Segmenting Italian Sentences for Easy Reading

Marta Cozzini[1,*,†], Horacio Saggion[2,†]

[1]Università di Bologna, Via Zamboni 33, 40126 Bologna, Italy
[2]Universitat Pompeu Fabra, Carrer de la Mercè 12, Ciutat Vella, 08002 Barcelona, Spain

Abstract

Easy Read texts are essential for individuals with reading difficulties. These texts are developed according to institutional guidelines that establish clear rules for writing and structuring content in an accessible way. A key feature of Easy Read texts is the segmentation of sentences into smaller grammatical units, often presented on separate lines, to enhance readability. While several studies have addressed content simplification in easy-to-read materials, much less attention has been paid to the automatic segmentation of such texts. This project investigates whether this kind of segmentation can be automated in a reliable and efficient way, even with limited resources. The main goal is to develop and evaluate automatic methods for splitting texts into simpler, shorter units to support text simplification and improve overall readability. The methods developed and evaluated are a decision tree classifier and a prompting-based method using a large language model (LLM). The work is focused on Italian, and the application of these methodologies to this language represents a novel contribution.

Keywords

Text simplification, easy-to-read, automatic segmentation, ER resources, CLiC-it

1. Introduction

Easy-to-read materials are important to ensure that as many people as possible can access information, especially people with cognitive disabilities, who might find it harder to understand complex texts or learn new things. These specific materials follow shared guidelines designed to make reading and understanding easier thanks to clear and consistent writing. Inclusion Europe created easy-to-read standards for preparing this kind of content in different languages [1]. Although these guidelines were originally designed for people with cognitive difficulties, they're also helpful for others, such as non-native speakers or anyone who finds reading challenging. Among the various recommendations, particular attention is paid to the use of simple vocabulary, short sentences, and a clear logical structure. Some guidelines also emphasize the importance of dividing the text into smaller grammatical units to improve readability. The Inclusion Europe guidelines state that each sentence should ideally fit on a single line and that longer sentences should be split at natural linguistic boundaries: where people would pause when reading out loud. This attention to segmentation is not only important for proper text layout, but, as the guidelines suggest and the following example demonstrates, it also plays a significant role in enhancing text comprehensibility.

> The Inclusion Europe guidelines advise against writing:
>
> ```
> Il modo in cui questa frase è
> divisa non è facile da leggere.
> ```
>
> Instead, they recommend:
>
> ```
> Il modo in cui questa frase è divisa
> è facile da leggere.
> ```

From a linguistic perspective, the first version interrupts a verbal phrase composed of the auxiliary "è" and the past participle "divisa." This separation breaks the syntactic and semantic unity of the clause, making the sentence harder to process. By splitting these tightly connected elements across two lines, the reader's comprehension effort increases. As the guidelines suggest, such breaks should be avoided in order to maintain clarity and facilitate understanding.

Despite the growing interest in text simplification, the task of sentence segmentation in easy-to-read materials remains largely underexplored. Currently, there are very few resources that address easy-to-read principles in relation to automatic segmentation, and only a limited number of studies have investigated how segmentation can be implemented computationally within this framework. This work aims to fill this gap by exploring whether segmentation can be automated reliably and efficiently. In particular, we evaluate two approaches: a decision tree classifier and a prompting-based method using a large language model (LLM). Both models are tested on easy-to-read materials that we collected from sources

CLiC-it 2025: Eleventh Italian Conference on Computational Linguistics, September 24 — 26, 2025, Cagliari, Italy

[*] Corresponding author.

[†] These authors contributed equally.

✉ marta.cozzini@studio.unibo.it (M. Cozzini); horacio.saggion@upf.edu (H. Saggion)
iD 0009-0004-1992-9132 (M. Cozzini); 0000-0003-0016-7807 (H. Saggion)

that we consider particularly trustworthy in adhering to official ER guidelines. These very materials not only serve as the basis for our evaluation, but also represent a secondary contribution of this study, as they form two new corpora that can support future research not only on segmentation, but more broadly in the domain of Italian text simplification. Although they do not include original–simplified text pairs, they offer quality examples of simplified texts segmented according to established ER criteria.

The paper is structured as follows: Section 2 reviews related work, followed by Section 3 that introduces the corpora used in our experiments, discussing their sources, the methodology behind their creation as easy-to-read materials, and other relevant details. Section 4 provides a detailed description of our methodology for the segmentation task, including both the decision tree and the prompting approaches. Section 5 presents our experimental setup, while Section 6 analyzes the results, evaluating each method, comparing their performance, and providing insights into the findings. Finally, Sections 7 and 8 conclude the paper by discussing key takeaways, addressing limitations, and describing future research directions.

