NATIONAL AND KAPODISTRIAN UNIVERSITY OF ATHENS

SCHOOL OF SCIENCE
DEPARTMENT OF INFORMATICS AND TELECOMMUNICATIONS

INTERDEPARTMENTAL MASTER'S PROGRAM

"LANGUAGE TECHNOLOGY"

THESIS

Document-Level Text Simplification

Dimitra K. Kontoe

Supervisor: **Dimitrios Galanis,** Researcher C' (ILSP)

ATHENS

JULY 2023

ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΔΙΑΤΜΗΜΑΤΙΚΟ ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ

«ΓΛΩΣΣΙΚΗ ΤΕΧΝΟΛΟΓΙΑ»

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Απλοποίηση Κειμένου ως Ολότητας

Δήμητρα Κ. Κοντοέ

Επιβλέπων: **Δημήτριος Γαλάνης,** Ερευνητής Γ’ (ΙΕΛ)

ΑΘΗΝΑ

ΙΟΥΛΙΟΣ 2023

THESIS

Document-Level Text Simplification

Dimitra K. Kontoe
A.M.: lt1200028

SUPEVISOR: **Dimitrios Galanis,** Researcher C' (ILSP)

EXAMINATION COMMITTEE:
Vassileios Papavassiliou, Research Associate (ILSP)
Harilaos Papageorgiou, Researcher A' (ILSP)

July 2023

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Απλοποίηση Κειμένου ως Ολότητας

Δήμητρα Κ. Κοντοέ
Α.Μ.: lt1200028

ΕΠΙΒΛΕΠΩΝ: **Δημήτριος Γαλάνης,** Ερευνητής Γ’ (ΙΕΛ)

ΕΞΕΤΑΣΤΙΚΗ ΕΠΙΤΡΟΠΗ: **Βασίλειος Παπαβασιλείου,** Συνεργαζόμενος Ερευνητής (ΙΕΛ)
Χαρίλαος Παπαγεωργίου, Ερευνητής Α’ (ΙΕΛ)

Ιούλιος 2023

ABSTRACT

Text simplification is widely recognized as a valuable technique for improving knowledge dissemination and enhancing information accessibility. By reducing the linguistic complexity of a text, it enables lay readers to better understand the content. While significant research efforts in NLP have been dedicated to automatic simplification methods, document-level text simplification (ADTS) remains a relatively new and underexplored area. This thesis addresses the task of ADTS by introducing a sentence deletion-driven approach. The proposed method aligns sentences between comparable source-simplified text pairs, revealing the transformation operations involved in simplifying a text, including sentence deletion, insertion, merging, splitting, and paraphrasing. A set of eight aligners utilizing Sentence Transformers embeddings were developed, among which "all-mpnet-base-v2 with Itermax" achieved the highest evaluation results in most of the operation categories. Notably, deletion emerged as the most prevalent and effectively captured operation, with micro P/R/F scores of 90.0/ 82.62/ 86.15, macro P/R/F scores of 87.33/ 81.21/ 84.16, and weighted macro P/R/F scores of 92.58/82.62/87.32. Subsequently, the aligner was utilized on a subset of the "D-Wikipedia" dataset to automatically generate a large-scale corpus, which served as training data for 2 binary classifiers: an SVM-based model and a BERT-based model. They were trained to predict sentence deletions for ADTS and achieved almost identical results (SVM-based: P/R/F 70.71/70.88/70.78, BERT-based: P/R/F 70.38/70.07/70.20). Derived from these predictions two sets of simplified texts was generated. For the assessment of these deletion-dependent simplifiers' performance, 6 baselines were established alongside an oracle model, and the corresponding set of simplified texts from the "D-Wikipedia" dataset was used as the reference set. The findings indicated the superiority of the BERT-based simplifier over the SVM-based counterpart in terms of D-SARI (25.64 compared to 21.21). Furthermore, the performance of the oracle model affirmed the effectiveness of the deletion based ADTS approach. It was also highlighted that existing measures lack adequacy in comprehensively evaluating ADTS systems.

SUBJECT AREA: Text Simplification

KEYWORDS: document-level, deletion-based, unsupervised monolingual sentence alignment, sentence embeddings, text classification, simplification operations

ΠΕΡΙΛΗΨΗ

Η απλοποίηση κειμένου αναγνωρίζεται ευρέως ως μια πολύτιμη τεχνική για την ενίσχυση της διάχυσης της γνώσης και της προσβασιμότητας της πληροφορίας. Μέσω της εξάλειψης της γλωσσικής πολυπλοκότητας σε ένα κείμενο, δίνεται στους αναγνώστες η δυνατότητα να κατανοήσουν καλύτερα το περιεχόμενό του. Παρά τις σημαντικές ερευνητικές προσπάθειες στην περιοχή της αυτόματης απλοποίησης κειμένου, η απλοποίηση κειμένου ως ολότητας αποτελεί ένα νέο πεδίο στον τομέα thw Επεξεργασίας Φυσικής Γλώσσας (NLP) που δεν έχει εξεταστεί επαρκώς. Η παρούσα εργασία προσεγγίζει την αυτόματη απλοποίηση κειμένου ως ολότητας με βάση τη διαγραφή προτάσεων. Η προτεινόμενη μέθοδος υλοποιεί αντιστοιχίσεις προτάσεων μεταξύ ζευγαριών συγκρίσιμων κειμένων (πρωτότυπο-απλοποιημένο κείμενο), για την αυτόματη μοντελοποίηση των τεχνικών απλοποίησης που έχουν προταθεί για κλίμακα κειμένου, συμπεριλαμβανομένης της διαγραφής προτάσεων, της προσθήκης προτάσεων, της συγχώνευσης προτάσεων, της διαίρεσης προτάσεων και της παράφρασης. Αναπτύχθηκαν συνολικά 8 μοντέλα αντιστοίχισης με χρήση Sentence Transformers Embeddings, εκ των οποίων το all-mpnet-base-v2 σε συνδυασμό με τον αλγόριθμο Itermax πέτυχε την βέλτιστη επίδοση στην πλειονότητα των κατηγοριών. Αξιοσημείωτο είναι ότι η διαγραφή αναδείχθηκε ως η πλέον κυρίαρχη κατηγορία, επιστρέφοντας τα πιο υψηλά σκορ αντιστοίχισης (micro P/R/F 90,0/82,62/86,15, macro P/R/F 87,33/81,21/84,16 και weighted macro P/R/F 92,58/82,62/87,32). Στη συνέχεια, το καλύτερο μοντέλο αντιστοίχισης εφαρμόστηκε σε ένα υποσύνολο του συνόλου δεδομένων «D-Wikipedia» για την αυτόματη δημιουργία ενός νέου μεγάλης κλίμακας συνόλου δεδομένων, το οποίο και χρησιμοποιήθηκε για την εκπαίδευση, παραμετροποίηση και αξιολόγηση 2 μοντέλων δυαδικής ταξινόμησης (binary classifier): το ένα αναπτύχθηκε με βάση τον αλγόριθμό SVM και το δεύτερο με βάση το BERT. Εκπαιδεύθηκαν ώστε να προβλέπουν τη διαγραφή προτάσεων για την απλοποίηση κειμένου ως ολότητας και παρήγαγαν σχεδόν τα ίδια αποτελέσματα (μοντέλο SVM: P/R/F 70,71/70,88/70,78, μοντέλο BERT: P/R/F 70,38/70,07/70,20). Από αυτές τις προβλέψεις προέκυψαν δύο σύνολα απλοποιημένων κειμένων. Για την αξιολόγηση της απόδοσης των εν λόγω μοντέλων απλοποίησης μέσω διαγραφής προτάσεων, αναπτύχθηκαν 6 μοντέλα baseline καθώς και ένα μοντέλο oracle και ως σύνολο κειμένων αναφοράς χρησιμοποιήθηκε το αντίστοιχο σύνολο απλοποιημένων κειμένων από το «D-Wikipedia». Τα αποτελέσματα απέδειξαν την υπεροχή του συστήματος απλοποίησης που βασίστηκε στο BERT ως προς τη μετρική D-SARI (25,64 έναντι 21,21 του SVM). Επιπλέον, η απόδοση του μοντέλου oracle επιβεβαίωσε την εγκυρότητα της προσέγγισης και επαλήθευσε την ερευνητική υπόθεση. Επισημάνθηκε επίσης ότι οι υφιστάμενες μετρικές αξιολόγησης συστημάτων απλοποίησης κειμένου ως ολότητας παρουσιάζουν σημαντικούς περιορισμούς.

ΘΕΜΑΤΙΚΗ ΠΕΡΙΟΧΗ: Απλοποίηση κειμένου

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: κείμενο ως ολότητα, διαγραφή προτάσεων, μη επιβλεπόμενη αντιστοίχιση προτάσεων στην ίδια γλώσσα, διανυσματική αναπαράσταση προτάσεων, κατηγοριοποίηση κειμένου, τεχνικές απλοποίησης

CONTENTS

LIST OF FIGURES

LIST OF TABLES

PREFACE

This research study is a thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in “Language Technology” in the Department of Informatics and Telecommunications of National and Kapodistrian University of Athens (NKUA).

1 INTRODUCTION

1.1 Aim and Scope of the Study

The task of simplifying a given piece of text involves modifying its content and/or structure to enhance readability and comprehension without compromising its core idea [1]. Most data-driven simplification studies have been focused on modelling Automatic Sentence-level Text Simplification (ASTS) and designed systems which were restricted to generate one simpler sentence at a time, by applying three major types of operations: sentence splitting, word or phrase deletion, and paraphrasing [2], [3]. Recently, the task of Automatic Document-level Text Simplification (ADTS) was introduced by Sun et al. [4], along with a large-scale dataset called "D-Wikipedia." Sun et al., as well as Alva-Manchego et al. [5], argued that document simplification involves transformation operations that go beyond sentence boundaries, including sentence joining, sentence splitting, sentence deletion, sentence reordering, sentence addition, and anaphora resolution.

Since in many real-life scenarios, simplification is valuable only when it is performed at document level, it is significant to extend the scope of research on Automatic Text Simplification (ATS) and go beyond sentence-level approaches. This study addresses this need by delving into the task of Automatic Document-level Text Simplification (ADTS) for English texts using a data-driven approach. Specifically, the devised method employs a modular pipeline that leverages sentence-level alignments obtained from a comparable corpus, to train a model for simplifying texts with multiple sentences under a deletion-centric perspective, i.e., remove those sentences of the source text which contain complex or less important information.

1.2 Related Work

In this section we present some of the most important document-level approaches for text simplification, along with relevant datasets, and evaluation metrics.

1.2.1 Evaluation Metrics

Automatic evaluation of text simplification is a rather challenging task [6], [7]. Unlike other NLP tasks like Information Retrieval (IR), Natural Language Inference (NLI), or Question Answering (QA), where outputs are expected to be (more) specific and factual, Text Simplification allows for multiple equally valid options [8]. Additionally, the absence of native simplified-language speakers, as highlighted by Siddharthan et al. [9], further complicates the matter, as there are no objective criteria to definitively determine the simplicity of a text. Consequently, there is often limited consensus among experts (linguists) and various groups of native speakers regarding what constitutes a simple text, not to mention the disparity between non-native proficient adult speakers and native primary school students [10]. Another contributing factor to the ambiguity is the genre of the text; for instance, simplifying a scientific abstract necessitates a different approach compared to simplifying a political article.

The preferred method to measure the quality of a text simplification system is to collect human judgments on its output based on three criteria [10] - [14]: fluency (grammaticality), simplicity, and meaning preservation. However, manual evaluation is resource-intensive and time-consuming. To address these limitations, automatic measures are commonly employed to fine-tune and compare sentence-level simplification models. These measures, such as SARI [15], BLEU [16], and BERTScore [17], estimate the similarity between a system's output and a set of human-generated simplifications. This methodology is considered more objective as it is unaffected by personal biases [10]. Below we present automatic measures which were developed for sentence-level simplification and other text-to-text generation tasks [18], which are also applicable to ADTS.

