# Unveiling the Segmentation Power of LLMs: Zero-Shot Invoice Item Description Analysis

**Vitória S. Santos**[1] **and Carina F. Dorneles**[1]

[1] Departamento de Informática e Estatística – Universidade Federal de Santa Catarina (UFSC)

vit2santoss@gmail.com, dorneles@gmail.com

***Abstract.*** *Segmenting invoice item description into attributes that describe its features may be a newsworthy alternative for subsequent entity resolution. This paper presents a set of experiments to show the performance of seven LLMs, including Llama-3, Sabiá-2-Medium, Command R+, Claude 3 Opus, GPT-3.5, GPT-4, and Mixtral 8x22B, in segmenting text within Invoice items descriptions using zero-shot learning techniques. We have employed accuracy, precision, recall, and $F_1$-score evaluation metrics to highlight the effectiveness of LLMs. The experiment involved segmentation preparation, model training, prompt optimization, attribute extraction, and output generation. The objective is to determine each model's precision in accurately identifying segmentation within invoice item descriptions.*

## 1. Introduction

Assessing whether distinct data instances represent the same real-world object (entity resolution [Hoffart et al. 2012]) is essential for a wide variety of applications, such as data integration [Varma et al. 2021], data matching [Dorneles et al. 2011], and so on. The big challenge of an entity resolution process is working with conflicting data, such as inconsistent, typos, missing, dirty, or even fake data. The main reason for this data conflict is the way people enter data, especially when there is no strict standard for doing so. Considering data from multiple sources or users, the problem is even more challenging.

A scenario where these conflicting data problems are prevalent is that of Electronic Invoices. Figure 1(a) shows real examples extracted from items from medicine sales invoices, which present high heterogeneity in the description of medicines. In a manual analysis of the items presented in the figure, it is possible to observe that the lines painted in yellow refer to the same object in the real world (Injectable Dipyrone, 500 mg with 50ml), as do the lines painted in green (Dipyrone in Drops, 500mg with 10ml). To solve the entity resolution problem in this scenario, segmenting the medicine description into attributes that describe the characteristics of each item (as shown in Figure 1(b)) is an interesting alternative for subsequent entity resolution.

Many works in the literature explore text segmentation, highlighting its critical role as a task of information extraction process from the web and its application to domains such as medication descriptions [Lerman et al. 2004], [Haider and Yeşilada 2022], and [Chen et al. 2022]. The granularity of the term "segmentation", used in literature, depends on the object to be segmented. Generally, the process of dividing an object into meaningful units is called "object segmentation" [Kayed et al. 2021]. Despite advancements, traditional methods often need help with varied and large-scale datasets. This gap

| Original |
|---|
| **Description** |
| Dipyrone 500 mg/mL Injectable, Amp. 2 mL |
| Injectable Dipyrone – Bottle with 50 Ml. Febrax or Similar. |
| Sodium Dipyrone 500 Mg/ML |
| Sodium Dipyrone 500 Mg, Injectable Solution Indicated for Various Conditions, Composed of Sodium Dipyrone 500 Mg, Aqueous Vehicle Q.S. to 1 Ml, 50 Ml Ampoule. |
| Sodium Dipyrone 500 MG/ML with 10 ML |
| Sodium Dipyrone 500 Mg/Ml Injectable Solution, Ampoule with 2 Ml Each |
| Sodium Dipyrone Drops 10 Ml |
| Sodium Dipyrone Ibasa 50% (50G) 50 Ml |
| Sodium Dipyrone, 50 Mg/Ml, Oral Solution, Bottle with 10 Ml (CIM10569) |
| Sodium Dipyrone, 500 mg/mL, Oral Solution (Drops), Bottle with 10 mL (CIM9104) |
| Sodium Dipyrone, Dosage: 500 Mg/Ml, Presentation: Drops, Bottle 10 Ml |
| (a) |