2. Related Work

Text segmentation plays an important role in promoting textual accessibility and can be considered a relevant component of both Automatic Text Simplification (ATS) and the development of easy-to-read materials. ATS is a Natural Language Processing (NLP) task aimed at reducing linguistic complexity of texts, while preserving their original meaning [2]. It may involve modifications at the lexical, syntactic, or discourse level. In recent years, research on ATS has focused on developing approaches to simplify and adapt texts for individuals with cognitive disabilities or language impairments [3]. While ATS relies on computational strategies, easy-to-read materials are instead based on institutional guidelines that define clear rules for structuring content in an accessible way. These two approaches often converge on similar features that enhance readability. These include the use of simple vocabulary and grammar, short sentences, a clear logical structure, and the explanation of complex concepts in simpler terms. Within both frameworks, text segmentation is frequently emphasized: each sentence should ideally fit on a single line, and if this is not feasible, it should be split at natural linguistic boundaries to enhance clarity and facilitate comprehension.

2.1. Sentence Segmentation

Sentence segmentation is particularly valuable for creating accessible materials for individuals with reading challenges. Line breaks strategically inserted within long sentences can significantly improve readability [4]. The core concept behind sentence segmentation for easy reading materials is the division of complex sentences into smaller, more digestible chunks. This segmentation must follow "natural linguistic boundaries, ending at a position in the sentence where a reader would naturally pause" [5]. While intuitive to understand, defining precise criteria for these natural boundaries remains challenging. Recent research has explored the optimal approach to sentence splitting for improved comprehension. Studies have found that dividing sentences does enhance readability, with a particular finding that bisecting the sentence leads to enhanced readability to a degree greater than when we create simplification by trisection [6][7]. This preference for two-sentence splits over three-sentence divisions has been confirmed through Bayesian modeling experiments using various linguistic and cognitive features [8]. For readers with learning difficulties, proper sentence segmentation is particularly valuable. Studies have found that sentence density is a significant negative predictor of inferential comprehension, meaning that "the higher the sentence density, the lower the ability of these students to find relationships between them" [9]. This finding underscores the importance of appropriate text segmentation for enhancing comprehension among diverse reader populations.

2.2. Automatic Sentence Segmentation

Despite increasing interest in text simplification, the specific task of automatic sentence segmentation in the context of easy-to-read (ER) materials remains largely underexplored. To our knowledge, only one study to date has directly investigated how segmentation can be computationally implemented within this framework [5], and currently, very few resources address ER principles in relation to automatic segmentation. However, segmentation plays a crucial role in related domains, most notably in subtitle generation, where readability is enhanced when subtitles are segmented at naturally occurring linguistic boundaries, in addition to meeting timing and space constraints. Research has shown that subtitle segmentation has a significant impact on readability [10], leading to the development of various computational approaches. For instance, Álvarez et al. [11] trained Support Vector Machine and Linear Regression models on professionally created subtitles to predict optimal subtitle breaks, later improving this method through the use of Conditional Random Fields [12]. These supervised approaches could, in principle, be adapted to ER settings, provided that sufficient annotated training data is available. Nonetheless, compared to subtitling, resources for ER segmentation are extremely limited. As we mentioned before, to date, only one study has directly addressed the problem of sentence segmentation for the generation of ER texts. This work explores multiple approaches, including the use of generative large language models (LLMs) under different prompting modalities and a scoring-based method compatible with both constituency parsing and masked language modeling (MLM). In addition, it tackles the problem of data sparsity by developing new segmentation-centric datasets for Basque, English, and Spanish, thus laying the groundwork for further research in this domain [5].

As the first study to focus specifically on automatic sentence segmentation within the context of ER materials, it has provided a valuable foundation for our work. Building on its insights, we aim to apply similar strategies to address the problem of sentence segmentation for Italian, a language for which text simplification resources and research remain scarcer compared to English or Spanish.

3. Corpora

To construct our corpora, we relied on two different websites: Due Parole [13] and Anffas [14], known for their adherence to ER guidelines. From each source, we created a separate corpus, which was later used in our experiments. We describe these corpora in more detail in the following subsections.

3.1. Corpus from Due Parole

On the Due Parole website, we accessed the online archive of Due Parole, an Italian easy-to-read magazine that was published, with some interruptions, between 1989 and 2006. The magazine was specifically designed to provide accessible information to a broad audience, with simplified texts created by a team of linguists, journalists, and teachers from the University of Rome 'La Sapienza'. The corpus collected from this source consists exclusively of magazine articles, providing a consistent and well-structured textual base for training and initial testing of our models. From the online archive of Due Parole, we collected only the articles available in digital format, as web scraping was necessary to build the corpus. During the web scraping process, we preserved all original line breaks present in the formatted texts as published online. To ensure that the Due Parole corpus complied with the Inclusion Europe guidelines, we referred to Piemontese[15], which outlines the guidelines followed by the Due Parole team when producing easy-to-read texts. Some of the key recommendations concerned text segmentation: whenever the page layout allowed, each line was designed to contain a complete unit of meaning. If it was not possible to keep a sentence on a single line, sentence breaks were carefully managed to avoid arbitrary line breaks, with each line always ending on a whole word and words were never split across lines. This careful approach to segmentation shows an understanding of its effect on readability, emphasizing that sentence splitting should be deliberate and meaningful, unlike the more random breaks often seen in standard newspapers. The final corpus contains 311 articles, comprising 4855 sentences. Each article was saved as a separate plain text file with a `.txt` extension. All files were encoded in UTF-8, with special characters and HTML tags removed during preprocessing to ensure a clean and consistent textual format. The articles are organized in a hierarchical folder structure reflecting the original metadata: first by publication year, then by month, and finally by magazine section (e.g., "sport", "cultura"). This structure reflects the original editorial organization and allows for easy filtering by date or topic.