- BLEU (BiLingual Evaluation Understudy) was originally introduced by Papineni et al. [16] as an evaluation metric for machine translation (MT) systems. Over time, it has been used for evaluating various generation tasks, including ASTS, which is often regarded as a monolingual translation task [19], [20]. BLEU captures lexical overlap between the generated and corresponding reference texts by averaging modified 1 to 4-sized n-gram precisions, proportion of n-grams in the output that appear in the reference as well. It also penalizes sentence shortening and word reordering. In the context of sentence simplification, BLEU score has been found to exhibit a stronger correlation with human evaluations of fluency and meaning preservation [20] - [22] compared to assessments of simplicity gain and perceived overall simplicity [11]. However, it is worth noting that the correlation reported in these studies was relatively low [23].
- SARI (System output Against References and against the Input sentence) score [15] is a refined measure developed for ASTS, particularly in the context of paraphrasing. It calculates the average F1-score for 3 n-gram overlap scores corresponding to 3 simplification operations: addition, keep, and deletion. That way, SARI attempts to quantify the adequacy of added, kept, and deleted n-grams of the simplified text by contrasting it first with the original text, to spot the differences, and secondly with the available references, to assess correctness. However, its reliability was criticized has been questioned in evaluating systems that implement changes beyond or in addition to lexical paraphrasing [11].
- The D-SARI metric introduced by Sun et al. [4], is a customized version of SARI specifically designed for ADTS. It follows the same calculation methodology as SARI, with the difference that each F1-score is multiplied by a penalty factor determined by the length of the output text in comparison to the reference text, either in terms of tokens or sentences.
- BERTScore is a relatively new metric which was proposed by Zhang et al. [17] for the evaluation of text generation tasks. It measures semantic equivalence between the ground truth and candidate text using a token-level matching function, specifically cosine similarity, applied to pre-trained contextualized embeddings from the BERT model [24]. The metric calculates Recall, Precision, and F1 score. Unlike BLEU, BERTScore leverages contextualized embeddings to capture synonyms and paraphrases that may not be present in the reference set. In the context of Automatic Simplification, BERTScore was included in the evaluation of automatic metrics in ASTS conducted by Alva-Manchego et al. [11], and BERTScore Recall was found to best correlate with human judgments in terms of direct assessment of the overall simplicity. Moreover, BERTScore was reported for the purpose of evaluating semantic relevance in another document-

level rewriting task, that of paraphrase generation [25], which shares similarities with simplification generation.

Overall, in the field of simplification research, numerous studies [8], [10], [11], [18], [23], [26], [27] have expressed concerns regarding the suitability of commonly used metrics for evaluating automatically generated simplified text. These studies have emphasized the absence of a universally accepted and comprehensive automatic measure in this regard.

1.2.2 Datasets

Research area of data-driven sentence simplification has long embraced Wikipedia and Simple English Wikipedia[1]-derived resources. Such datasets, PWKP/WikiSmall [28], Coster and Kauchack [29], Hwang et al. [30], Tomoyuki Kajiwara and Mamoru Komachi [31], WikiLarge [32], ASSET [22] consist of original-simplified sentence pairs which are extracted from sets of articles by means of alignment algorithms.

In 2015, Xu et al. [33] questioned the reliability of Simple Wikipedia-collected datasets by providing evidence of noisy alignments and dull or unsuitable simplifications. They also introduced the Newsela corpus, a simplification dataset comprising 1,130 news articles professionally reproduced into 4 simplified versions of different reading complexity levels (ranging from 0 to 5, with 0 being the original text and the 5 the most simplified). Although originally a document-level simplification resource, Newsela corpus has been mainly used for sentence simplification research purposes so far [32], [34] - [36]. Several scholars [1], [22], [37], [38] have acknowledged the high quality of the Newsela dataset, but even so, they have also highlighted the dataset's restrictive, non-open license.

Sun et al. [4] were the first to publish a large-scale open-source dataset specifically in accordance with the novel task of ADTS. Namely, the “D-Wikipedia” dataset was built automatically upon dumps from 170,000 Wikipedia and Simple English Wikipedia article pairs, and it consists of 143,000 pairs of comparable article abstracts, up to 1,000 words each. The pairs were split into training, validation, and test sets; see Table 1 below. As far as the six operations [6] on which ADTS is dependent, authors employed three annotators to identify them in a sample of 100 article pairs and provided relevant statistics (refer to Table 2 and 3) evidencing that most of the annotated articles involved all six of them, with deletion being the most prevalent (Table 3).

Table 1: Statistics for the D-Wikipedia Dataset

Statistics	Train	Validation	Test
Article pairs	132,000	3,000	8,000

[1] Simple English Wikipedia (https://simple.wikipedia.org/wiki/Main_Page) is an online encyclopaedia that offers free and open content. It acts as a companion to the regular English Wikipedia and is specifically designed to cater to English learners, children, and individuals who may be unfamiliar with certain topics or complex concepts. Article contributors are expected to follow some specific guidelines when writing content for Simple English Wikipedia. They are mainly referring to the use of Basic English words and simple structures.

Table 2: Estimated Percentage of Articles Containing Each Simplification Operation

Operation	Percentage of Articles
Sentence Joining	96%
Sentence Splitting	84%
Sentence Deletion	91%
Sentence Reordering	92%
Sentence Addition	92%
Anaphora Resolution	92%

Table 3: Estimated Percentage Distribution of Document-Level Simplification Operations

Operation	Frequency of Appearance
Sentence Joining	8%
Sentence Splitting	17%
Sentence Deletion	44%
Sentence Reordering	6%
Sentence Addition	16%
Anaphora Resolution	9%

1.2.3 Document-Level Text Simplification Approaches/Methods

Although ATS is a prolific research area in NLP, the approach of ADTS has received relatively less attention, particularly in terms of methods. Nevertheless, there have been significant preliminary studies in the field. For example, Alva-Manchego et al. [5] analyzed the Newsela dataset considering inter-sentence transformations that occur in the simplified version of articles and proposed a relevant taxonomy. They also examined the applicability of a standard neural sequence-to-sequence (seq-2-seq) model for ASTS in a pseudo-document-level context. They used it to generate simplified sentences in isolation, and then they assembled them into complete documents. According to the evaluation results, they concluded that simplifying an entire text requires considering aspects that span beyond single sentences. In another study, Sun et al. [4] employed the D-Wikipedia dataset to train four models as baselines for the document-level simplification task: a standard seq-2-seq Transformer model [39], a BART model [40], another Transformer-based model augmented with a context information module for the task of sentence simplification, SUC [41], and BertSumExtAbs [42], a model which achieved SOTA results in text summarization, in both extractive and abstractive settings, by introducing pretrained BERT [24] as a

document-level encoder. The evaluation results, from both automatic and manual assessments, highlighted the distinct nature of ADTS compared to ATS, as SUC's performance was found to be poor. Taking these findings into account, Sun et al. concluded that new models and methods need to be developed specifically for ADTS, as they encountered several challenges, including the low correlation between reported automatic metrics (D-SARI, SARI, BLEU, FKGL [43]) and human ratings.

Furthermore, there are other research works that adopt an operation-centric approach. For instance, in the context of ADTS, Srikanth [44] delved into the operation of content addition, in the form of introducing explanations, definitions or clarifications to complex ideas of the original text, and proposed a new task thereof, the Elaborative Simplification. Using the Newsela Corpus, Srikanth constructed a corpus of over 1.3K sentences that served as elaborations. This involved automatic extraction, manual verification, and annotation with labels related to their contextual specificity. Next, she utilized this corpus to establish two reliable baselines for the subtasks of contextual specificity prediction and elaboration generation. For contextual specificity prediction, she employed pretrained BERT, while for elaboration generation, she fine-tuned the pretrained GPT-2 [45]. The results showed that in such settings, only considering the context of the simplified text can be beneficial for both predicting the degree of contextual specificity and generating elaborations. Also, regarding the latter, specificity-guided generation, i.e., applying top-k sampling and then making a decision that is grounded on the prediction from contextual specificity model and the gold labels, yielded better performance. In practice, this suggested that a system can generate better quality text when it is informed of the type of the desired elaboration.

Regarding other operations, Zhong et al. [46] were the first to conduct a data-driven study on sentence deletion in ADTS, by hypothesizing that besides content, what accounts for sentence deletion is discourse-level information. They relied their analysis on the Newsela corpus, particularly on documents written for elementary and middle schoolers, and they created two datasets thereof, a manually and an automatically derived sentence-aligned corpus. The former served as a resource for inspecting various discourse factors which are associated with sentence deletion, such as document length, topics, rhetorical structure, and discourse relations and the latter for training classification models to predict sentence deletions. More specifically, they used logistic regression and feedforward neural networks as classifiers and experimented with different combinations of dense and sparse features, to capture sentence-level semantics, document-level information, and discourse relations. They also performed a relevant feature ablation study. The results indicated the difficulty of the task and highlighted the significance of document-level features in predicting sentence deletions across all examined settings.

Along similar lines, Zhang et al. [2] focused on sentence deletion as a salient discourse-level operation in ADTS and researched on the relation borne by discourse structures to sentence importance within the context of a document. They used a part of the Newsela corpus, automatically annotated by the alignment tool CATS [47] in terms of deleted-not deleted sentences, to explore the predictability of sentence deletions under three settings. First, they trained a baseline, a document-level two-layer Bi-LSTM, with a self-attention mechanism between the two layers and BERT embeddings as initial layer, to predict sentence deletions. Then, they applied an automatic news genre-specific discourse parser [48] to label each complex sentence of a single text with a category reflecting its function role around the main event. Subsequently, they built on top of the baseline, by proposing two refined models; one incorporated content type labels as additional features, and the other was trained to jointly predict both sentence deletion and discourse content type labels. Both methodologies demonstrated a substantial

enhancement in prediction performance, as evidenced by significant improvements in precision, recall, and F1 scores. These findings strongly suggest that incorporating discourse information is valuable when selecting content for the simplification of entire documents.

Another family of ADTS approaches is lay language generation for biomedical content. For example, Devaraj et al. [49] were the first to introduce a domain-specific dataset of paired technical abstracts and simplified summaries of reviews on various clinical topics. They also proposed a new SciBERT [50] masked probabilities-dependent metric that could efficiently estimate the technicality of language in which a text is written. Finally, they presented baselines for the task of paragraph-level simplification of medical texts that exploited BART augmented with a variant of unlikelihood loss training to explicitly penalize production of jargon terms. According to their results, thanks to the incorporated training objective, output summaries gained in simplicity and abstractiveness. Phatak et al. [51] employed the same dataset and tackled the task under a different methodology; they implemented a reinforcement learning (RL) algorithm (Self-Critical Sequence Training, SCST [52]) to build their baselines. They first fine-tuned the pretrained BART on the document-level paired dataset and secondly, they further trained the fine-tuned model to learn an optimal policy that maximizes two rewards specific for text simplification: relevance, i.e., semantic similarity with the original text, and simplicity, i.e., lexical plainness. Authors' extensive experimentations showed that the proposed method achieved comparable performance with other baselines when measured with commonly used metrics, while manual evaluation of the generated outputs affirmed improvement in fluency and coherence.

Another RL paradigm for ADTS has been recently launched by Laban et al. [53], but that time within an entirely unsupervised learning framework. In essence, the authors extended SCST algorithm and introduced *k*-SCST to train a text generator (GPT-2) under a reference-free reward-driven regime; accordingly, the model proposed several candidate simplifications (let *k* be candidates, $k>2$) and learned to elicit outputs that jointly maximized a reward across three factors: fluency, salience, and simplicity. Tested on the Newsela test set, the aforementioned methodology outperformed strong supervised and unsupervised baselines in terms of the SARI metric.

1.3 Contribution of the Study

While there is an abundance of research on sentence-level simplification, the applicability of its findings and methodologies to document-level text simplification is limited. This thesis stands out as one of the few data-driven studies focusing on deletion-based document-level text simplification, making significant contributions to the field. The main contributions of this research are as follows:

1) We devised a fully unsupervised and computationally efficient method for aligning sentences between source-simplified texts pairs. These alignments can capture the respective simplification operations applied, i.e., sentence deletion, insertion, merging, splitting, and paraphrasing.
2) Utilizing this method, we automatically generated a new large-scale aligned dataset from the "D-Wikipedia" Dataset, which can serve as a valuable resource for training and evaluating future simplification systems.
3) We trained novel classifiers on the generated dataset to accurately predict sentence deletions in an ADTS setting.

4) We demonstrated that the use of a large pretrained-model, such as BERT, for training such classifiers improved the performance in terms of D-SARI.
5) We also showed that the current evaluation measures for ADTS are highly influenced by the length of the generated simplified text.

2 PROPOSED SIMPLIFICATION METHODS

2.1 Approach and Background

Our methodology stemmed from the idea that the task of ADTS can be broken down into subproblems. This perspective potentially mirrors the process of simplification performed by human annotators, such as linguists. They typically begin by identifying areas of complexity within a source text, and then, guided by semantic and/or syntactic considerations, make decisions regarding whether to delete certain parts, retain them as they are, or possibly transform them [18]. Consequently, deleting content that is either of a) minor importance to the core meaning or b) excessively sophisticated/complex is given precedence and is prioritized over other simplification operations when addressing an entire document [54], [55]. Building upon these assumptions, we developed a deletion-based method for ADTS. This approach yields the following research outcomes:

- It proposes a novel unsupervised method for document-wide sentence alignments. Specifically, it sets up an automatic pipeline that transduces a sequence of sentences into a sequence of interdependent discrete labels to encode each of the following simplification operations: sentence deletion (1<=>null), insertion (null<=>1), merging (n<=>1), splitting (1<=>n), paraphrasing (1<=>1). In other words, the pipeline attempts to extract the operations that are used in simplifying a text.
- It builds a new resource (using the aforementioned pipeline) comprising over 100K sentences that are automatically labelled as deleted or retained, and mapped to a source document, and employs it to train several classifiers to directly predict deletions for document-level text simplification.
- It attempts to provide an estimate on the extent to which a deletion-only technique delivers simpler texts, by aggregating each model's outputs to generate and then evaluate target texts.