| Processed | | | | | |
|---|---|---|---|---|---|
| **Description** | **Volume** | **Concentration** | **Type** | **Presentation** | **Others** |
| Dipyrone | 2Ml | 500Mg | Ampoule | Injectable | N/A |
| Dipyrone | 50Ml | | Bottle | Injectable | Febrax or Similar |
| Sodium Dipyrone | N/A | 500Mg | N/A | N/A | N/A |
| Sodium Dipyrone | 50Ml | 500Mg | Ampoule Bottle | Injectable | Indicated for Various Conditions - Aqueous Vehicle Q.S. to 1 Ml |
| Sodium Dipyrone | 10Ml | 500Mg | N/A | N/A | N/A |
| Sodium Dipyrone | 2Ml | 500Mg | Ampoule | Injectable | |
| Sodium Dipyrone | 10Ml | NA | N/A | Drops | N/A |
| Sodium Dipyrone | 50Ml | 50G | N/A | N/A | Ibasa 50% (50G) |
| Sodium Dipyrone | 10Ml | 50Mg | Bottle | Oral Solution | (Cim10569) |
| Sodium Dipyrone | 10Ml | 500Mg | Bottle | Oral Solution (Drops) | (Cim9104) |
| Sodium Dipyrone | 10Ml | 500Mg | Bottle | Drops | Dosage: |
| (b) | | | | | |

**Figure 1. (a) Original data from invoices; (b) Treated data after processing**

underscores the potential of Large Language Models for robust and scalable segmentation, given their superior context understanding and ability to handle unstructured data [Simon and Lausen 2005], [Chen et al. 2023].

Large Language Models (LLMs) exhibit impressive versatility, addressing a wide range of tasks as evidenced by [Yao et al. 2024] and [Chang et al. 2023]. Their exceptional performance in natural language processing and domain-specific tasks has led to increased adoption, particularly for urgent information access needs. The segmentation of Electronic Invoice items is an application where LLMs could demonstrate significant potential, enhancing data management, solving entity resolution issues, and supporting analysis and evidence-based decision-making. In this context, LLMs' ability to extract relevant information from initial descriptions is crucial. Zero-shot prompting, where the prompt provided to the model does not include examples or demonstrations, allows us to evaluate this capability. This approach tests the models' adaptability and performance, providing insights into their ability to handle diverse linguistic variations and formatting styles.

In this article, we present a set of experiments that compare seven LLM models and their performance in segmenting Electronic Invoice items. The experiments and evaluations involving Llama-3[1], Sabiá-2-Medium[2], Command R+[3], Claude 3 Opus [4], GPT-3.5[5], GPT-4[6], and Mixtral 8x22B[7]. We aim to assess the models effectiveness in accurately identifying and extracting key attributes from medication descriptions, employing stringent evaluation criteria such as accuracy, precision, recall, and the $F_1$-score. Despite challenges like model context window limitations and tendencies toward "laziness" with longer descriptions, our study underscores the importance of evaluating models based not only on accuracy but also on their ability to handle diverse information types and nuances across various attributes. By enhancing our understanding of LLMs capabilities and limitations in segmentation tasks, especially in medication descriptions, this research contributes to improving their applicability and reliability in contexts requiring segmentation and data extraction tasks.

[1]https://llama.meta.com/
[2]https://www.maritaca.ai/
[3]https://cohere.com/
[4]www.anthropic.com/claude
[5]https://openai.com/
[6]https://openai.com/
[7]https://mistral.ai/

This paper is organized as follows. We explore some related work in segmentation, reviewing previous studies and methodologies in similar research endeavors in Section 2. Section 3 presents details of our segmentation process, outlining its methodology and procedural steps. In Section 4, we describe our sample, the evaluation metrics used for assessing the models, the experiments conducted and the results obtained. Finally, Section 5 concludes the paper and proposes avenues for future research.

## 2. Related Work

The segmentation process is a common task in the information extraction area, where from a plain text, it is desirable to have meaningful data units that characterize a given entity. Some works address the problem of extracting and segmenting references from PDF documents [Boukhers et al. 2019, Peng and McCallum 2006, Zhang et al. 2011], while another focusing on extracting and segmenting posting addresses [Borkar et al. 2001, Kayed et al. 2021, Cruz et al. 2021]. Considering greater granularity, the work proposed in [Misra et al. 2011] is to segment texts using topic modeling.

