling linguistic style aspects in neural language generation. In: Proceedings of the Workshop on Stylistic Variation. pp. 94–104. Association for Computational Linguistics, Copenhagen, Denmark (Sep 2017)
25. Glavaš, G., Štajner, S.: Simplifying lexical simplification: Do we need simplified corpora? In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers). pp. 63–68. Association for Computational Linguistics, Beijing, China (Jul 2015)
26. Horn, C., Manduca, C., Kauchak, D.: Learning a lexical simplifier using Wikipedia. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). pp. 458–463. Association for Computational Linguistics, Baltimore, Maryland (Jun 2014)
27. Howard, J., Ruder, S.: Universal language model fine-tuning for text classification. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 328–339. Association for Computational Linguistics, Melbourne, Australia (Jul 2018)
28. Khallaf, N., Sharoff, S., Soliman, R.: Towards Arabic sentence simplification via classification and generative approaches. In: Proceedings of the The Seventh Arabic Natural Language Processing Workshop (WANLP). pp. 43–52. Association for Computational Linguistics, Abu Dhabi, United Arab Emirates (Hybrid) (Dec 2022)
29. Khan, A.W., Al-Obeidat, F., Khalid, A., Amin, A., Moreira, F.: Sentence embedding approach using lstm auto-encoder for discussion threads summarization. In: Computer Science and Information Systems, Vol. 20, No. 4. (2023)
30. Kikuchi, Y., Neubig, G., Sasano, R., Takamura, H., Okumura, M.: Controlling output length in neural encoder-decoders. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing. pp. 1328–1338. Association for Computational Linguistics, Austin, Texas (Nov 2016)

31. Kincaid, P., Fishburne, R.P., Rogers, R.L., Chissom, B.S.: Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel (1975)
32. Lastra-Díaz, J.J., Lara-Clares, A., Garcia-Serrano, A.: Hesml: a real-time semantic measures library for the biomedical domain with a reproducible survey. BMC bioinformatics 23(1), 23–23 (2022)
33. Levenshtein, V.I.: Binary codes capable of correcting deletions, insertions, and reversals. Soviet physics. Doklady 10, 707–710 (1965)
34. Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., Zettlemoyer, L.: 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. pp. 7871–7880. Association for Computational Linguistics, Online (Jul 2020)
35. Li, B., Yang, P., Zhao, H., Zhang, P., Liu, Z.: Hierarchical sliding inference generator for question-driven abstractive answer summarization. ACM Trans. Inf. Syst. 41(1) (2023)
36. Mallinson, J., Sennrich, R., Lapata, M.: Sentence compression for arbitrary languages via multilingual pivoting. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. pp. 2453–2464. Association for Computational Linguistics, Brussels, Belgium (Oct-Nov 2018)
37. Martin, L., Fan, A., de la Clergerie, É., Bordes, A., Sagot, B.: MUSS: Multilingual unsupervised sentence simplification by mining paraphrases. In: Proceedings of the Thirteenth Language Resources and Evaluation Conference. pp. 1651–1664. European Language Resources Association, Marseille, France (Jun 2022)
38. Martin, L., Humeau, S., Mazaré, P.E., de La Clergerie, É., Bordes, A., Sagot, B.: Reference-less quality estimation of text simplification systems. In: Proceedings of the 1st Workshop on Automatic Text Adaptation (ATA). pp. 29–38. Association for Computational Linguistics, Tilburg, the Netherlands (Nov 2018)
39. Martin, L., Sagot, B., de la Clergerie, É., Bordes, A.: Controllable sentence simplification. arXiv preprint arXiv:1910.02677 (2019)
40. Menta, A., Sánchez-Salido, E., García-Serrano, A.: Transcripción de periódicos históricos: aproximación clara-hd. SEPLN-PD 2022: Annual Conference of the Spanish Association for Natural Language Processing 2022: Projects and Demonstrations 69, 70–74 (2022)
41. Menta, A., García-Serrano, A.: Controllable sentence simplification using transfer learning. In: Simple Text Task at PAN - CLEF 2022 - Conference and Labs of the Evaluation Forum, Bolonia, Septb. 5-8, 2022. CEUR-WS, vol 3180, pp: 2818-2825. ISSN: 1613-0073 https://ceur-ws.org/Vol-3180/ (2022)
42. Monteiro, J.C., Aguiar, M., Araújo, S.: Using a pre-trained simplet5 model for text simplification in a limited corpus. In: Conference and Labs of the Evaluation Forum (2022)
43. Moreno Sandoval, A., Gisbert, A., Montoro Zamorano, H.: Fint-esp: A corpus of financial reports in spanish (01 2020)
44. Nisioi, S., Štajner, S., Ponzetto, S.P., Dinu, L.P.: Exploring neural text simplification models. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). pp. 85–91. Association for Computational Linguistics, Vancouver, Canada (Jul 2017)
45. North, K., Ranasinghe, T., Shardlow, M., Zampieri, M.: Deep learning approaches to lexical simplification: A survey. arXiv (05 2023)
46. Paetzold, G., Specia, L.: SemEval 2016 task 11: Complex word identification. In: Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016). pp. 560–569. Association for Computational Linguistics, San Diego, California (Jun 2016)
47. Paetzold, G., Specia, L.: Lexical simplification with neural ranking. In: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume

2, Short Papers. pp. 34–40. Association for Computational Linguistics, Valencia, Spain (Apr 2017)
48. Paetzold, G.H., Specia, L.: A survey on lexical simplification. J. Artif. Int. Res. 60(1), 549–593 (sep 2017)
49. Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.J.: Exploring the limits of transfer learning with a unified text-to-text transformer. J. Mach. Learn. Res. 21(1) (jan 2020)
50. Reimers, N., Gurevych, I.: Sentence-bert: Sentence embeddings using siamese bert-networks. In: Conference on Empirical Methods in Natural Language Processing (2019)
51. Rello, L., Baeza-Yates, R.A., Dempere-Marco, L., Saggion, H.: Frequent words improve readability and short words improve understandability for people with dyslexia. In: IFIP TC13 International Conference on Human-Computer Interaction (2013)
52. Rubio, A., Martínez, P.: Hulat-uc3m at simpletext@clef-2022: Scientific text simplification using bart. In: Conference and Labs of the Evaluation Forum (2022)
53. Saggion, H., Hirst, G.: Automatic Text Simplification. Morgan & Claypool Publishers (2017)
54. Saggion, H., Štajner, S., Bott, S., Mille, S., Rello, L., Drndarevic, B.: Making it simplext: Implementation and evaluation of a text simplification system for spanish. ACM Transactions on Accessible Computing 6(4) (may 2015)
55. Scarton, C., Specia, L.: Learning simplifications for specific target audiences. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). pp. 712–718. Association for Computational Linguistics, Melbourne, Australia (Jul 2018)
56. Schroff, F., Kalenichenko, D., Philbin, J.: FaceNet: A unified embedding for face recognition and clustering. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE (jun 2015)
57. Sheang, K.C., Saggion, H.: Controllable sentence simplification with a unified text-to-text transfer transformer. In: Proceedings of the 14th International Conference on Natural Language Generation. pp. 341–352. Association for Computational Linguistics, Aberdeen, Scotland, UK (Aug 2021)
58. Štajner, S., Sheang, K., Saggion, H.: Sentence simplification capabilities of transfer-based models. AAAI 36(11), 12172–12180 (2022)
59. Štajner, S.: Automatic text simplification for social good: Progress and challenges. In: Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021. pp. 2637–2652. Association for Computational Linguistics, Online (Aug 2021)
60. Štajner, S., Béchara, H., Saggion, H.: A deeper exploration of the standard PB-SMT approach to text simplification and its evaluation. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers). pp. 823–828. Association for Computational Linguistics, Beijing, China (Jul 2015)
61. Stajner, S., Ferres, D., Shardlow, M., North, K., Zampieri, M., Saggion, H.: Lexical simplification benchmarks for english, portuguese, and spanish (2022)
62. Štajner, S., Popović, M.: Automated text simplification as a preprocessing step for machine translation into an under-resourced language. In: Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019). pp. 1141–1150. INCOMA Ltd., Varna, Bulgaria (Sep 2019)
63. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In: Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2. p. 3104–3112. NIPS'14, MIT Press, Cambridge, MA, USA (2014)
64. Tarouti, F.A., Kalita, J.K., McGrory, C.: Sentence simplification for question generation. In: Proceedings of the Conference of the North American Chapter of the ACL: Human Language Technologies. pp. 609–617 (2010)