The primary reason for adopting this modular, deletion-oriented methodology was the lack of large-scale open-source parallel corpora specifically aligned at a fine-grained level for document simplification. This scarcity contrasts with other NLP text-to-text tasks, like machine translation. For example, although a recent large-scale dataset, the D-Wikipedia, became available for ADTS, it comprises pairs of comparable texts that cover the same topic but exhibit heterogeneous structures and lack specific editing guidelines. Consequently, many source-target pairs within the dataset differ significantly, posing challenges for training standard sequence-to-sequence models. This difficulty has been also noted in simplification approaches that employ sentence-level datasets [36]. The limitations of training encoder-decoder models in this context have been substantiated by both manual and automatic evaluation results obtained by Sun et al [4].

2.2 Sentence Alignment for Text Simplification

Text alignment can be outlined as the process of juxtaposing two or more texts to unveil correspondences between their textual units, namely paragraphs, sentences, words. It is regarded as an exceptionally valuable step in supporting, either in abstract or concrete terms, various downstream NLP tasks, such as machine translation,

paraphrase and simplification generation, question answering, and natural language inference [56]. In the context of data-driven research on sentence simplification, a significant focus has been placed on mining parallel monolingual sentence pairs to obtain training data. In this chapter, we will first introduce some of the key alignment methods used in this context and then discuss their limitations when it comes to document-level text simplification.

Xu et al. [33] adopted a greedy alignment strategy and utilized the Jaccard coefficient to gauge similarities between sentence pairs of adjacent versions of articles in the Newsela corpus, i.e., a sentence from the simpler version with a sentence from the immediate more complex one. Pairs with the highest similarity, exceeding a threshold-specific ratio, were regarded as aligned.

MASSAlign [57] provides a collection of unsupervised methods for matching monolingual parallel documents at both paragraph and sentence level. Built upon the assumption that both source and target texts share the same flow of information, and by leveraging a Vicinity-Driven alignment method: Vicinity-driven paragraph and sentence alignment for comparable corpora], they formulated a similarity matching algorithm under a dynamic programming paradigm by using TF-IDF cosine similarity as the measure. Given two documents/paragraphs, their method's output is a similarity matrix. Starting from the cell in the matrix that is closest to [0,0] and has a score higher than a predefined threshold, the alignment path is determined. It iteratively searches for corresponding sentence pairs in a hierarchy of vicinities. This approach enables the retrieval of sentence alignments that can model 1→1, 1→n, and n→1 relation.

Stanjer et al. introduced CATS [47], an unsupervised alignment tool which relies on lexical similarities and employs a greedy algorithm. They proposed three similarity measures that can be applied at the paragraph or sentence level using cosine similarity: character 3-gram overlapping ratio, average of word embedding vectors (Word2vec trained in English Wikipedia), and all the word embeddings in the chunk. They also developed two alignment methods relying on the assumption that every "simple" sentence/paragraph has at least one corresponding "source" counterpart. The first method selects the most similar pair using any of the suggested metrics, while the second method initially follows the same strategy but prioritizes preserving the order of sentences/paragraphs in the original text when choosing the sequence of alignments. Thus, the second method allows for 1→1, 1→n, and n→1 alignment instance.

Recently, Jiang et al. [34] introduced two high-quality manually annotated sentence-alignment datasets (Newsela-Manual and Wiki-Manual) and used them to fine-tune BERT with the aim of measuring semantic similarity. Building upon this, the authors proposed a two-fold process to enhance the retrieval of aligned sentence pairs from parallel corpora. The first step involved aligning paragraphs between the aligned documents, while the second one focused on training a CRF model to identify similar sentences within the aligned paragraphs. The model leveraged both sentence-level similarities and alignment label transitions, assuming that the content of two parallel documents followed a similar order. This NN model-dependent approach delivered better performance in the task of monolingual sentence alignment compared to previous model-agnostic ones. However, it is important to note that this performance improvement comes with the cost of requiring a substantial amount of annotated data.

The above-reported methods primarily aim at matching and extracting pairs of text segments to generate datasets for sentence-level simplification. They have been specifically designed for parallel corpora, such as Newsela, where the target text closely

resembles the original in structure and phrasing. Furthermore, all of them implement a unidirectional mechanism to identify the target text snippet that best matches the original. They rely on surface-level, shallow similarity measures, which may lose their effectiveness when tasked to estimate resemblance between two text sequences that contain significantly different lexicons or diverse grammatical structures. The aforementioned factors represent inherent restrictions in the context of text alignment for ADTS. For instance, the D-Wikipedia dataset consists of comparable texts, and as such, it calls for a robust and semantically aware alignment technique that does not stick to structural constraints, such as the order of information. Given that this study employs an operation-centric methodology to automatically generate data for the task of document-level simplification, the objective of the alignment technique is not to retrieve the maximum-similarity sentence pairs. Instead, the focus is on maximizing recall. This entails distributing any semantic relatedness between two texts among sentence pairs in a manner that uncovers as many optimal sentence intercorrelations as possible, resembling a pseudo-parallelization of texts. Through these intercorrelations, document-level simplification operations can be revealed.

2.2.1 Proposed Sentence Alignment Methods

The proposed alignment methods were built upon SimAlign [58], an unsupervised word aligner that leverages BERT-induced contextualized token embeddings. Despite not relying on parallel sentences, SimAlign achieves competitive or even superior performance compared to conventional alignment strategies. In our research, we extended this approach by incorporating contextually aware sentence embeddings for matching sentences across two texts. Namely, given a pair of comparable texts, the aligner first mapped each sentence to a vector space such that semantically related sentences were positioned close to each other. Next, a similarity matrix was constructed, with entries equaling the cosine similarity scores between each source sentence vector and each target sentence vector. To output any possible sentence alignments, we employed a bidirectional extraction technique. Importantly, our methodology was computationally efficient, operating without any cross-textual supervision or prerequisites, and yielding symmetrical document-wide alignments.

As to sentence embeddings, we experimented with various pre-trained models from the Sentence Transformers family [59]. This paradigm was recently established by Reimers and Gurevych [60] with the introduction of Sentence-BERT (SBERT), an adaptation of the standard pretrained BERT that was configured to derive semantically meaningful fixed-sized sentence embeddings to support tasks such as semantic textual similarity, semantic search, paraphrase mining, and natural language inference. SBERT fine-tuned BERT adjoined with a pooling layer into a twin network structure by optimizing one of the three objective functions, depending on the selected dataset: classification, regression, triplet. SBERT's sentence embeddings were experimentally shown to outperform sentence embeddings that are yielded either by averaging static token embeddings, such as GloVe [61], or BERT embeddings, or by using BERT's output CLS token.

To obtain sentence alignments, we experimented with two matching algorithms: Itermax and Match [58]. Itermax is a greedy approach that, relying on the similarity matrix, operated recursively to align two elements that were the reciprocal column-wise and row-wise maxima, i.e., one sentence was aligned to another only if the latter could be inversely aligned to the former. After each iteration, previously aligned positions were

excluded from the similarity matrix, while elements that had been unidirectionally aligned were multiplied by a discount factor. Most importantly, if the similarity with an already aligned sentence was exceptionally high, the algorithm would enable multiple sentence matching. Outputting all possible alignments was the condition that allowed the algorithm to stop recursion. On the other hand, Match algorithm adopts a non-greedy global optimization approach by using maximum-weight maximal matching problem. Sentences were represented as nodes in a bipartite graph where cosine similarity scores were used as weights. By computing maximum-weight maximal matching this method retrieved all possible globally optimal paths from the graph, which means, all possible pairs of aligned sentences.

The subsequent step involved utilizing the identified sentence alignments explicitly for document-level text simplification. Our objective was to harness the capabilities of our proposed method in automatically modeling simplification operations by leveraging its key feature: the ability to identify all valid sentence pairs, enabling the matching of multiple sentences to a single sentence and vice versa. Thus, given two comparable texts (the original and the simplified text) that were split into sentences and their predicted alignments, we devised a novel automatic annotator for identifying simplification operations and deriving operation-informed data instances.

The way in which the annotator functions is as follows: any source sentence whose index did not appear in the alignment results was considered for deletion (1<=>null alignment), while any source sentence that appeared more than once was treated as a splitting operation (1<=>n alignment). Similarly, any target sentence that did not appear in the alignment results was regarded as an insertion (null<=>1 alignment), and any target sentence that appeared more than once was considered for merging (n<=>1 alignment). In cases where a source sentence was exclusively mapped to a target sentence and vice versa, it was possible to model paraphrasing (1<=>1 alignment).

2.3 Proposed Sentence Deletion-Based Simplification Methods

The underlying principle behind our designed document-level simplification methods was the notion that enhancing readability and comprehension could be achieved by removing less important or excessively complex sentences from a text.

We formulated the task as a binary classification problem: assign each sentence in a text a label of either "deleted" or "retained" depending on its perceived significance and simplicity. To accomplish this, we constructed two binary classifiers with the objective of learning sentence representations to predict which sentences should be deleted for text simplification Using the predictions from these classifiers, we generated a new text that included only the subset of sentences that best represented a simplified version of the original text. For training the classifier, we explored with the use of Support Vector Machines (SVM) [62] and a BERT model [24] as potential approaches.

2.3.1 The SVM classifier

For the SVM model we mapped each sentence (of the source text) to a vector of the following features scores, after having applied the initial preprocessing steps of sentence tokenization, word tokenization, and stop words removal. The features we utilized were mainly derived from commonly employed features in the context of extractive summarization [63] – [66].

- Sentence-level Semantics
    - Ratio of Numerical Data: For each sentence, we counted the instances of numbers and digits proportionally to its length in terms of the total count of tokens. We assumed that sentences containing more numerical data are probably richer in information.
    - Ratio of Subordinating Conjunctions, Ratio of Coordinating Conjunctions: They were defined as the number of subordinating/coordinating conjunctions divided by the total number of tokens in a sentence. The intuition behind this feature was that subordinating clauses denote a more complex syntax than coordinating ones, and vice versa.
    - Ratio of Adverbs: It was defined similarly to the features above. High frequency of adverbs in a text could be interpreted as a sign of elaborated semantics.
    - Ratio of Nouns to Verbs: It was defined similarly to the features above, except that instead of the total count of tokens, we divided with the total count of verbs. Higher relative frequency of nouns compared to verbs in a text tends to create dense information.
    - Flesch Reading Ease Score [27]: It is a readability index that measures the complexity of text as a function of the weighted average length of sentences in terms of words, and the weighted average number of syllables per word. Score ranges from 0 to 100, with higher scores indicating passages that are easier to read.
    - Dale–Chall Readability Formula [67]: It is another statistical readability test that assesses the difficulty of a text using words and sentences counts. It also uses weights which are calibrated by a lookup table consisting of the 3000 most frequently used English words. Score ranges from 0 to 9.9, with higher scores indicating passages that are more difficult to read.
    - Sentence Position: Each sentence received a score according to its distance from the first sentence of the text. This score was normalized with respect to text length in terms of sentences. According to this approach, the greater the distance from the beginning of the text, the lower the score. Intuitively, sentences at the beginning or the end of a text are more likely to include important information.
- Document-level Features:
    - Term Frequency-Inverse Sentence Frequency (TF-ISF): It is a method to quantify the significance of a sentence as a function of its context. Both term frequency and inverse sentence frequency are aspects which involve properties of context, and consequently are document aware. More specifically, we calculated this feature by averaging TF-ISF values of all words in a sentence. TF-ISF score rises proportionally to the number of instances of a word in a sentence and it is offset by the number of sentences in a text that include the same word. Essentially, this feature assigned higher score to sentences that included words that were common enough to be considered important.
    - Sentence-to-Sentence Cohesion: To obtain this feature, first we computed Jaccard similarity between all possible pairs of sentences in the same text. Then, each sentence was assigned the average value of all its similarity

scores with the other sentences. The higher the score, the more likely that a sentence was highly informative.

- Similarity to the Most Important Sentence: Based on TF-ISF scores, we identified the most semantically significant sentence of each text. Then, we computed its similarity with every other sentence of the same text by using the cosine similarity method.

- BERT-Induced Feature:
  - Finally, we created an additional feature to model textual relatedness between the most informative sentence of a text (i.e., the sentence with the highest score according to Sentence-to-Sentence Cohesion) and each of the remaining ones. This feature leveraged the extensive language knowledge that is encoded by NLP foundation models in place of the surface-level measure of cosine similarity. More specifically, the idea was to modify the pretrained BERT base model to return outputs for sentence pair classification and then got the fine-tuned model to predict the probability of correlation between two sentences. The scores ranged from 1 to 0; the most informative sentence of a text was assigned 1 and each of the rest ones was assigned the probabilistic score which was predicted by the fine-tuned model.