3.2. Corpus from Anfass

Our second source of easy-to-read materials is the website of Anffas, a national association of families of individuals with intellectual and/or relational disabilities. Anffas was one of the partners involved in the project that led to the definition of the European easy-to-read Guidelines [16]. Therefore, we can expect that the texts in the section “Documenti facili da leggere” ("Easy-to-read documents") that we can find on the website follow these official guidelines. From all the easy-to-read materials published there, we selected only the texts included in the easy-to-read magazine 'A modo mio'. This choice was motivated by the need to align with the other corpus, which also consisted exclusively of magazine articles. The Anffas corpus was used exclusively as a test set. Unlike the Due Parole corpus, creating this corpus as plain text was more difficult because the texts were only available in PDF format, which ruled out the use of web scraping. We therefore had to convert them manually. However, because there were significantly fewer Anffas texts compared to Due Parole, this operation did not require too much time. Similarly to Due Parole, we preserved all original line breaks present in the formatted texts as published online. The final corpus contains 38 articles comprising 481 sentences. The articles are organized into folders corresponding to each magazine issue, labeled by month and year. Within each issue folder, there is one plain text file (`.txt`) per magazine section (e.g., "sport", "spettacoli e televisione").

Table 1 summarizes the statistics of our corpora, including the total number of sentences, the number of sentences that contain at least one segmentation point, and the number of sentences without any segmentation.

Table 1
Corpora statistics

Sentences	Due Parole	Anfass
Total	4855	481
With segmentation	4271	204
Without segmentation	584	277

From Table 1, we observe the differences between the two corpora in terms of segmented sentences. Specifically, in the Due Parole corpus, the number of sentences containing at least one segmentation point is 4,271, corresponding to 88% of the total sentences. In contrast, this percentage drops to 42% in the Anfass corpus. This discrepancy is expected to affect the performance of our decision tree model, which was trained on the Due Parole corpus and subsequently tested on the Anfass corpus, as we will see in Section 6.

4. Methodology

To explore the viability of automatic text segmentation in low-resource settings, we adopted two different approaches: a traditional machine learning method informed by linguistic features (a decision tree) [17] and a current prompting Large Language Model approach.

4.1. Automatic Segmentation Using Decision Tree

We first approached the task of automatic text segmentation as a binary classification problem. In this framework, the model is trained to assign a binary label, 0 or 1, to each token in the input text, where 1 indicates that a segmentation should occur immediately after that token, while 0 means no segmentation. To build the training data, we started from raw texts extracted from the Due Parole dataset, that we described in the previous section. We first segmented the texts into sentences using spaCy's sentence tokenizer. Before sentence segmentation, we replaced all new line characters (\n) occurring within the text with a special marker <seg>, in order to preserve formatting information for subsequent processing (see step 2 of the example below). We then used the <seg> markers to split each sentence into smaller chunks, corresponding to the original internal line breaks (as shown in step 3). These splits helped us identify potential segmentation points within the sentence. For each token in the sentence, we assigned a binary label: 1 if it ended a chunk (except the final chunk in a sentence, labeled 0), and 0 otherwise. These labels serve as the target outputs that the model is trained to predict. Only after creating these target labels, the <seg> markers were removed, and the cleaned sentences reconstructed and re-tokenized with spaCy to prepare the data for further processing (see step 4). The following example illustrates the prepocessing steps applied to our corpus before training the decision tree model:

1. **Original input**

   This example sentence, extracted from the raw text, will be used to illustrate the prepocessing steps. Note that at this stage of the pipeline, the sentence is provided for demonstration purposes only, as the original text has not yet been segmented into sentences. In this example, newline characters indicate editorial line breaks.:

   ```
   La Costituzione è l'insieme
   delle leggi più importanti
   della Repubblica italiana.
   ```

2. **Intermediate representation**

   In this intermediate form, the raw text is segmented into sentences, and newline characters are replaced with a special segmentation marker:

   ```
   La Costituzione è l'insieme <seg>
   delle leggi più importanti <seg>
   della Repubblica italiana.
   ```

3. **Segmented output**

   The text is then split into segments at the positions marked by the <seg> tokens, which serve to identify potential segmentation boundaries:

   ```
   ['La Costituzione è l'insieme',
      'delle leggi più importanti',
      'della Repubblica italiana.']
   ```

4. **Linguistic analysis**

   Finally, the reconstructed sentence is used for token-level feature extraction in the classification model:

   ```
   La Costituzione è l'insieme delle
   leggi più importanti della Repubblica
   italiana.
   ```

After reconstructing the sentences, we performed feature extraction, including token-level features such as part-of-speech (POS) tags, sentence length (in tokens and characters), token length (in characters), and the token's position within the sentence. We converted POS tags into binary features using one-hot encoding. Then, all the features and target labels were organized into a tabular structure. A decision tree classifier was then trained on these data to predict segmentation.