The work by [Boukhers et al. 2019] presents an innovative probabilistic approach aimed at enhancing segmentation accuracy by addressing variability in the content, length, and location of references within documents. This improvement is demonstrated through evaluations, with segmentation being performed using Conditional Random Fields (CRFs). Similarly, in their research on CRFs, [Peng and McCallum 2006] highlight the importance of accurate segmentation in enhancing search engine accuracy, particularly in the extraction of header and citation fields. Additionally, the utilization of structured SVM and CRFs by [Zhang et al. 2011] underscores the effectiveness of these methods in analyzing references with high precision, thus emphasizing the pivotal role of structured learning approaches in achieving accurate segmentations.

In [Cruz et al. 2021], the need for innovative segmentation methods to deal with challenges such as non-standardized or incomplete addresses is emphasized. This study investigates automated approaches to match addresses, recognizing segmentation as a critical step in this process. For instance, [Borkar et al. 2001] present the datamold method, which enhances Hidden Markov Models (HMM) to automatically segment unformatted text records into structured elements, such as addresses. This enhancement results in significant accuracy in extracting addresses from different cultural contexts. [Kayed et al. 2021] discuss the extraction of addresses from the web, highlighting the importance of segmentation in preparing unstructured data obtained from social media.

Furthermore, in [Misra et al. 2011], text segmentation was examined through the lens of latent Dirichlet allocation (LDA) and multinomial mixture (MM) models, revealing the superior segmentation performance offered by topic model-based methods compared to traditional techniques. Their work also addressed the computational overhead associated with LDA by proposing a modified dynamic programming algorithm. Expanding upon this foundational research, [Aumiller et al. 2021] introduce a novel legal document segmentation system leveraging transformer networks to predict the topical coherence of text segments. Trained on a comprehensive dataset comprising approximately 74,000 Terms of Service documents, their model surpasses baseline performance metrics and demonstrates robust adaptation to the nuanced structures inherent in legal documents.

Lastly, other types of work, such as those presented in [Lerman et al. 2004,

Haider and Yeşilada 2022, Chen et al. 2022, Simon and Lausen 2005, Chen et al. 2023], introduce methodologies ranging from automatic extraction and segmentation of web table records to unsupervised web data extraction and estimating classifier performance in unlabeled population segments. These works collectively underscore the importance of segmentation in various tasks related to web data processing, highlighting its role in facilitating accurate data extraction, pattern recognition, and model performance estimation. Effective segmentation is shown to be crucial for optimizing the performance and efficiency of diverse web data processing tasks.

Through these works, it becomes evident that segmentation is a foundational element in various web data extraction methodologies, enabling precise analysis and insight extraction from complex structures. However, a significant gap remains in performing segmentation across highly varied and large-scale datasets, such as medication descriptions. Traditional methods often struggle with the complexity and variability inherent in such data. This is where Large Language Models (LLMs) show great promise. With their deep understanding of context and semantics, LLMs can handle diverse and unstructured data more effectively than traditional approaches. They leverage vast amounts of training data to learn intricate patterns and relationships within the data, enabling higher accuracy in segmentation.

## 3. Segmentation process

The proposed segmentation process requires a pipeline with three sequential steps: (i) preprocessing; (ii) training; and (iii) attribute segmentation. Each step builds on the previous one, ensuring a structured and efficient flow from raw data to segmented outputs.

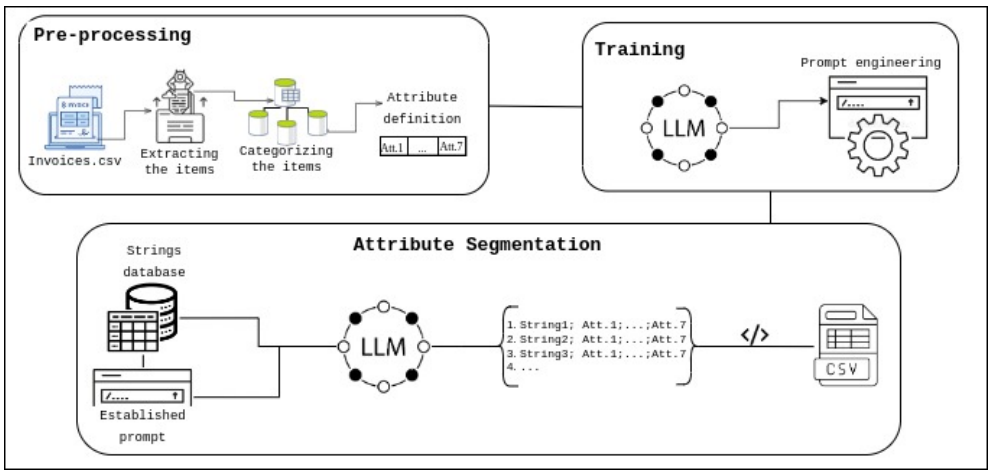


**Figure 2. Segmentation pipeline**

Figure 2 illustrates the pipeline, which follows the following steps:

- **Pre-processing:** This initial phase concerns data preparation and consists of three sub-steps:
  - (i) Strings containing medication descriptions were extracted from electronic invoices. This extraction was automated using Python scripts and Pandas for data manipulation, resulting in a dataset containing only the item descriptions.

(ii) The extracted descriptions were categorized into specific product classes using the NCM number. This categorization process was semi-automated with manual validation to ensure accuracy.
(iii) The fields or attributes for segmentation, were manually defined by the authors based on the data structure and segmentation requirements.

This pre-processing procedure results in a dataset of categorized items ready to be segmented according to the defined attributes.

- **Training:** Building on the pre-processed data, this step involves engineering prompts for the language models. We established a general prompt and tested various variations to identify the most effective one across all models. Despite setting a low model temperature to minimize creativity, we instructed the models to return only the provided information and avoid inferring non-existent attributes. This approach ensured accuracy, as some models tended to infer information not explicitly stated. The final prompt defined for this purpose was:

  ```
  Reorganize the data from <string> according to:
  <attribute1,...,attribute7>.  For any missing data,
  put N/A. Return the data in CSV format, separated by
  ";".
  ```

  The term ”N/A” denotes ”not applicable,” indicating the absence of this information in the string.
- **Attribute Segmentation:** In this final step, the LLM processes the item description strings stored in the String Database and segments them according to the defined attributes (Att.1, Att.2, ..., Att.7), guided by the established prompt. After segmentation, the data is exported to a structured CSV file, where each string is listed alongside its corresponding attributes for streamlined data analysis and manipulation. The final output is a detailed and well-organized CSV file containing segmented medication descriptions.

## 4. Experimental Evaluation

The main objective of this study is to evaluate the performance of models in segmenting attributes by feeding all the description data directly into the models' prompt, utilizing zero-shot prompting. This approach allows us to assess the models' ability to accurately identify and extract key attributes from medication descriptions, such as name, volume, concentration, and active ingredient. In this section, we describe the models used in the experiments, the dataset and data sample, the evaluation metrics used for assessing the models, the experiments conducted, and the results obtained.

### 4.1. Models

In our study, we selected the seven models described above for evaluation. These models were accessed via API, facilitating seamless integration into our research framework. The selection process prioritized models known for their proficiency in natural language processing tasks, ensuring robustness and reliability in our evaluation methodology.

- **Llama**. Meta AI's Llama 3, with custom 24K GPU clusters and over 15 trillion data tokens, offers enhanced productivity and creativity. With an 8K context length, it is available in 8B and 70B versions, supporting diverse AI applications.

Evaluations, such as those in [Yoon et al. 2024, Zhang et al. 2024], highlight its performance in various tasks.

- **Sabiá**. The Sabiá-2, developed by Maritaca AI, is a family of language models specialized in Portuguese text processing. It includes Sabiá-2 Small and Sabiá-2 Medium, with the latter featuring an 8192-token context window. Designed for domain-specific tasks,Sabiá-2 Medium surpassing the performance of GPT-3.5 and matching or exceeding GPT-4 in Brazilian exams, as demonstrated in [Almeida et al. 2024].
- **Claude**. Claude 3 Opus, Anthropic's premier language model, features an extensive context window of 200,000 tokens and can produce outputs up to 4096 tokens long. With approximately 150,000 words and 680,000 Unicode characters, it's designed for advanced tasks. Continuously updated training data until August 2023 ensures Claude 3 Opus remains at the forefront of state-of-the-art language models, as noted in [Uppalapati and Nag 2024].
- **Command**.Command R+ is Cohere's latest and most advanced model, designed for conversational interactions and tasks requiring extensive context. With a 128k context window, it efficiently processes information and outperforms previous Cohere models in tasks requiring broader context.
- **Mixtral**. The Mixtral 8x22B, developed by Mistral AI, stands out as the company's most efficient open model. With a 22B sparse Mixture-of-Experts (SMoE) architecture, it utilizes 39B of active parameters out of 141B and features a 64k context window.
- **GPT**.OpenAI's GPT-4 Turbo and GPT-4 models, built upon GPT-3.5, excel in natural language understanding and code generation. The gpt-4-0125-preview model, with a maximum output of 4,096 tokens and a context window of 128,000 tokens, demonstrates enhanced performance, especially in code generation tasks. Additionally, the gpt-3.5-turbo-0125 model, an updated version of GPT-3.5 Turbo, provides improved accuracy in responding to requested formats. Trained until September 2021, it returns a maximum of 4,096 tokens with a context window of 16,385 tokens.