    BERT base was made up of 12 stacked-up Transformer's encoder layers and implemented a multi-head self-attention mechanism. It was trained on two tasks: "Masked Word Prediction" and "Next Sentence Prediction", and as a byproduct of that training, it developed the ability to "understand" natural language.

2.3.2 The BERT classifier

BERT fine-tuning was also the second method that we implemented to build the sentence deletion-based simplifier. Specifically, we used the pre-trained BERT base model, we added an untrained linear layer on top of it and continued training the whole network on a dataset for the task of single sentence binary classification. Finally, we applied a sigmoid function on model's logits to output a probability value, and then we determined a threshold to obtain the predictions for "deleted" and "retained" instances. Essentially, we applied a transfer learning technique that did not require handcrafted features to operate, and we compared it with the SVM-based classifier.

3 EXPERIMENTS AND RESULTS

3.1 Sentence Alignment Experiments and Evaluation

The sentence alignment methods were implemented using the publicly available code of SimAlign[2] [58], incorporating sentence embeddings to meet the requirements of sentence-level matching. To test the hypothesis that the proposed alignment method was adequate for the task of operation-guided ADTS, experimented with various configurations of its components, including the embedding model and the matching algorithm. Intrinsic evaluations were conducted to assess the overall performance as well as the performance specific to each operation (deletion, splitting, insertion, merging, paraphrasing).

3.1.1 Embeddings

To split each source-target text pair into sentences we applied NLTK's sentence tokenizer [68]. For sentence embedding, we experimented with four different pretrained models from the HuggingFace Sentence Transformers library [69]. These models were the top four best-performing models on producing general-purpose sentence embeddings suitable for cosine similarity scoring function: 'all-mpnet-base-v2', 'all-distilroberta-v1', 'all-MiniLM-L12-v2', and 'all-MiniLM-L6-v2'. They were trained on a 1B sentence pairs dataset with a mean pooling layer under a contrastive learning objective; given a sentence from the pair, the model should predict which out of a set of randomly sampled sentences, was paired with it in the dataset. Additional details about the pretrained models [69] can be found in the Table 4.

Table 4: Comparison of Sentence-Transformers Pre-trained Models

Comparison	**all-mpnet-base-v2**	**all-distilroberta-v1**	**all-MiniLM-L12-v2**	**all-MiniLM-L6-v2**
Base Model	microsoft/mpnet-base	distilroberta-base	microsoft/MiniLM-L12-H384-uncased	nreimers/MiniLM-L6-H384-uncased
Max Sequence Length	384	512	256	256
Output Vector Dimensions	768	768	384	384
Model Size	420 MB	290 MB	120 MB	80 MB

[2] https://github.com/cisnlp/simalign

Encoding Speed	2800 sentence/sec on V100 GPU	4000 sentence/sec on V100 GPU	7500 sentence/sec on V100 GPU	14200 sentence/sec on V100 GPU
Training Data	1B+ sentence pairs	1B+ sentence pairs	1B+ sentence pairs	1B+ sentence pairs
Average Performance on Sentence Embeddings (14 diverse-tasks Datasets)	69.57	68.73	68.70	68.06

3.1.2 Aligners

We combined each of the four sentence embedding models with Itermax and Match matching algorithms and we produced eight versions of the sentence aligner. The aligners were evaluated by using the measures which are presented in the following section (3.1.3), to identify which of them achieved the best results and thus could better model simplification operations.

3.1.3 Evaluation Measures

We evaluated each sentence alignment extraction system by comparing the computed sentence alignments against a set of manually prepared gold-standard sentence alignments. The typically used measures for evaluating such systems are Precision, Recall, and F1-score [70]. Specifically, given a set of predicted alignment edges A_i and a set of gold standard alignment edges G_i for each text in a collection of comparable pairs of texts, we used the following evaluation measures:

$$micro\ Precision = \frac{|G_i \cap A_i| + |G_{i+1} \cap A_{i+1}|}{|A_i| + |A_{i+1}|} \qquad micro\ Recall = \frac{|G_i \cap A_i| + |G_{i+1} \cap A_{i+1}|}{|G_i| + |G_{i+1}|}$$

$$micro\ F\text{-}Measure = \frac{2PR}{P+R}$$

$$macro\ Precision = \frac{\sum_i^l \frac{|G_i \cap A_i|}{|A_i|}}{l} \qquad macro\ Recall = \frac{\sum_i^l \frac{|G_i \cap A_i|}{|G_i|}}{l}$$

$$macro\ F\text{-}Measure = \frac{2PR}{P+R}$$

$$weighted - macro\ Precision = \frac{|G_i| \sum_i^l \frac{|G_i \cap A_i|}{|A_i|}}{\sum_i^l |G_i|} \qquad weighted - macro\ Recall = \frac{|G_i| \sum_i^l \frac{|G_i \cap A_i|}{|G_i|}}{\sum_i^l |G_i|}$$

$$weighted - macro\ F\text{-}Measure = \frac{2PR}{P + R}$$

According to the formulas, the micro averaging method treats all samples in aggregate, by counting their successes. On the other hand, macro metrics calculate the performance of each sample individually and then take the average, giving equal importance to all samples regardless of the number of sentences they contain. In contrast, the weighted-macro averaging method calculates scores as a function of each sample's length, considering the contribution of each sample relative to its length.

3.1.4 Data

We conducted the evaluation using a manually sentence-aligned dataset that we created from a random sample of 200 texts pairs taken from the training set of the D-Wikipedia dataset (see section 1.2.2). The average length of source and target texts in terms of tokens and sentences is reported in Table 5. Our annotation scheme consisted of five simplification operations at sentence level which were also used in the D-Wikipedia dataset: Paraphrasing, Splitting, Insertion, Deletion, and Merging. The annotation process consisted of three stages:

1) Sentence Splitting: We used NLTK's sentence tokenizer [68] to split each complex-simple text pair into sentences and assigned them ascending indices [0, 1, ...]. We manually reviewed the sentence splits and made corrections as needed.

2) Semantic Overlap: We manually compared each complex sentence with each simple sentence to identify semantic overlaps. If any part of the information in a complex sentence was semantically covered by any part of a simple sentence, and vice versa, we matched their respective indices (e.g., [(0-2), (0-3)]).

3) Simplification Operation Tagging: We examined the resulting pairs and assigned tags for simplification operations. If both indices of a pair did not occur in any other pairs, it represented a paraphrasing example. If a specific first index appeared in multiple pairs, they were categorized as splitting instances (e.g., (1,1), (1,3), (1,4)). If the second index appeared in several pairs, they were labeled as merging instances (e.g., (1,4), (2,4), (3,4)). If a sentence in the complex text was not aligned, it was considered as deleted, and conversely, if a sentence in the simple text was not aligned, it was considered as inserted.

Table 6 illustrates an example of annotated texts from our dataset.

Table 5: Average Length of Texts in the Alignment Evaluation Dataset

	Number of sentences (Average)	Number of tokens (Average)
Source texts (complex)	5.26	135.32
Target texts (simple)	4.36	73.4

Table 6: A Manually Annotated Pair of Texts

Source	**Target**
Samuel Barclay Beckett (; 13 April 1906 – 22 December 1989) was an Irish novelist, playwright, short story writer theatre director, poet, and literary translator. [0]	Samuel Barclay Beckett (; 13 April 1906 – 22 December 1989) was born in Dublin, Ireland. [0]
A resident of Paris for most of his adult life, he wrote in both French and English. [1]	He was a writer of novels, plays, and poetry. [1]
Beckett 's work offers a bleak, tragi-comic outlook on human existence, often coupled with black comedy and gallows humor, and became increasingly minimalist in his later career. [2]	He also translated other famous works of literature. [2]
He is considered one of the last modernist writers, and one of the key figures in what Martin Esslin called the “Theatre of the Absurd”. [3]	He was given the Nobel Prize for Literature in 1969. [3]
His most well-known work is his 1953 play “Waiting for Godot”. [4]	His best-known play is “Waiting for Godot”. [4]
Beckett was awarded the 1969 Nobel prize in literature “for his writing, which—in new forms for the novel and drama—in the destitution of modern man acquires its elevation.” [5]	It has often been acted on stage and on TV. [5]
He was the first person to be elected Saoi of Aosdána in 1984. [6]	Beckett was stabbed in Paris in 1938. [6]
	He died of breathing problems in Paris in 1989. [7]
	Many writers of plays (playwrights) and

	others think he is one of the most important writers of the 20th century. [8]
	There have been many books written about him. [9]
	His books are often about people going through hard times and seeing life as both sad and funny. [10]
Paraphrasing: [(2,10), (4, 4), (5, 3)] Splitting: [(0, 0), (0, 1), (0, 2)] Insertion: [5, 6, 7, 8, 9] Deletion: [1, 3, 6] Merging: -	

For the same dataset of 200 pairs of texts, we had each aligner issue candidate alignments. The evaluation results are presented in the next section.

3.1.5 Results and Discussion

Table 7 presents the performance of the aligners for all operations: deletion, splitting, insertion, merging, paraphrasing.

Table 7: Results for All Simplification Operations

Aligner	**micro-P/R/F**	**macro- P/R/F**	**weighted macro-P/R/F**
all-distilroberta-v1 itermax	71.58/ 70.45/ 71.01	71.42/ 71.03/ 71.22	74.98/ 70.45/ 72.65
all-distilroberta-v1 match	63.78/ 62.56/ 63.17	64.99/ 67.67/ 66.30	63.13/ 62.56/ 62.84
all-mpnet-base-v2 itermax	**72.87/ 71.89/ 72.37**	**71.58/ 71.62/ 71.60**	**75.56/ 71.89/ 73.68**
all-mpnet-base-v2 match	64.86/ 63.63/ 64.24	65.59/ 68.45/ 66.99	63.92/ 63.63/ 63.77
all-MiniLM-L6-v2 itermax	70.14/ 69.20/ 69.67	69.79/ 69.79/ 69.79	72.99/ 69.20/ 71.05
all-MiniLM-L6-v2	64.01/ 62.80/ 63.40	65.14/ 68.05/ 66.56	63.20/ 62.80/ 63.00

match			
all-MiniLM-L12-v2 itermax	70.91/ 69.99/ 70.44	69.95/ 69.78/ 69.87	74.05/ 69.99/ 71.96
all-MiniLM-L12-v2 match	64.01/ 62.80/ 63.40	65.11/ 67.74/ 66.40	63.59/ 62.80/ 63.19

The results indicated that: a) the most effective configuration, as highlighted in bold, was all-mpnet-base-v2 with Itermax, b) the Itermax extraction algorithm consistently outperformed Match when employing the same sentence transformers model, c) the difference between Itermax-Match scores remained relatively consistent (e.g., 3-9% in F1), d) the alignment methods demonstrated comparable performance in terms of precision and recall across all settings, and e) there was no substantial difference observed among micro, macro, and weighted macro scores.

Table 8 presents the performance of the aligners for the paraphrasing operation (1<=>1 alignments).

Table 8: Results for Paraphrasing

Aligner	micro-P/R/F	macro- P/R/F	weighted macro-P/R/F
all-distilroberta-v1 itermax	59.88/ 72.28/ 65.50	59.50/ 64.07/ 61.70	69.72/ 72.28/ 70.98
all-distilroberta-v1 match	44.67/ 81.92/ 57.82	51.33/ 71.82/ 59.87	60.41/ 81.92/ 69.54
all-mpnet-base-v2 itermax	**62.37/ 74.69/ 67.98**	**61.21/ 67.00/ 63.97**	**70.35/ 74.69/ 72.46**
all-mpnet-base-v2 match	46.64/ 85.54/ 60.37	52.96/ 74.3/ 61.84	62.70/ 85.54/ 72.36
all-MiniLM-L6-v2 itermax	59.48/ 71.80/ 65.06	60.27/ 64.60/ 62.36	69.30/ 71.80/ 70.53
all-MiniLM-L6-v2 match	44.94/ 82.40/ 58.16	51.36/ 72.43/ 60.10	60.47/ 82.40/ 69.75
all-MiniLM-L12-v2	60.52/ 72.77/ 66.08	60.34/ 65.71/ 62.91	69.56/ 72.77/ 71.13

itermax			
all-MiniLM-L12-v2 match	44.80/ 82.16/ 57.99	51.40/ 71.90/ 59.95	60.62/ 82.16/ 69.77

The above results confirmed the superiority of all-mpnet-base-v2-Itermax configuration, in the category of 1<=>1 alignments as well. In all cases recall was substantially higher than precision. Nevertheless, the optimal setup, specifically the all-mpnet-base-v2-Itermax configuration, showcased the best balance between precision and recall. Lastly, it was noticeable that the weighted macro scores were the highest overall, surpassing the others by a significant margin, which probably means that the aligners performed better in computing 1<=>1 sentence pairs in longer texts.