4.2. Generative LLM Segmentation

Our second approach to automatic text segmentation involved using an instruction-tuned large language model (LLM) with zero-shot prompting. The design of our prompts was based on both the prompt strategies proposed in Calleja et al.[5] and the recommendations outlined in the Inclusion Europe easy-to-read (ER) guidelines. Following the approach of Calleja et al. [5], we designed two separate prompts. The first prompt (Prompt 1) aligns with the formal Inclusion Europe guidelines that state "tagliate la frase lì dove le persone farebbero una pausa leggendo la frase a voce alta" [1], while the second (Prompt 2) relies on the identification of natural grammatical boundaries. Unlike Prompt 1, Prompt 2 avoids explicit mentions of reading pauses, which could be less accessible or meaningful to the model. To make the prompts more specific, we introduced an additional constraint on the length of the segment, specifying that each segment should contain between 5 and 15 words. As is standard when prompting LLMs, we added also explicit instructions to ensure that the model would only output the requested content, without generating any additional text. In particular, we specified that the model should not include numbers, symbols, or bullet points at the beginning of lines, as our preliminary tests revealed a tendency to introduce such formatting elements.

- **Prompt 1:** `Dividi la seguente frase in segmenti separati, inserendo un ritorno a capo dove le persone farebbero una pausa leggendo la frase ad alta voce. Ogni segmento di testo dovrebbe contenere tra le 5 e le 15 parole. Il contenuto della frase originale non deve essere alterato in nessun modo; pertanto non deve essere aggiunta nuova informazione di alcun tipo. Scrivi ogni segmento su una nuova riga, senza numerazione o simboli all'inizio. Non generare altro testo ad eccezione del testo originale segmentato.`
- **Prompt 2:** `Dividi la seguente frase in segmenti separati, che rispettino i confini grammaticali naturali. Ogni segmento di testo dovrebbe contenere tra le 5 e le 15 parole. Il contenuto della frase originale deve essere mantenuto rigorosamente; pertanto non deve essere aggiunta nuova informazione di alcun tipo. Scrivi ogni segmento su una nuova riga, senza numerazione o simboli all'inizio. Non generare altro testo ad eccezione del testo originale segmentato.`

5. Experiments

Our first approach to automatic sentence segmentation was based on a traditional machine learning model. In particular, we employed a decision tree Classifier implemented via the `DecisionTreeClassifier` class in the `sklearn.tree` Python library [18]. To ensure replicability of our results, we set the `random_state`. Additionally, we configured the classifier with the parameter `class_weight='balanced'`, which automatically adjusts weights inversely proportional to the class frequencies in the input data. This choice was motivated by the significant imbalance in our dataset, where the target label 1 (indicating a segmentation point) is much less frequent than label 0 (no segmentation). To reduce the negative impact of this imbalance on model performance we adopted this built-in balancing strategy provided by scikit-learn.

For the prompting experiments, we used Gemma 2 9b, part of Google's Gemma family of lightweight, state-of-the-art decoder-only large language models. A key advantage of this family is the relatively small model size and the availability of open weights, which make the models suitable for deployment in resource-limited environments such as laptops or personal cloud infrastructure. We loaded the model and tokenizer via the Hugging Face Transformers library, employing automatic device mapping and `bfloat16` precision for efficient inference. Text generation was performed with controlled sampling parameters: a maximum of 150 new tokens, temperature set to 0.7, and nucleus sampling `top_p` at 0.9.

The decision tree classifier was initially trained and tested on a portion of the Due Parole corpus (see Table 2), allowing an initial evaluation of its performance. Subsequently, to assess the model's behavior on different types of texts, the decision tree was also tested on the Anffas corpus. At the same time, the LLM-based segmentation approach was applied exclusively to sentences from the Anffas corpus, in order to ensure that the results produced by the decision tree and the LLM would be directly comparable. As will be explained in more detail below, applying the same evaluation procedure to the Due Parole test set would have required excluding a substantial portion of the data, potentially biasing the results.

Table 2 shows the distribution of the Due Parole corpus across the training, validation, and test sets.

Table 2
Data partition statistics (number of tokens)

	Due Parole
Train	64252
Validation	7140
Test	7933

6. Results

To evaluate the performance of our approaches, we relied on standard metrics commonly used in binary classification tasks, such as precision, recall, and F1-score. These metrics provide a comprehensive overview of model effectiveness, particularly in scenarios with imbalanced classes.