65. Vu, T., Hu, B., Munkhdalai, T., Yu, H.: Sentence simplification with memory-augmented neural networks. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers). pp. 79–85. Association for Computational Linguistics, New Orleans, Louisiana (Jun 2018)
66. Xu, W., Callison-Burch, C., Napoles, C.: Problems in current text simplification research: New data can help. Transactions of the Association for Computational Linguistics 3, 283–297 (2015)
67. Xu, W., Napoles, C., Pavlick, E., Chen, Q., Callison-Burch, C.: Optimizing statistical machine translation for text simplification. Transactions of the Association for Computational Linguistics 4, 401–415 (2016)
68. Xue, L., Constant, N., Roberts, A., Kale, M., Al-Rfou, R., Siddhant, A., Barua, A., Raffel, C.: mT5: A massively multilingual pre-trained text-to-text transformer. In: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 483–498. Association for Computational Linguistics, Online (Jun 2021)
69. Zhang, X., Lapata, M.: Sentence simplification with deep reinforcement learning. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. pp. 584–594. Association for Computational Linguistics, Copenhagen, Denmark (Sep 2017)
70. Zhang, Y., Ye, Z., Feng, Y., Zhao, D., Yan, R.: A constrained sequence-to-sequence neural model for sentence simplification. ArXiv abs/1704.02312 (2017)
71. Zhao, H., Cao, J., Xu, M., Lu, J.: Variational neural decoder for abstractive text summarization. In: Computer Science and Information Systems, Vol. 17, No. 2, 537–552. (2020)
72. Zhao, S., Meng, R., He, D., Saptono, A., Parmanto, B.: Integrating transformer and paraphrase rules for sentence simplification. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. pp. 3164–3173. Association for Computational Linguistics, Brussels, Belgium (Oct-Nov 2018)
73. Zilio, L., Braga Paraguassu, L., Leiva Hercules, L.A., Ponomarenko, G., Berwanger, L., Bocorny Finatto, M.J.: A lexical simplification tool for promoting health literacy. In: Proceedings of the 1st Workshop on Tools and Resources to Empower People with REAding DIfficulties (READI). pp. 70–76. European Language Resources Association, Marseille, France (May 2020)
74. Zou, J., Aliannejadi, M., Kanoulas, E., Pera, M.S., Liu, Y.: Users meet clarifying questions: Toward a better understanding of user interactions for search clarification. ACM Transactions on Information Systems 41, 1 – 25 (2022)

Antonio Menta is a PhD student in Computer Science. He joined ETSI Informática - UNED in 2015 to study a Master's degree in Computer Systems and Languages. Subsequently, he started his PhD research on the application of Neural Networks in Digital Humanities. His current research uses language models to simplify Spanish newspapers from the 18th century. https://orcid.org/0000-0002-3172-2829 [1]

Ana Garcia-Serrano is a PhD in Computer Science. Associated Professor at the Artificial Intelligence Department - UPM, she joined the ETSI Informática - UNED in June 2007. Since then, she is participating in the NLP & IR consolidated research group of the university, being the technical coordinator in different research projects. For a list of her work visit http://orcid.org/0000-0003-0975-7205 [2].

Received: September 12, 2023; Accepted: February 05, 2024.