Table 9 presents the performance of the aligners in terms of deletion operation (1<=>0 alignments).

Table 9: Results for Deletion

Aligner	**micro-P/R/F**	**macro- P/R/F**	**weighted macro-P/R/F**
all-distilroberta-v1 itermax	88.59/ 80.08/ 84.12	85.33/ 78.78/ 81.92	90.61/ 80.08/ 85.02
all-distilroberta-v1 match	84.00/ 67.51/ 74.86	71.25/ 64.48/ 67.70	78.25/ 67.51/ 72.48
all-mpnet-base-v2 itermax	**90.0/ 82.62/ 86.15**	**87.33/ 81.21/ 84.16**	**92.58/ 82.62/ 87.32**
all-mpnet-base-v2 match	84.88/ 68.22/ 75.64	71.83/ 65.12/ 68.31	78.97/ 68.22/ 73.20
all-MiniLM-L6-v2 itermax	88.75/ 80.22/ 84.27	86.52/ 79.81/ 83.03	91.86/ 80.22/ 85.65
all-MiniLM-L6-v2 match	84.53/ 67.93/ 75.33	70.88/ 64.35/ 67.46	78.51/ 67.93/ 72.84
all-MiniLM-L12-v2 itermax	88.81/ 80.79/ 84.61	85.95/ 79.47/ 82.58	91.49/ 80.79/ 85.81
all-MiniLM-L12-v2	84.18/ 67.65/ 75.01	71.24/ 64.32/ 67.60	78.54/ 67.65/ 72.69

match			

The model utilizing all-mpnet-base-v2-Itermax delivered the best results for 1<=>0 alignments as well. However, unlike 1<=>1 alignment, precision consistently surpassed recall. This indicates that most of the sentences which were labeled as deletions by the aligner were indeed deleted in the gold standard dataset. The results for all-mpnet-base-v2-Itermax showed that the text length had minimal impact on the model's accuracy in retrieving deletions. This makes it particularly suitable for obtaining reliable training data.

Table 10 presents the performance of the aligners for insertion operation (0<=>1 alignments).

Table 10: Results for Insertion

Aligner	**micro-P/R/F**	**macro- P/R/F**	**weighted macro-P/R/F**
all-distilroberta-v1 itermax	70.41/ 77.99/ 74.01	73.87/ 76.19/ 75.01	78.35/ 77.99/ 78.17
all-distilroberta-v1 match	65.45/ 61.61/ 63.47	62.86/ 62.58/ 62.72	67.81/ 61.61/ 64.56
all-mpnet-base-v2 itermax	**71.46/ 78.97/ 75.02**	**72.64/ 75.11/ 73.85**	**79.58/ 78.97/ 79.27**
all-mpnet-base-v2 match	66.23/ 62.34/ 64.23	63.68/ 63.35/ 63.51	68.42/ 62.34/ 65.24
all-MiniLM-L6-v2 itermax	67.61/ 75.55/ 71.36	69.48/ 72.85/ 71.12	74.39/ 75.55/ 74.97
all-MiniLM-L6-v2 match	65.45/ 61.61/ 63.47	63.98/ 63.97/ 63.97	67.70/ 61.61/ 64.51
all-MiniLM-L12-v2 itermax	69.86/ 78.23/ 73.81	72.42/ 74.85/ 73.61	78.41/ 78.23/ 78.32
all-MiniLM-L12-v2 match	66.23/ 62.34/ 64.23	63.43/ 62.99/ 63.21	69.54/ 62.34/ 65.74

Once again, the all-mpnet-base-v2-Itermax model achieved the best results in 0<=>1 alignments. Analysing the micro, macro, and weighted macro scores of all-mpnet-base-v2-Itermax, we deduced that the length of the text appeared to have a greater impact on precision rather than recall.

Table 11 presents the performance of the aligners in terms of splitting operation (1<=>n alignments).

Table 11: Results for Splitting

Aligner	micro-P/R/F	macro- P/R/F	weighted macro-P/R/F
all-distilroberta-v1 itermax	**66.66/ 46.23/ 54.60**	**65.62/ 63.05/ 64.31**	**54.70/ 46.23/ 50.11**
all-distilroberta-v1 match	57.99/ 31.18/ 40.55	58.0/ 58.0/ 58.0	31.18/ 31.18/ 31.18
all-mpnet-base-v2 itermax	62.68/ 45.16/ 52.5	62.45/ 60.35/ 61.38	51.38/ 45.16/ 48.07
all-mpnet-base-v2 match	57.99/ 31.18/ 40.55	58.0/ 58.0/ 58.0	31.18/ 31.18/ 31.18
all-MiniLM-L6-v2 itermax	60.83/ 43.01/ 50.39	61.31/ 59.83/ 60.56	47.32/ 43.01/ 45.06
all-MiniLM-L6-v2 match	57.99/ 31.18/ 40.55	58.0/ 58.0/ 58.0	31.18/ 31.18/ 31.18
all-MiniLM-L12-v2 itermax	61.74/ 43.81/ 51.25	61.33/ 59.38/ 60.34	49.71/ 43.81/ 46.58
all-MiniLM-L12-v2 match	57.99/ 31.18/ 40.55	58.0/ 58.0/ 58.0	31.18/ 31.18/ 31.18

The results obtained for this category exhibited notable discrepancies across the three metrics: micro, macro, and weighted macro. The all-distilroberta-v1-Itermax aligner emerged as the top performer, although its overall performance fell short compared to other operation categories. Importantly, the weighted scores for all models were particularly low, indicating that the effectiveness of computing splitting operations diminished in longer texts compared to shorter ones. Additionally, it was observed that the Match algorithm consistently produced identical outputs, suggesting its limited potential in modelling this operation, regardless of the quality of sentence embeddings.

Table 12 presents the performance of the aligners in terms of merging operation (n<=>1 alignments).

Table 12: Results for Merging

Aligner	micro-P/R/F	macro- P/R/F	weighted macro-P/R/F
all-distilroberta-v1 itermax	59.62/ 63.88/ 61.68	72.77/ 73.05/ 72.91	64.21/ 63.88/ 64.05
all-distilroberta-v1 match	**81.5/ 64.68/ 72.12**	**81.5/ 81.5/ 81.5**	**64.68/ 64.68/ 64.68**
all-mpnet-base-v2 itermax	63.07/ 65.07/ 64.06	74.29/ 74.43/ 74.36	65.54/ 65.07/ 65.31
all-mpnet-base-v2 match	**81.5/ 64.68/ 72.12**	**81.5/ 81.5/ 81.5**	**64.68/ 64.68/ 64.68**
all-MiniLM-L6-v2 itermax	59.02/ 62.30/ 60.61	71.37/ 71.85/ 71.61	61.70/ 62.30/ 62.00
all-MiniLM-L6-v2 match	**81.5/ 64.68/ 72.12**	**81.5/ 81.5/ 81.5**	**64.68/ 64.68/ 64.68**
all-MiniLM-L12-v2 itermax	57.79/ 60.31/ 59.02	69.70/ 69.50/ 69.60	61.27/ 60.31/ 60.79
all-MiniLM-L12-v2 match	**81.5/ 64.68/ 72.12**	**81.5/ 81.5/ 81.5**	**64.68/ 64.68/ 64.68**

The only case that the Match algorithm outperformed Itermax was merging operation. Once again, like the splitting operation, the performance of the underlying embeddings model had no influence on the results obtained from Match algorithm.

In a nutshell, our analysis provided valuable insights into the task of monolingual sentence alignment for document-level text simplification. Between the extraction methods, Itermax proved to be more suitable for modelling most of the ADTS-specific operations. We showed that, when using the Itermax algorithm, the performance of the aligner depends on the sentence embeddings model. Specifically, the all-mpnet-base-v2 and the all-distilroberta-v1, the top performing sentence transformers models, consistently achieved the top 2 highest scores across all evaluated operations. The combination of all-mpnet-base-v2 with Itermax delivered the most robust overall performance. Deletion emerged as the operation with the best results across all metrics, exhibiting a reasonable balance between precision and recall, and minimal variations

among micro, macro, and weighted macro scores. This suggests that the aligner effectively identified deletion instances regardless of the length of the text. The significance of these findings was further reinforced by the fact that deletion was the most frequent operation in both computed and gold-standard alignment instances (see Figure 1).

Figure 1: Frequency of Operations

3.2 Sentence Deletion-Based Simplification Experiments and Evaluation

To develop the SVM-based classifier we used scikit-learn framework [71]. We scaled all features values by using the StandardScaler object. From the relevant classification metrics module, we used Accuracy, Precision, Recall, F1, and AUC scores to evaluate all classifiers that were included in our experimental setup. For the feature extraction methods, we utilized NumPy, NLTK, Textstat modules, and bert-base-uncased via the HuggingFace Transformers library [69]. Bert-base-uncased was also used for developing a sentence deletion classifier. The fine-tuning of all BERT models was implemented with reference to the official notebooks that are provided by HuggingFace [72] and to a community notebook on fine-tuning ALBERT for sentence-pair classification [73]. Those models were implemented in Pytorch 73] and their training was run in mixed precision (with tensor autocasting for the forward pass) on the GPU infrastructure that is provided by Google Colab [75]. To evaluate simplifiers, we used the public code for BERTscore[3] and D-SARI[4] that is available on GitHub, and the "corpus_bleu" function that is provided by NLTK for calculating BLEU score for multiple sentences. We rescaled BERTscore values according to the baselines[5]-based method which was proposed by Zhang et al. [17] and we used the default model for English, i.e., RoBERTa-large, layer17.

3.2.1 Dataset for Training Sentence Deletion-Based Simplifiers

The conclusions that we drew from the aligner-related set of experiments were connected to the initial premise that deletion-based text simplification at document scale can be modeled autonomously. Deletion emerged as the most prevalent operation and exhibited the highest alignment results (micro F1: 86.15%, macro F1: 84.16%, weighted

[3] https://github.com/Tiiiger/bert_score

[4] https://github.com/RLSNLP/Document-level-text-simplification

[5] By using rescale_with_baseline=True

macro F1: 87.32%) among all operations, which means that it was the most well-encoded operation by the document-level aligner that we implemented. Building upon this foundation, we proceeded to automatically construct a new sentence deletion-based resource. This resource was created from a random sample of 10K pairs of complex-simple texts taken from the training portion of the "D-Wikipedia" dataset. The resulting dataset comprised over 100K sentences, each labeled as either deleted or not. The objective was to utilize this dataset for our classifier-related experiments, thereby assessing the broader applicability of the proposed methods in the context of document-level text simplification.

Specifically, to create the dataset, we first applied our best performing aligner, all-mpnet-base-v2-Itermax, on the aforementioned 10K pairs of texts. Then, for each pair of texts, we retrieved all source sentences that were tagged as deleted and we added them to the dataset with the label “0” (deleted). The remaining source sentences were assigned the label “1” (retained).[6] Table 13 displays the details of the dataset.

Table 13: Statistics of the Classification Dataset

Sentences (Total)	Word count per sentence (Average)	Label ‘0-deleted’ (Distribution)	Label ‘1-retained’ (Distribution)
100,049	25.33	58.5%	41.5%

The subsequent step involved shuffling and dividing the resource into training, validation, and test sets, with an allocation ratio of 80%, 10%, and 10% respectively. This task was approached as a knapsack problem [76]. Namely, we had to split the samples under a double constraint; we needed to ensure that label categories were evenly distributed according to the initial distribution of the data (58.5% - 41.5%) and that all splits included mutually exclusive data points. Specifically, we aimed at grouping all sentences from a given source text into the same dataset bucket, so that the model would be validated and tested on sentences from texts with entirely unknown content. To achieve that, we employed a custom loss function previously developed for a similar Stratified Group Splitting task [77]. This loss function’s inputs were the average squared difference in the percentage representation of each text compared to the entire dataset, and the squared difference between the proportional length of the dataset compared to the target-ratio (80:10:10). A weighting factor, ranging from 0 to 1, was introduced to control the tradeoff between split ratio and stratification. We calibrated this weight to 0.1. We iterated through each text (group of sentences) and assigned it to the appropriate dataset (training, validation, or test) relying on the highest weighted sum of losses. Table 14 shows the details of the three splits.

[6] Each sentence in the dataset inherited the ID number of the source text it originated from. This means that sentences from the same complex text were assigned the same ID. Additionally, each sentence also inherited the ID number of the target (simple) text it corresponds to.

Table 14: Statistics of the Classification Dataset (per split)

Split	Sentences (Total)	Word count per sentence (Average)	Label '0-deleted' (Distribution)	Label '1-retained' (Distribution)
Train	80,199	25.27	58.47%	41.53%
Validation	9,931	25.39	58.48%	41.52%
Test	9,919	25.77	58.48%	41.52%

Guided by the id numbers of the target (simple) text, we also collected all simple text counterparts for the test split, to build a test set and evaluate our simplifiers at document level. Table 15 shows the relevant statistics.