6.1. Decision Tree Evaluation

Table 3
Results of automatic segmentation using decision tree and Due Parole as a test set

Target label	Precision	Recall	F1-score
No segmentation (0)	0.90	0.90	0.90
Segmentation (1)	0.38	0.38	0.38

Table 4
Results of automatic segmentation using decision tree and Anfass as a test set

Target label	Precision	Recall	F1-score
No segmentation (0)	0.96	0.91	0.93
Segmentation (1)	0.12	0.27	0.17

The decision tree model was assessed using the `classification_report` function from the `sklearn.metrics` module [18], which computes precision, recall, and F1-score. The initial evaluation was performed on a held out portion of the Due Parole corpus used as the test set. Table 3 summarizes the results obtained from this first test. Subsequently, to assess the model's behavior on different types of texts, the decision tree was also tested on the Anffas corpus. As shown in Table 4, the results differ substantially: the model performs notably worse. This performance drop can be attributed to the mismatch between the training data and the new test data. Although both corpora adhere to the Inclusion Europe guidelines and both consist of magazine articles, the texts in the Due Parole corpus exhibit a more uniform structure, largely influenced by the magazine's fixed layout. In contrast, the Anffas 'A modo mio' texts, while also published in magazine format, feature a more variable graphic layout, which may have affected the model's ability to generalize. Another contributing factor is the discrepancy in the proportion of segmented sentences between the two corpora that we described in 3.2: while Due Parole contains 88% of sentences with at least one segmentation point, this percentage drops to only 42% in Anfass. This results in fewer positive instances (i.e., target variable = 1) in the Anfass corpus, which further contributes to the already critical issue of target variable imbalance. This imbalance, as discussed earlier, consistently influences model performance both on the Due Parole test set and on the Anfass corpus, as reflected in the results tables. It notably affects the model's ability to correctly identify the minority class (label 1), which corresponds to segmentation points, resulting in lower precision, recall, and F1 scores. This trend is especially visible in the results obtained on the Anffas corpus, where the model, trained on the more uniform Due Parole texts, struggles even more to generalize. The confusion matrix for the texts tested in the Anffas corpus (Table 6) further confirms the difficulty of the model in performing the segmentation task. This matrix reveals a high number of false positives (487), where the model incorrectly inserts a segmentation point (label 1) when none is required (label 0), leading to unnecessary breaks in the text. Moreover, the model fails to identify 172 actual segmentation points (false negatives), highlighting its tendency to miss where a break should occur. With only 65 true positives out of 237 actual positive cases, the model demonstrates a limited ability to detect segmentation points. This issue is not limited to the Anffas corpus: although results are slightly better on the Due Parole test set (Table 5) the overall performance remains sub-optimal. The model tends to generalize poorly when deciding where to segment, struggling both to avoid over-segmentation and to reliably identify the appropriate break points.

6.1.1. Feature Importance Analysis

To further understand the model's behavior, we examined the feature importance values extracted from the trained decision trees.

As reported in Table 7, the most influential predictors in both corpora are not morphosyntactic cat-

Table 5
Confusion matrix for the decision tree model on the Due Parole test set

	Actual 0	Actual 1
Predicted 0	6212	662
Predicted 1	654	405

Table 6
Confusion matrix for the decision tree model on the Anfass test set

	Actual 0	Actual 1
Predicted 0	4664	172
Predicted 1	484	65

Table 7
Feature Importance Values from Decision Trees on Anffas and Due Parole Corpora

Feature	Anffas	Due Parole
distanza_da_prima_parola	0.2320	0.1715
frase_len_token	0.2051	0.1765
frase_len_char	0.1725	0.2500
PRON	0.0977	0.0976
CCONJ	0.0968	0.0970
token_len_char	0.0907	0.0971
ADP	0.0252	0.0233
ADV	0.0190	0.0213
NOUN	0.0150	0.0185
VERB	0.0134	0.0151
NUM	0.0137	0.0138
ADJ	0.0096	0.0086
DET	0.0060	0.0063
PUNCT	0.0033	0.0036

egories, but rather positional features. In the Anffas corpus, distanza_da_prima_parola, frase_len_token, and frase_len_char dominate the ranking (23.2%, 20.5%, and 17.2% respectively), together accounting for more than 60% of the model's decisions. These features capture sentence length (in tokens and characters) as well as token position within the sentence. Similarly, in Due Parole, the top positions are held by frase_len_char (25%), frase_len_token (17.6%), and distanza_da_prima_parola (17.1%), confirming the central role of sentence length and token positioning. Among morphosyntactic categories, PRON and CCONJ are consistently relevant in both datasets (around 9–10%), while core lexical classes such as VERB, NOUN, and ADJ play a comparatively minor role (below 2% in both corpora). One unexpected result concerns punctuation. Despite the intuitive assumption that punctuation strongly signals natural break points (e.g., commas, periods, dashes), the PUNCT feature accounts for only 0.3% of the total feature importance in both corpora. This is striking, considering that many segmentation guidelines, including those from easy-to-read standards, emphasize splitting long sentences "where a reader would naturally pause"[1], and punctuation marks are prototypical indicators of such pauses. One plausible explanation for the low importance assigned to punctuation is related to the length of the sentences in the training data. Since many of the texts adhere to easy-to-read principles, the sentences are often already short and simple, which means that internal punctuation marks (such as commas or colons) appear less frequently. As a result, punctuation rarely aligns with actual segmentation points in the dataset, reducing its statistical weight in the model's learning process. Moreover, punctuation that does appear, such as final periods, is not annotated as a segmentation point, as it naturally marks the end of a sentence. Taken together, these factors contribute to the surprisingly low feature importance of punctuation observed in the analysis. An ablation study, which systematically removes or isolates features to assess their individual and combined effects, could improve the overall understanding of feature contributions. Additionally, the influence of punctuation could be investigated by partitioning the dataset into sentences with and without punctuation and comparing feature importance between these groups. This would clarify whether punctuation plays a different role depending on its presence in the sentence. These investigations are left for future work.