Table 1 presents a comparison among the models, showing the context window and the maximum output. These are currently limiting factors for models, preventing the insertion of large inputs or restricting model responses. Comprehending these details is crucial to assessing the capabilities and limitations of each model, which directly influences its applicability in different scenarios.

| ***Model*** | ***API model name*** | ***Context window (Cw)*** | ***Max output*** |
|---|---|---|---|
| *Llama-3* | llama3-70b | 8k tokens | $\leq$ 4096 tokens |
| *Sabiá-2-Medium* | sabia-2-medium | 8.192 tokens | $\leq$ (8192 - Cw) tokens |
| *GPT-3.5 Turbo* | gpt-3.5-turbo-0125 | 16.385 tokens | $\leq$ 4096 tokens |
| *Mixtral 8x22B* | open-mixtral-8x22b | 64k tokens | $\leq$ (64k - Cw) tokens |
| *Command R+* | command-r-plus | 128k tokens | $\leq$ 2048 tokens |
| *GPT-4* | gpt-4-0125-preview | 128k tokens | $\leq$ 4096 tokens |
| *Claude 3 Opus* | claude-3-opus-20240229 | 200k tokens | $\leq$ 4096 tokens |

**Table 1. Comparison of models with varying context window sizes and maximum output lengths**

### 4.2. Dataset and Sampling method

For our analysis, we utilized a dataset comprising 4,861 medication descriptions related to the term "dipyrone", sourced from invoices provided by the Public Prosecutor's Office of Santa Catarina. These descriptions amount to approximately 114,208 tokens within a context window. However, this quantity exceeds the capacity supported by some of the models we are working with, both for input and for generating outputs that meet or exceed this volume, as demonstrated in Table 1. To overcome this challenge, we opted to extract a representative sample from this population of descriptions, enabling us to conduct our tests more efficiently, even if we have to perform them in batches.

When selecting this sample, we considered both the diversity and the frequency of the descriptions, given the non-probabilistic nature of our description population. According to [Rea and Parker 2012], which defines a small population as one with fewer than 100,000 items, we applied the formula provided in the book for calculating sample size. This approach allowed us to determine a sample size that is representative of our dataset. For our case, we used the following formula:

$$sample = \frac{Z^2 \cdot (0.25) \cdot N}{Z^2 \cdot (0.25) + (N-1) \cdot M_E^2}.$$

Where $M_E$ represents the margin of error in terms of proportions, $Z$ is the $Z$-score for different confidence levels, and $N$ is the population size. The margin of error $M_E$ indicates the precision of the sample estimate relative to the true population mean. A $Z$-score is a value representing how many standard deviations a given data point is away from the mean of a distribution. In this case, we used $Z = 2.575$ for a confidence level of $99\%$ and $M_E = 0.1$, resulting in a $10\%$ margin of error. This led us to a sample of $161$ descriptions, totaling approximately $4,446$ tokens, which is an acceptable amount for the context window of the models.

### 4.3. Prompt Configuration

Initially, our strategy aimed to input all description data directly into the model's prompt, facilitating segmentation and the generation of a structured CSV file. However, due to the limit of the model's context windows, which are incapable of processing prompts with thousands of input tokens, executing this process all at once became unfeasible. Given these limitations, and since our dataset sample has approximately 4.446 tokens, we had to split the input data into batches to comply with the maximum token limit supported by all the model's input/output. This partitioning allowed us to manage the data effectively and ensure that each model processed a manageable portion of the dataset.