Table 15: Statistics of the Simplification Dataset (Test Split)

	Simple Texts (Total)	Word count per text (Average)	Number of sentences per text (Average)
Test	1721	77.49	5.38

3.2.2 Classifier Experiments, Results, and Discussion

3.2.2.1 SVM-Based Classifier

The SVM-based classifier with Radial Basis Function (RBF) kernel was trained on the training part of the dataset and was optimized by cross-validation on cross-entropy loss which was done using the GridSearchCV object. For hyperparameter tuning we selected the following range of values: C: [0.1, 0.5, 1, 1.5, 2, 5, 10, 20, 50, 100], gamma: [5, 1, 1e-1, 2e-1, 1e-2, 5e-2, 1e-3, 5e-3, 1e-4, 5e-4]. The best values were C = 1, gamma = 1e-1.

We tested the SVM-based classifier on the test part of the dataset. All relevant results are reported on Table 16. Considering the class imbalance in our dataset, we experimented with threshold moving to find the optimal trade-off between the true-positive and the false-positive rates, i.e., sensitivity and specificity, respectively. To that end, we calculated the Receiver Operating Characteristic curve (ROC) (see Figure 3) and we selected the threshold with the largest geometric mean (G-mean) score. Geometric mean equals to the root of the product of class-wise sensitivity, that, if optimized, can be a good indicator of balance between the sensitivity and the specificity of an estimator [78].

3.2.2.2 BERT-Based Classifier

The BERT-based classifier was fine-tuned on the same training dataset as above. To prepare the text sequences for encoding, we needed to determine the maximum sentence length for padding/truncating. Thus, we first applied BERT WordPiece tokenizer to all three splits of the dataset, and we found that, after adding special tokens [CLS] and [SEP], the minimum length that was required to include all subtokens in 95% of the sentences was 61 (see Figure 2). This length was utilized in the subsequent experiments. We fed the encoded input sequence into BERT base and the [CLS] token representation of the sequence to the top classification layer. We used AdamW optimizer with weight decay of 1e-2, learning rate warmup over the first 50 steps, linear decay of the learning rate, and a gradient accumulation of 2. We used the default dropout probability of 0.1 for all 12 pre-trained layers, and we added the same dropout to the final classification layer as well. We tuned batch size, learning rate and epochs on validation set, according to the range that was proposed by the authors of BERT paper [24], i.e., batch size = [16, 32], learning rate: [5e-5, 3e-5, 2e-5], number of epochs: [2, 3, 4]. We found that a batch size of 16 and a learning rate of 3e-5 returned the lowest loss score on the validation dataset after 2 epochs of fine-tuning.

We tested our BERT-based classifier on the same test set as above. All relevant results are reported on Table 16. We also tuned the threshold for positive-negative class labels as above. Figure 3 illustrates the ROC curve.

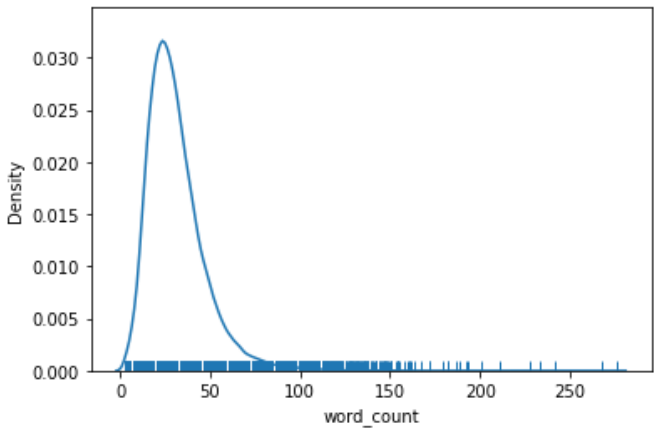

Figure 2: Variation in Word Count Distribution

Table 16: Classification Results

	Precision	Recall	F1	G-mean	Best Threshold
SVM-based Classifier	**70.71**	**70.88**	**70.78**	0.708	0.551
BERT-based Classifier	70.38	70.07	70.20	0.697	0.455

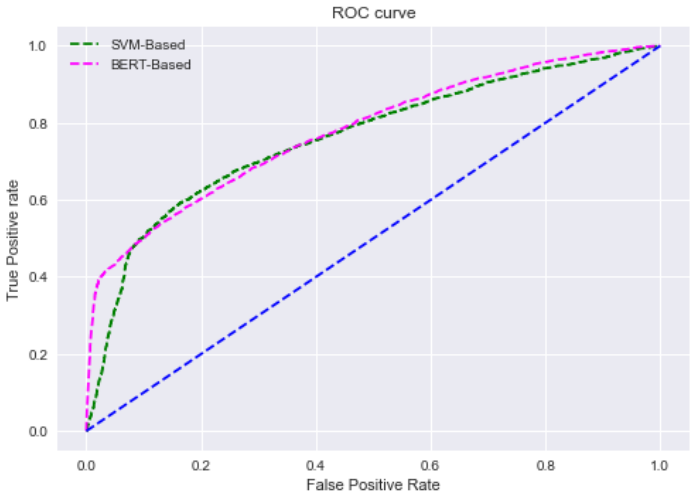

Figure 3: ROC Curve

The results presented in Table 16 demonstrate that the BERT-based classifier and the SVM-based classifier, despite their dissimilarities, achieved similar performance. Remarkably, although the BERT-based classifier encoded each sentence independently of its context (i.e., the whole document), it produced nearly identical results to the classical ML algorithm that incorporated document-aware features. Considering that our dataset was automatically created and exhibited class imbalance, these findings further support the effectiveness of the proposed alignment and tagging methods for directly predicting sentence deletions in text simplification.

3.2.2.3 BERT-Induced Feature Extraction

We applied the all-mpnet-base-v2-Itermax aligner and our tagging method on a randomly selected sample of 5,000 text pairs from the training part of the “D-Wikipedia” Dataset to build the pair relatedness training set; this sample of texts was different than the one we used to build the dataset for training the sentence deletion classifiers. Then, we further processed the data by taking the following two steps:

- For every source text find the most informative sentence, using the Sentence-to-Sentence Cohesion function.
- Pair that sentence with each of the remaining sentences in the source text and assign to each pair the label “related” or “not related”. If the second sentence of the pair is tagged as “deleted” by the aligner, then the label “not related” is assigned to the pair, else “related”.

Likewise, we built the validation and test splits, out of 400 texts pairs from the validation and test sets of the “D-Wikipedia” Dataset, respectively. Table 17 shows the details of the dataset.

Table 17: Statistics of the Sentence Pair Relatedness Dataset (per split)

Split	Pair of Sentences (Total)	Label ‘not related’ (Distribution)	Label ‘related’ (Distribution)
Train	15,000	48%	52%
Validation	1,700	48%	52%
Test	1,700	48%	52%

We fine-tuned ‘bert-base-uncased’ on the train split for the task of sentence pair classification. The pair was treated as a single input which was separated by the [SEP] token ([CLS]sent1 tokens [SEP]sent2 tokens). We used the validation split for hyperparameters tuning, as above. We found that maximum length of 128 tokens, a batch size of 64 and a learning rate of 5e-5 returned the lowest loss score on the validation dataset after 1 epoch of fine-tuning. We tested the fine-tuned model on the test set, and we obtained the results that are reported in Table 18.

Table 18: Sentence Pair Relatedness Results

	Precision	Recall	F1
Sentence Pair Classifier	68.34	68.01	68.34

3.2.3 Simplification Experiments, Results, and Discussion

To generate the simplified texts, we used all sentences that were assigned the label “retained” by the respective classifier. This allowed us to obtain two sets of automatically simplified texts, one from the SVM-based classifier and another from the BERT-based classifier. We evaluated each set of automatically generated simplifications against the test set of references (simple texts). Table 19 shows the simplification results.

Table 19: Simplification Results

Simplifier	Word count per text (Average)	D-SARI (F1)	BLEU	BertScore _F1	BertScore _Precision	BertScore _Recall
SVM-based	56.69	21.21	9.391	16.9	15.5	18.8
BERT-based	51.65	25.64	8.899	17	17.7	16.8

Oracle	54.61	28.25	10.58	21.4	21.2	21.1

To interpret the simplification results, we also implemented an oracle simplifier. This involved generating a set of simplified texts relying on the labels for the sentences in the test set, as predicted by the aligner. The oracle simplifier serves as a rough approximation of the upper bound for sentence-deletion-dependent simplification on the dataset. We used the same evaluation measures for both the oracle simplifier and the other simplification methods. The results of the evaluation are presented in Table 19.

Prior to evaluating the simplifiers, we conducted a preliminary analysis to assess the applicability of a deletion-only approach in document-level text simplification and to evaluate the effectiveness of the aligner. For this purpose, we developed six baseline models (Lead-1, Lead-2, Lead-3, Lead-4, Lead-5, All-sentences) that generated target texts by sequentially copying sentences from the source text. Each baseline progressively included more sentences from the source text.

To evaluate the baselines, we calculated their respective scores and plotted the results against the average word count for each metric. Additionally, we included the performance of the oracle simplifier in the plot for comparison. As depicted in Figures [], the oracle simplifier consistently outperformed the other baselines. These findings provide compelling evidence for the effectiveness of sentence-deletion-focused simplification as a viable approach.

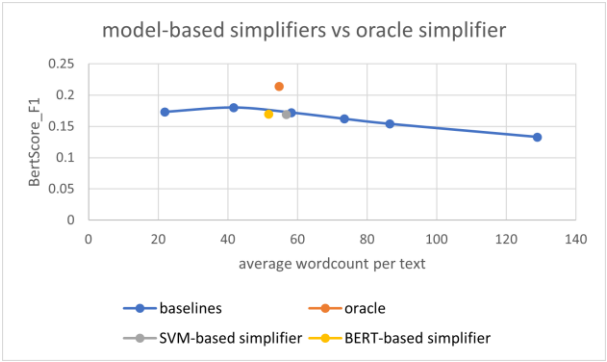

Figure 4: BertScore F1

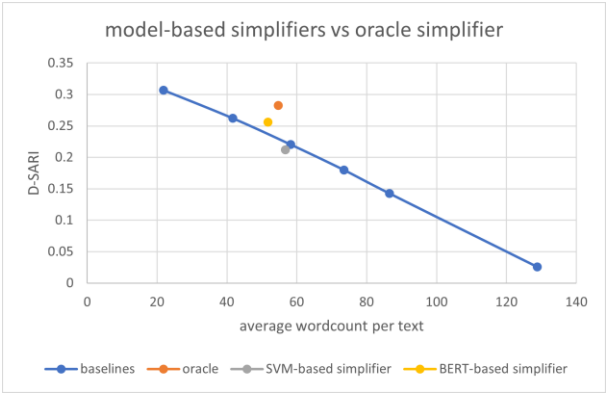

Figure 5: D-SARI

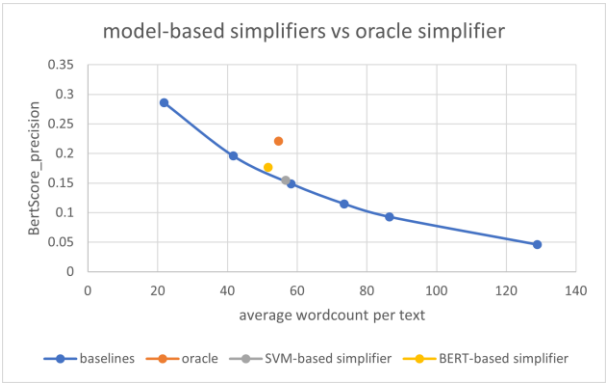

Figure 6: BertScore Precision

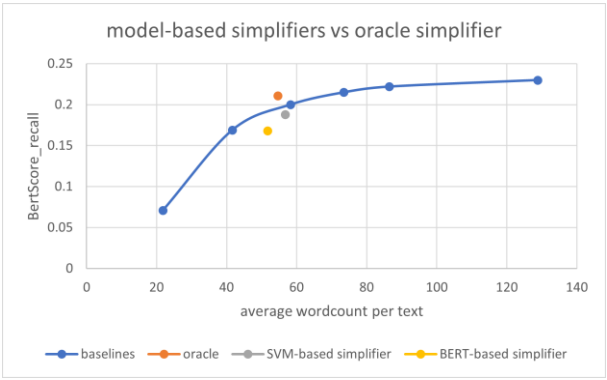

Figure 7: BertScore Recall

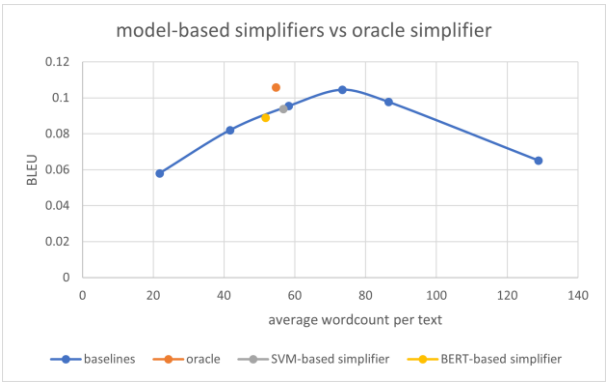

Figure 8: BLEU

Upon examining the plots above (Figures 5, 6, 7, 8, 9), it becomes apparent that in all metrics the system's performance was influenced by the length of the texts. It is worth mentioning that some researchers [11], [23], [27], [79], [80] have raised concerns regarding the reliability of these metrics, particularly in cases where only a single set of references is available. Another limitation was the absence of research analyzing the correlation between NLG evaluation metrics and document-level simplification.