6.2. LLM Evaluation

Evaluating the performance of the decision tree model was straightforward thanks to the availability of standard metrics and the `classification_report` function from the `sklearn.metrics` module. However, assessing the performance of the Large Language Model (LLM) proved to be more complex. This is because, whereas the decision tree outputs a binary label (0 or 1) for each token, the LLM produces fully segmented sentences as output. To enable a direct comparison with the decision tree, we first converted each segmented sentence into a binary sequence. In this sequence, tokens immediately preceding a line break were assigned a label of 1, except for line breaks corresponding to the final period of a sentence or cases where an entire sentence appeared on a single line, which were labeled 0 since they do not represent meaningful segmentation points in our task. To ensure a fair comparison with the decision tree, we aligned the length of the sequences produced by the LLM with those of the reference data, since the evaluation metrics used, such as precision, recall, and F1 score, are sensitive to sequence length and require a one-to-one correspondence between tokens. For this reason, before converting the segmented sentences into binary sequences, we manually reviewed the LLM outputs to identify and remove noisy cases.

Table 8
Sentence count in the original and reduced versions of the Anffas dataset

Prompt	Original sentences	Modified outputs
Prompt 1	481	58
Prompt 2	481	139

Table 9
Sentence count in the original and reduced versions of the Due Parole dataset

Prompt	Original sentences	Modified outputs
Prompt 1	480	123
Prompt 2	480	218

Despite explicit instructions in the prompt to generate no additional text beyond the original sentence, the LLM occasionally violated this rule. Consequently, we excluded from both our test sets, Anfass and Due Parole:

- Sentences in which the LLM added additional content, despite the prompt instructions explicitly prohibiting it;
- Sentences where the LLM altered the original punctuation, introducing tokens and segmentation breaks not present in the reference.

After this filtering step, we converted the cleaned LLM outputs into binary sequences and computed the same evaluation metrics used for the decision tree, allowing for a consistent and comparable analysis.

Table 8 shows the number of sentences per prompt that had to be removed from the Anfass test set due to changes made by the LLM in generating the output. In the case of the first prompt, the model introduced new content or altered the original sentence in 58 out of 481 cases, indicating relatively good adherence to the instructions. In contrast, the second prompt led to 139 modified outputs. This total includes the 58 cases affected by the first prompt, most of which were also altered in the second output. The higher number of 139 modified sentences for the second prompt reflects both these overlapping cases and additional sentences uniquely altered in the second output. This increase is likely due to the vagueness of the expression "grammatical boundaries," which the model tended to interpret more strongly, often replacing simple line breaks with stronger punctuation marks, possibly due to the presence of the term "boundaries". As a result, we were able to evaluate the LLM's performance on only 342 sentences from the original 481 in the Anffas dataset. To ensure comparability, we applied the same filtering to the decision tree evaluation, testing it exclusively on this same subset of sentences. On the Due Parole test set, even more sentences had to be excluded from the evaluation, as shown in Table 9: 123 from the first prompt and 218 from the second. Although these exclusions occurred, we decided not to proceed with the evaluation on the Due Parole test set. Following the methodology described above, this would have left us with only 260 evaluable sentences, corresponding to just 54% of the dataset. Such a reduction could bias the evaluation, as it might disproportionately exclude not only correctly segmented instances but also those where the model fails to segment properly. Future work will investigate alternative evaluation strategies more appropriate for this setting, including metrics such as BLEU and edit distance.