Upon examination of the graphs in Figure 5 and Figure 9, it is evident that the performance of both classifier-based simplifiers aligned closely with the performance of the baselines. This finding underscores the challenge of accurately distinguishing between two systems with similar performance using metrics such as BLEU and BertScore. However, our analysis of the D-SARI plot (Figure []) indicated that the BERT-based simplifier outperformed the SVM-based simplifier, despite that the respective classifiers achieved similar performance. This discovery holds significance considering that D-SARI was tailored to cater to the evaluation requirements of document-level simplification.

Essentially, the results provided evidence for the superiority of the BERT-based classifier in predicting sentences that lead to simpler texts. This can be attributed to the rich knowledge that is encoded by BERT's contextual embeddings. To further investigate this trend and assess the impact of the precision-recall trade-off on the model's ability to generate simpler texts, we conducted an evaluation of the BERT-based classifier's performance at different decision thresholds (0.35, 0.37, 0.4, 0.42, 0.45, 0.47, 0.5, 0.52, 0.55, 0.57). From the graph in Figure 10, it is evident that

increasing the decision threshold resulted in a greater deviation from the performance line of the baselines. Notably, the performance of the simplifier was not negatively affected by adjusting the threshold, indicating robustness to threshold variations. This finding highlights the classifier's ability to maintain consistent performance across different decision thresholds while effectively generating simpler texts.

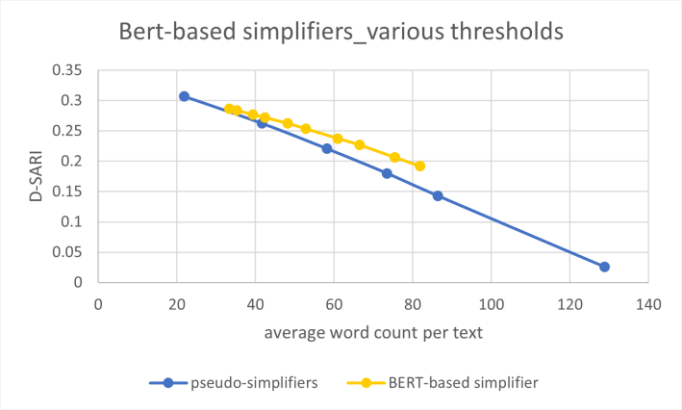

Figure 9: D-SARI (BERT-based simplifier at different thresholds)

4 CONCLUSIONS, LIMITATIONS AND FUTURE WORK

The experiments and analysis of the ADTS-related operation distribution that we conducted in this study demonstrate that sentence deletion serves as a fundamental mechanism in generating simpler texts. Moreover, the automatic generation of aligned data through unsupervised methods demonstrates its potential as a valuable resource for training models, including sentence deletion classifiers. This approach proves to be both feasible and effective in facilitating the development of robust systems for document-level text simplification. As anticipated, the utilization of large language models such as BERT has shown to boost the performance of simplification systems, a trend observed in various other NLP tasks.

However, there are certain limitations to this thesis that should be acknowledged. Firstly, the study examines the deletion-driven approach and does not extensively explore other operations. While sentence deletion is indeed a key instrument in simplifying texts, there is still much to be explored in terms of how other operations can contribute to the overall simplification process. By expanding the framework to include other operations, such as sentence insertion, merging, splitting, and paraphrasing, a more comprehensive and versatile document-level simplification system can be developed.

Additionally, the evaluation and performance analysis of the simplification systems in this thesis rely on existing evaluation measures, such as D-SARI, BERTScore, and BLEU, which have their own limitations. These metrics are influenced by the length of the texts and may not fully capture the complexities and nuances of document-level text simplification, particularly when it comes to assessing the overall quality and readability of the simplified texts. Therefore, there is a need for the development of new evaluation metrics that specifically address the unique challenges and requirements of document-level text simplification.

Furthermore, the thesis focuses on the English language, and the findings and methods presented may not directly apply to other languages without further investigation and adaptation. Language-specific characteristics and linguistic variations may require tailored approaches for effective document-level text simplification in different languages.

In conclusion, while this thesis makes significant contributions to the field of document-level text simplification, further research is needed to explore new methods and additional simplification operations, develop more suitable evaluation metrics, and extend the approach to other languages. These areas of research will contribute to the advancement and practical application of document-level text simplification techniques in various domains and languages.

ACRONYMS

ATS	Automatic Text Simplification
ASTS	Automatic Sentence-level Text Simplification
ADTS	Automatic Document-level Text Simplification
IR	Information Retrieval
NLI	Natural Language Inference
QA	Question Answering
NLP	Natural Language Processing
BLEU	BiLingual Evaluation Understudy
MT	Machine Translation
SARI	System output Against References and against the Input sentence
D-SARI	Document-level System output Against References and against the Input sentence
BERT	Bidirectional Encoder Representations from Transformers
SOTA	State-of-the-Art
GPT	Generative Pre-trained Transformer
Bi-LSTM	Bidirectional Long Short-Term Memory
RL	Reinforcement Learning
SCST	Self-Critical Sequence Training
NN	Neural Network
SBERT	Sentence-BERT
GloVe	Global Vectors (for Word Representation)
SVM	Support Vector Machines
TF-ISF	Term Frequency-Inverse Sentence Frequency
AUC	Area Under the ROC Curve
ALBERT	A Lite BERT
RoBERTa	Robustly optimized BERT pre-training approach
RBF	Radial Basis Function
ROC	Receiver Operating Characteristic
NLG	Natural Language Generation

REFERENCES

[1] F. Alva Manchego, “Automatic Sentence Simplification with Multiple Rewriting Transformations,” Ph.D. dissertation, Dept. Comp. Sci., Fac. of Sci., The Univ. of Sheffield, 2020. [Online]. Available: https://etheses.whiterose.ac.uk/28690/.B. Zhang.

[2] B. Zhang, P.K. Choubey, and R. Huang, “Predicting Sentence Deletions for Text Simplification Using a Functional Discourse Structure,” in *Proc. of the 60th Annu. Meeting of the ACL*, vol. 2: Short Papers, 2022, pp. 255–261. [Online]. Available: https://aclanthology.org/2022.acl-short.28.pdf.

[3] W. Xu, C. Napoles, E. Pavlick, Q. Chen, and C. Callison-Burch, “Optimizing Statistical Machine Translation for Text Simplification,” Transactions of the Association for Computational Linguistics, vol. 4, 2016, pp. 401–415. [Online]. Available: https://aclanthology.org/Q16-1029.pdf.

[4] R. Sun, H. Jin, and X. Wan, “Document-Level Text Simplification: Dataset, Criteria and Baseline,” 2021. [Online]. Available: arXiv:2110.05071.

[5] F. Alva-Manchego, C. Scarton, and L. Specia, “Cross-Sentence Transformations in Text Simplification,” in *Proc. of the 2019 Workshop on Widening NLP*, 2019, pp. 181–184, [Online]. Available: https://aclanthology.org/W19-3656/.

[6] W. Zhao, M. Peyrard, F. Liu, Y. Gao, C. Meyer, and S. Eger, “MoverScore: Text Generation Evaluating with Contextualized Embeddings and Earth Mover Distance,” 2019. [Online]. Available: arXiv:1909.02622.

[7] T. Scialom, P.-A. Dray, S. Lamprier, B. Piwowarski, and J. Staiano, “Discriminative Adversarial Search for Abstractive Summarization,” 2020. [Online]. Available: arXiv:2002.10375.

[8] N. Grabar and H. Saggion, “Evaluation of Automatic Text Simplification: Where are we now, where should we go from here,” in *Proc. of the 29th Conference on Natural Language Processing,* vol. 1, 2022, pp. 453-463. [Online]. Available: https://aclanthology.org/2022.jeptalnrecital-taln.47.pdf.

[9] A. Siddharthan and A. Mandya, “Hybrid text simplification using synchronous dependency grammars with hand-written and automatically harvested rules,” in *Proc. of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, 2014, pp. 722–731. [Online]. Available: https://aclanthology.org/E14-1076.pdf.

[10] F. Alva-Manchego, C. Scarton, and L. Specia, “Data-Driven Sentence Simplification: Survey and Benchmark,” *Computational Linguistics*, 2020, pp. 135–187. [Online]. Available: https://aclanthology.org/2020.cl-1.4.pdf.

[11] F. Alva-Manchego, C. Scarton, and L. Specia, “The (Un)Suitability of Automatic Evaluation Metrics for Text Simplification,” *Computational Linguistics*, 2021, pp. 861-889. [Online]. Available: https://aclanthology.org/2021.cl-4.28.pdf.

[12] S. Štajner, M. Popović, H. Saggion, L. Specia and M. Fishel, “Shared Task on Quality Assessment for Text Simplification,” in *Proc. of the Workshop on Quality Assessment for Text Simplification (QATS)*, 2016, pp. 22-31.

[13] S. Nisioi, S. Štajner, S. P. Ponzetto, and L. P. Dinu, “Exploring Neural Text Simplification Models,” in *Proc. of the 55th Annual Meeting of the Association for Computational Linguistics*, vol. 2: Short Papers, 2017, pp. 85-91. [Online]. Available: https://aclanthology.org/P17-2014.pdf.

[14] T. Scialom, L. Martin, J. Staiano, É. Villemonte de la Clergerie, and B. Sagot, “Rethinking Automatic Evaluation in Sentence Simplification,” 2021. [Online]. Available: arXiv: 2104.07560.

[15] W. Xu, C. Napoles, E. Pavlick, Q. Chen, and C. Callison-Burch, “Optimizing Statistical Machine Translation for Text Simplification,” *Transactions of the Association for Computational Linguistics*, vol. 4, 2016, pp. 401–415. [Online]. Available: https://aclanthology.org/Q16-1029.pdf.

[16] K. Papineni, S. Roukos, T. Ward, and W.J. Zhu, "Bleu: A Method for Automatic Evaluation of Machine Translation," in *Proc. of the 40th Annu. Meeting of the ACL*, 2002, pp. 311-318. [Online]. Available: https://aclanthology.org/P02-1040.pdf.

[17] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, “BERTScore: Evaluating Text Generation with BERT,” 2019. [Online]. Available: arXiv:1904.09675.

[18] S.S. Al-Thanyyan and A.M. Azmi, “Automated Text Simplification: A Survey,” ACM Computing Surveys, vol. 54, article 43, 2021. [Online]. Available: https://doi.org/10.1145/3442695.

[19] F. Alva-Manchego, L. Martin, C. Scarton, and L. Specia, “EASSE: Easier Automatic Sentence Simplification Evaluation,” 2019. [Online]. Available: arXiv:1908.04567.

[20] S. Wubben, A. van den Bosch, and E. Krahmer, “Sentence Simplification by Monolingual Machine Translation,” in *Proc. of the 50th Annu. Meeting of the ACL*, vol. 1: Long Papers, 2012, pp. 1015–1024. [Online]. Available: https://aclanthology.org/P12-1107.pdf.

[21] S. Štajner, R. Mitkov, and H. Saggion, “One Step Closer to Automatic Evaluation of Text Simplification Systems,” in *Proc of the 3rd Workshop on Predicting and Improving Text Readability for Target Reader Populations (PITR)*, 2014, pp. 1–10. [Online]. Available: https://aclanthology.org/W14-1201.pdf.

[22] F. Alva-Manchego, L. Martin, A. Bordes, C. Scarton, B, Sagot, and L. Specia, “ASSET: A Dataset for Tuning and Evaluation of Sentence Simplification Models with Multiple Rewriting Transformations,” 2020. [Online]. Available: arXiv:2005.00481.
[23] E. Sulem, O. Abend, and A. Rappoport, “BLEU is Not Suitable for the Evaluation of Text Simplification,” 2018. [Online]. Available: arXiv:1810.05995.
[24] J. Devlin, MW. Chang, K. Lee, K. Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” 2018. [Online]. Available: arXiv:1810.04805.
[25] Z. Lin, Y. Cai, and X. Wan, “Towards Document-Level Paraphrase Generation with Sentence Rewriting and Reordering,” 2021. [Online]. Available: arXiv:2109.07095.
[26] R. Cardon and N. Grabar, “French Biomedical Text Simplification: When Small and Precise Helps,” in in *Proc. of the 28th International Conference on Computational Linguistics*, 2020, pp. 710–716. [Online]. Available: https://aclanthology.org/2020.coling-main.62.pdf.
[27] T. Tanprasert and D. Kauchak, “Flesch-Kincaid is Not a Text Simplification Evaluation Metric,” in *Proc. of the 1st Workshop on Natural Language Generation, Evaluation, and Metrics (GEM 2021)*, 2021, pp. 1-14. [Online]. Available: https://aclanthology.org/2021.gem-1.1.pdf.
[28] Z. Zhu, D. Bernhard, and I. Gurevych, “A Monolingual Tree-based Translation Model for Sentence Simplification,” in *Proc. of the 23rd International Conference on Computational Linguistics*, 2010, pp. 1353–1361. [Online]. Available: https://aclanthology.org/C10-1152.pdf.
[29] W. Coster and D. Kauchak, “Simple English Wikipedia: A New Text Simplification Task,” in *Proc. of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, 2011, pp. 665-669. [Online]. Available: https://aclanthology.org/P11-2117.pdf.
[30] W. Hwang, H. Hajishirzi, M. Ostendorf, and W. Wu, “Aligning Sentences from Standard Wikipedia to Simple Wikipedia,” in *Proc. of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2015, pp. 211–217. [Online]. Available: https://aclanthology.org/N15-1022.pdf.
[31] T. Kajiwara and M. Komachi, “Building a Monolingual Parallel Corpus for Text Simplification Using Sentence Similarity Based on Alignment between Word Embeddings,” in *Proc. of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, 2016, pp. 1147–1158. [Online]. Available: https://aclanthology.org/C16-1109.pdf.
[32] X. Zhang and M. Lapata, “Sentence Simplification with Deep Reinforcement Learning,” 2017. [Online]. Available: arXiv:1703.10931.
[33] W. Xu, C. Callison-Burch, and C. Napoles, “Problems in Current Text Simplification Research: New Data Can Help,” *Transactions of the Association for Computational Linguistics*, 2015, pp. 283–297. [Online]. Available: https://aclanthology.org/Q15-1021.pdf.
[34] C. Jiang, M. Maddela, W. Lan, Y. Zhong, and W. Xu, “Neural CRF Model for Sentence Alignment in Text Simplification,” 2021. [Online]. Available: arXiv:2005.02324.
[35] Y. Dong, Z. Li, M. Rezagholizadeh, and J.C.K. Cheung, “EditNTS: An Neural Programmer-Interpreter Model for Sentence Simplification through Explicit Editing,” 2019. [Online]. Available: arXiv:1906.08104.
[36] F. Alva-Manchego, J. Bingel, G. Paetzold, C. Scarton, and L. Specia, “Learning How to Simplify From Explicit Labeling of Complex-Simplified Text Pairs,” in *Proc. of the Eighth International Joint Conference on Natural Language Processing*, vol. 1: Long Papers, 2017, pp. 295–305. [Online]. Available: https://aclanthology.org/I17-1030.pdf.
[37] L. Vásquez-Rodríguez, M. Shardlow, P, Przybyła, and S. Ananiadou, “The Role of Text Simplification Operations in Evaluation,” in *Proc. of the First Workshop on Current Trends in Text Simplification*, 2021, pp. 57-69. [Online]. Available: https://ceur-ws.org/Vol-2944/paper4.pdf.
[38] N.P. Srikanth, “Characterizing content addition and explanation generation in document-level text simplification,” M.S. thesis, The Univ. of Texas at Austin, 2020. [Online]. Available: https://repositories.lib.utexas.edu/handle/2152/89500.
[39] A. Vaswani et al., “Attention Is All You Need,” 2017. [Online]. Available: arXiv:1706.03762.
[40] M. Lewis et al., “BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension,” 2019. [Online]. Available: arXiv:1910.13461.
[41] R. Sun, Z. Lin, and X. Wan, “On the Helpfulness of Document Context to Sentence Simplification,” in *Proc. of the 28th International Conference on Computational Linguistics*, 2020, pp. 1411–1423. [Online]. Available: https://aclanthology.org/2020.coling-main.121.pdf.
[42] Y. Liu and M. Lapata, “Text Summarization with Pretrained Encoders,” 2019. [Online]. Available: arXiv:1908.08345.
[43] J.P. Kincaid, R.P. Fishburne Jr., R.L. Rogers, and B.S. Chissom, “Derivation of New Readability Formulas (Automated Readability Index, Fog Count And Flesch Reading Ease Formula) For Navy Enlisted Personnel,” 1975. [Online]. Available: https://stars.library.ucf.edu/istlibrary/56.
[44] N. Srikanth and J.J. Li, “Elaborative Simplification: Content Addition and Explanation Generation in Text Simplification,” 2021. [Online]. Available: arXiv:2010.10035.

[45] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, “Language models are unsupervised multitask learners,” OpenAI blog, 2019. [Online]. Available: https://d4mucfpksywv.cloudfront.net/better-language-models/language_models_are_unsupervised_multitask_learners.pdf.
[46] Y. Zhong, C. Jiang, W. Xu, and J.J. Li, “Discourse Level Factors for Sentence Deletion in Text Simplification,” 2019. [Online]. Available: arXiv:1911.10384.
[47] S. Štajner, M. Franco-Salvador, P. Rosso, and S.P. Ponzetto, “CATS: A Tool for Customized Alignment of Text Simplification Corpora,” in *Proc. of the Eleventh International Conference on Language Resources and Evaluation*, 2018. [Online]. Available: https://aclanthology.org/L18-1615.pdf.
[48] P.K. Choubey, A. Lee, R. Huang, and L. Wang, “Discourse as a Function of Event: Profiling Discourse Structure in News Articles around the Main Event,” in *Proc. of the 58th Annu. Meeting of the ACL,* 2020, pp. 5374–5386. [Online]. Available: https://aclanthology.org/2020.acl-main.478.pdf.
[49] A. Devaraj, I.J. Marshall, B.C. Wallace, and J.J. Li, “Paragraph-level Simplification of Medical Texts,” 2021. [Online]. Available: arXiv:2104.05767.
[50] I. Beltagy, K. Lo, and A. Cohan, “SciBERT: A Pretrained Language Model for Scientific Text,” 2019. [Online]. Available: arXiv:1903.10676.
[51] A. Phatak, D.W. Savage, R. Ohle, J. Smith, and V. Mago, “Medical Text Simplification Using Reinforcement Learning (TESLEA): Deep Learning–Based Text Simplification Approach,” *JMIR Medical Informatics*, 2022. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9719064/.
[52] S.J. Rennie, E. Marcheret, Y. Mroueh, J. Ross, and V. Goel, “Self-critical Sequence Training for Image Captioning,” 2016. [Online]. Available: arXiv:1612.00563.
[53] P. Laban, T. Schnabel, P. Bennett, and M.A. Hearst, “Keep It Simple: Unsupervised Simplification of Multi-Paragraph Text,” *in Proc. of the 59th Ann. Meeting of the ACL and the 11th International Joint Conference on NLP*, vol. 1: Long Papers, 2021, pp. 6365–6378. [Online]. Available: https://aclanthology.org/2021.acl-long.498.pdf.
[54] B. Drndarevic and H. Saggion, “Reducing text complexity through automatic lexical simplification: an empirical study for Spanish,” *Procesamiento del lenguaje natural 49*, 2012, pp. 13-20. [Online]. Available: https://www.redalyc.org/pdf/5157/515751749001.pdf.
[55] K. Woodsend and M. Lapata, “Learning to Simplify Sentences with Quasi-Synchronous Grammar and Integer Programming,” in *Proc. of the 2011 Conference on Empirical Methods in NLP*, 2011, pp. 409–420. [Online]. Available: https://aclanthology.org/D11-1038.pdf.
[56] A. Smolka, H.M. Wang, J.S. Chang, and K.Y. Su, “Is Character Trigram Overlapping Ratio Still the Best Similarity Measure for Aligning Sentences in a Paraphrased Corpus?,” in *Proc. of the 34th Conference on Computational Linguistics and Speech Processing*, 2022, pp. 49–60. [Online]. Available: https://aclanthology.org/2022.rocling-1.7.pdf.
[57] G. Paetzold, F. Alva-Manchego, and L. Specia, “MASSAlign: Alignment and Annotation of Comparable Documents,” in *Proc. Proceedings of the IJCNLP 2017, System Demonstrations*, 2017, pp. 1-4. [Online]. Available: https://aclanthology.org/I17-3001.pdf.
[58] M.J. Sabet, P. Dufter, F. Yvon, and H. Schütze, “SimAlign: High Quality Word Alignments without Parallel Training Data using Static and Contextualized Embeddings,” 2021. [Online]. Available: arXiv:2004.08728.
[59] SentenceTransformers Documentation. https://www.sbert.net/# (accessed Feb. 25, 2023).
[60] N. Reimers and I. Gurevych, “Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks,” 2019. [Online]. Available: arXiv:1908.10084.
[61] J. Pennington, R. Socher, C.D. Manning, “Glove: Global vectors for word representation,” in *Proc. of the 2014 Conference on EMNLPI,* 2014, pp 1532-1543. [Online]. Available: https://aclanthology.org/D14-1162.pdf.
[62] M.A. Hearst, S.T. Dumais, E. Osuna, J. Platt, and B. Scholkopf, "Support vector machines," in *IEEE Intelligent Systems and their Applications*, vol. 13, no. 4, pp. 18-28, 1998, doi: 10.1109/5254.708428.
[63] M.A. Fattah and F. Ren, “Automatic text summarization”, *World Academy of Science, Engineering and Technology*, 2008. [Online]. Available: https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=dd7ef1aecb0c5f6a41c317bae62099e3cec2f0ea
[64] T. Hirao, H. Isozaki, E. Maeda, and Y. Matsumoto, “. Extracting important sentences with support vector machines,” *in Proc. of the 19th International Conference on Computational Linguistics*, 2002. [Online]. Available: https://aclanthology.org/C02-1053.pdf.
[65] J. Larocca Neto, A.A. Freitas, and C.A.A. Kaestner, “Automatic text summarization using a machine learning approach,” *in Brazilian Symposium on Artificial Intelligence*, 2002, pp. 205-215. [Online]. Available: https://link.springer.com/chapter/10.1007/3-540-36127-8_20.

[66] R. Ferreira et al., “Assessing sentence scoring techniques for extractive text summarization,” *Expert systems with applications*, 2013, pp. 5755-5764. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S0957417413002601.
[67] E. Dale and J.S. Chall, “A Formula for Predicting Readability: Instructions,” *Educational research bulletin*, pp. 37-54, 1948.
[68] NLTK Documentation. https://www.nltk.org/api/nltk.tokenize.html#nltk-tokenize-package (accessed Sept. 18, 2022).
[69] https://huggingface.co/sentence-transformers (accessed Oct. 20, 2022).
[70] R. Mihalcea and Ted Pedersen, “An Evaluation Exercise for Word Alignment,” in *Proc. of the HLT-NAACL 2003 Workshop on Building and Using Parallel Texts: Data Driven Machine Translation and Beyond*, 2003, pp. 1-10. [Online]. Available: https://aclanthology.org/W03-0301.pdf.
[71] F. Pedregosa et al., “Scikit-learn: Machine Learning in Python,” 2012. [Online]. Available: arXiv:1201.0490
[72] https://huggingface.co/docs/transformers/notebooks (accessed Dec. 10, 2022).
[73] https://github.com/NadirEM/nlp-notebooks/blob/master/Fine_tune_ALBERT_sentence_pair_classification.ipynb (accessed Dec. 10, 2022).
[74] A. Paszke et al., “PyTorch: An Imperative Style, High-Performance Deep Learning Library,” 2019. [Online]. Available: arXiv:1912.01703.
[75] https://colab.research.google.com (accessed Dec. 10, 2022).
[76] https://www.geeksforgeeks.org/introduction-to-knapsack-problem-its-types-and-how-to-solve-them/ (accessed Jan. 10, 2023).
[77] amin_nejad. Stack Overflow. https://stackoverflow.com/questions/56872664/complex-dataset-split-stratifiedgroupshufflesplit (accessed Jan. 25, 2023).
[78] J. Brownlee. “A Gentle Introduction to Threshold-Moving for Imbalanced Classification”. Machine Learning Mastery. https://machinelearningmastery.com/threshold-moving-for-imbalanced-classification/ (accessed Jan. 30, 2023).
[79] T. Scialom, L. Martin, J. Staiano, É. Villemonte de la Clergerie, and B. Sagot, “Rethinking Automatic Evaluation in Sentence Simplification,” 2021. [Online]. Available: arXiv:2104.07560.
[80] M. Martinc, S. Pollak, and M. Robnik-Šikonja, “Supervised and Unsupervised Neural Approaches to Text Readability,” *Computational Linguistics*, vol. 47, issue 1, 2021. [Online]. Available: https://aclanthology.org/2021.cl-1.6.pdf.