6.3. Comparison between the Approaches

Table 10
Comparative results for decision tree and Prompting (Prompt 1 and Prompt 2) on the Anfass reduced test set

Label	Model	Precision	Recall	F1-score
0	Decision Tree	0.97	0.91	0.94
	Prompt 1	0.98	0.94	0.96
	Prompt 2	0.98	0.93	0.96
1	Decision Tree	0.10	0.24	0.15
	Prompt 1	0.48	0.51	0.49
	Prompt 2	0.44	0.47	0.45

To provide a comprehensive evaluation, we compared the performance of the LLM-based approach, tested exclusively on the Anfass dataset, with the decision tree results, as summarized in Table 10. The LLM results reveal, once more, a marked imbalance between the two target labels (0 and 1). It is important to note that, when converting the LLM outputs into binary sequences, all sentences that appeared entirely on a single line in the corpus were automatically assigned only 0s. In cases where the corresponding gold standard sentence was also on a single line and contained no segmentation points, we modified the default behavior of the `precision_recall_fscore_support` function to better reflect this scenario. By default, the function may return undefined or misleading values when both `y_true` and `y_pred` contain only 0s. To avoid this, we configured the function so that it would treat such predictions as fully correct and automatically assign precision, recall, and F1-score values of 1.0. As reported in Table 10 on the Anffas reduced dataset, the LLM outperformed the decision tree overall. However, this result should be interpreted with caution, especially considering that, as shown in Section 6.2, it required excluding approximately one-quarter of the original corpus. The exclusion was necessary due to the model's tendency to introduce extra punctuation or to generate text exceeding the original input. This behavior resulted in the loss of valuable data, which is particularly critical in contexts where data are already scarce, such as in easy-to-read materials.

6.4. Comments on the Results

These results should be interpreted with caution, as segmentation is a non-standard and inherently subjective task within the context of text simplification and easy-to-read materials, precisely because multiple segmentations can be valid for any given sentence, each potentially facilitating comprehension in different ways. However, conventional evaluation metrics such as precision and recall enforce a strict binary framework, classifying predicted segmentations as either entirely correct or completely incorrect. This approach fails to consider cases where a segmentation, although different from the reference, is still reasonable or partially appropriate in terms of improving readability. As a result, predictions that are close to the gold standard or practically acceptable are often penalized as errors, which can underestimate the model's true performance and limit its applicability in real-world contexts.

7. Conclusion

The results obtained indicate that LLMs outperform a simple decision tree in the task of automatic sentence segmentation. However, as previously noted, these improved results come at a cost; to properly evaluate the LLM, we had to substantially reduce our test set, resulting in the loss of valuable data in a domain where data availability is already limited. Additionally, LLMs demand significantly more computational resources and runtime, requiring GPU acceleration to produce their outputs. Given these important considerations, it is worth discussing whether traditional machine learning approaches may still be appropriate for tasks of this nature. While our results do not provide conclusive evidence in this regard, it remains possible that more sophisticated traditional models, beyond simple decision trees, could achieve competitive performance in automatic segmentation. Future research could explore alternative models better suited to handling imbalanced features and class distributions, an issue evident in our datasets. Another contribution of this work lies in the creation and compilation of the Anffas and the Due Parole datasets. Although these corpora do not include the original source texts typically present in other resources for Italian text simplification, they nonetheless represent valuable assets. Beyond their utility for segmentation research, they provide a source for broader investigations within the field of text simplification. Currently, these datasets are pending authorization for public release. Once approved, they will be made openly accessible to the research community, supporting future research on various aspects of Italian text simplification.

8. Limitations and Further Work

The Inclusion Europe guidelines provide only vague instructions on segmentation, and there are cases in which our benchmarks even contradict these guidelines. Moreover, segmentation remains a subjective task: while text layout influences decisions, multiple strategies can be equally valid for improving comprehension. Another limitation is that the psycholinguistic impact of segmentation and its role in enhancing understanding have only been explored to a limited extent. Due to time constraints, our study did not differentiate between grammatical and ungrammatical segmentations, such as splitting an article from its noun, but this represents an interesting area for future research. For our evaluation, we used precision, recall, and F1-score, mainly to ensure comparability with the decision tree results. However, these metrics present two main limitations: first, they impose a rigid binary judgment that fails to account for the inherent subjectivity of segmentation; second, they require a strict one-to-one token correspondence, which led to the loss of valuable data whenever the model added informative tokens to the output. As mentioned in section 6.2, future work should explore alternative evaluation strategies, such as BLEU or edit distance metrics, although the use of edit distance would require a careful discussion to define what constitutes a meaningful edit. In addition, human evaluation should be considered to gain deeper insights beyond what quantitative metrics alone can offer.

Acknowledgments

This document is part of a project that has received funding from the European Union's Horizon Europe research and innovation program under Grant Agreement No. 101132431 (iDEM Project). The views and opinions expressed in this document are solely those of the author(s) and do not necessarily reflect the views of the European Union. Neither the European Union nor the granting authority can be held responsible for them. We also acknowledge support from the Spanish State Research Agency under the Maria de Maeztu Units of Excellence Program (CEX2021-001195-M). We are grateful to the reviewers for their valuable comments, which have significantly contributed to improving this work. This work was conducted during a mobility funded by the Erasmus+ Traineeship Programme of the European Union, whose support is gratefully acknowledged.

References

[1] I. Europe, Information For All: European Standands for making information easy to read and understand (Easy-to-read ed.), 2009.

[2] S. Bott, H. Saggion, Text simplification resources for spanish, Lang. Resour. Evaluation 48 (2014) 93–120. URL: https://doi.org/10.1007/s10579-014-9265-4. doi:10.1007/S10579-014-9265-4.

[3] H. Saggion, J. O'Flaherty, T. Blanchet, S. Sharoff, S. Sanfilippo, L. Muñoz, M. Gollegger, A. Rascón, J. L. Martí, S. Szasz, S. Bott, V. Sayman, Making democratic deliberation and participation more accessible: The idem project., in: A. Bonet-Jover, R. Sepúlveda-Torres, R. M. Guillena, E. Martínez-Cámara, E. L. Pastor, Rodrigo-Yuste, A. Atutxa (Eds.), SEPLN (Projects and Demonstrations), volume 3729 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2024, pp. 71–76. URL: http://dblp.uni-trier.de/db/conf/sepln/sepln2024pd.html#SaggionOBSSMGRM24.

[4] Y. Hayashibe, K. Mitsuzawa, Sentence boundary detection on line breaks in japanese, in: WNUT, 2020. URL: https://api.semanticscholar.org/CorpusID:226283860.

[5] J. Calleja, T. Etchegoyhen, D. Ponce, Automating Easy Read Text Segmentation, in: Y. Al-Onaizan, M. Bansal, Y.-N. Chen (Eds.), Findings of the Association for Computational Linguistics: EMNLP 2024, Association for Computational Linguistics, Miami, Florida, USA, 2024, pp. 11876–11894. URL: https://aclanthology.org/2024.findings-emnlp.694/. doi:10.18653/v1/2024.findings-emnlp.694.

[6] T. Nomoto, Does splitting make sentence easier?, Frontiers in Artificial Intelligence 6 (2023). URL: https://api.semanticscholar.org/CorpusID:262193456.

[7] T. Nomoto, The fewer splits are better: Deconstructing readability in sentence splitting, ArXiv abs/2302.00937 (2023). URL: https://api.semanticscholar.org/CorpusID:256460905.

[8] T. Passali, E. Chatzikyriakidis, S. Andreadis, T. G. Stavropoulos, A. Matonaki, A. Fachantidis, G. Tsoumakas, From lengthy to lucid: A systematic literature review on nlp techniques for taming long sentences, ArXiv abs/2312.05172 (2023). URL: https://api.semanticscholar.org/CorpusID:266149795.

[9] I. Fajardo, V. Ávila, A. Ferrer, G. Tavares, M. Gómez, A. M. Hernández, Easy-to-read texts for students with intellectual disability: linguistic factors affecting comprehension., Journal of applied research in intellectual disabilities : JARID 27 3 (2014) 212–25. URL: https://api.semanticscholar.org/CorpusID:33895340.

[10] E. Perego, F. D. Missier, M. Porta, M. M. and, The cognitive effectiveness of subtitle processing, Media Psychology 13 (2010) 243–272. doi:10.1080/15213269.2010.502873.

[11] A. Álvarez, H. Arzelus, T. Etchegoyhen, Towards customized automatic segmentation of subtitles, in: J. L. Navarro Mesa, A. Ortega, A. Teixeira, E. Hernández Pérez, P. Quintana Morales, A. Ravelo García, I. Guerra Moreno, D. T. Toledano (Eds.), Advances in Speech and Language Technologies for Iberian Languages, Springer International Publishing, Cham, 2014, pp. 229–238.

[12] A. Álvarez, C.-D. Martínez-Hinarejos, H. Arzelus, M. Balenciaga, A. del Pozo, Improving the automatic segmentation of subtitles through conditional random field, Speech Communication 88 (2017) 83–95. URL: https://www.sciencedirect.com/science/article/pii/S0167639316300127. doi:https://doi.org/10.1016/j.specom.2017.01.010.

[13] Due Parole, Due parole, s.d. URL: https://www.dueparole.it/.

[14] Anffas, Documenti facili da leggere, https://www.anffas.net/it/linguaggio-facile-da-leggere/documenti-facili-da-leggere/, s.d.

[15] M. E. Piemontese, Scrittura e leggibilità: «due parole», in: M. A. Cortelazzo (Ed.), Scrivere nella scuola dell'obbligo, Quaderni del Giscel, La Nuova Italia, Firenze, 1991, pp. 151–167.

[16] Inclusion Europe, Pathways2, s.d. URL: https://www.inclusion-europe.eu/pathways-2/.

[17] D. Steinberg, Cart: Classification and regression trees, 2009. URL: https://api.semanticscholar.org/CorpusID:116184048, technical report.

[18] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. VanderPlas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, É. Duchesnay, Scikit-learn: Machine learning in python, Journal of Machine Learning Research 12 (2011) 2825–2830. URL: https://scikit-learn.org/stable/modules/tree.html.

Declaration on Generative AI

During the preparation of this work, the author(s) used ChatGPT (OpenAI) in order to: Paraphrase and reword. After using these tool(s)/service(s), the author(s) reviewed and edited the content as needed and take(s) full responsibility for the publication's content.