We decided to divide the tasks into batches of 15 descriptions due to the output limitations of some models, which, even within the context window limit, could not handle more than 15 descriptions at a time. The decision to use batches instead of processing individual descriptions underscores the efficiency of this approach, especially with larger datasets. Batch processing significantly reduces the required time compared to the one-by-one method.

The one-by-one method refers to processing each description individually, one at a time, which can be highly time-consuming, especially for large datasets. In this method,

each description is processed separately, and the model's output for each description is handled individually. This approach does not take advantage of batch processing efficiencies and can result in increased overall processing time. For instance, processing 4,861 descriptions using the batch method of 15 descriptions takes approximately 2 hours and 36 minutes, whereas the one-by-one method would take around 4 hours and 35 minutes. This illustrates that batch processing is more efficient, as it significantly reduces the total time required for processing large volumes of data.

The Figure 3 illustrates the comparison of execution times for different quantities of descriptions (5, 10, and 15), for both batch processing and individual processing. The blue and red lines represent the execution time for batch processing and one-by-one processing, respectively. The data were obtained by running both methods with the Claude model and recording the time required to process each quantity of descriptions.

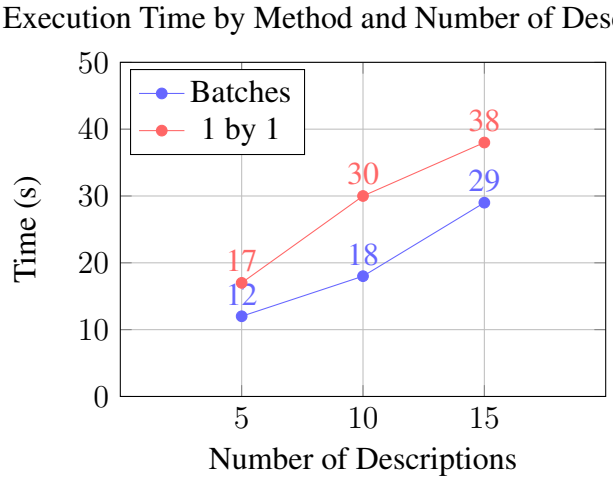


**Figure 3. Execution time comparison between batch processing and one-by-one processing using the Claude model.**

### 4.4. Evaluation criteria

For evaluating our experiments, a direct comparison between the model's responses was performed line by line to calculate fundamental classification evaluation metrics such as precision, recall, accuracy, and F1-score. Our segmentation process can be treated as a classification problem, where each segmentation unit identified by the model corresponds to a specific class, such as medication name, concentration, type of use, among others. These metrics allow us to evaluate the model's ability to make accurate predictions and correctly identify the attributes of interest.

Based on the methodology presented in [Géron 2019], precision and recall metrics are calculated as follows: precision is the ratio of true positives ($T_P$) to the sum of true positives and false positives ($F_P$), while recall is the ratio of true positives to the sum of true positives and false negatives ($F_N$). These metrics enable the evaluation of the model's ability to make accurate predictions and correctly identify the attributes of interest. Accuracy, in turn, represents the proportion of correct predictions made by the model relative to the total number of predictions. This metric provides a general measure of the model's performance and is often used as a first approach for classifier evaluation. The Figure 4 presents the formulas used for precision, recall and accuracy.

$$recall = \frac{T_P}{T_P + F_N}, \qquad precision = \frac{T_P}{T_P + F_P}, \qquad accuracy = \frac{T_p}{Total}.$$

**Figure 4. Recall, Precision and Accuracy formulas**

The $F_1$-score ($F_1$), derived from the harmonic mean of precision and recall, provides a unique measure that combines these two metrics into a single value. This measure is particularly useful when there is an imbalance between the classes because it penalizes classifiers that favor one metric over the other.

$$F_1 = \frac{2 \times precision \times recall}{precision + recall}.$$

### 4.5. Methodology

The segmentation process began with the pre-processing phase, where 4,861 medication descriptions related to the term "dipyrone" were extracted from electronic invoices. This dataset was then sampled down to 161 descriptions. During this phase, we established the attributes for segmentation: "Description", "Volume", "Concentration", "Type", "Active Ingredient", "Batch" and "Expiration Date".

In the subsequent prompt engineering phase, a general prompt was designed and tested across all models using 15 descriptions from the sample in a zero-shot manner to ensure its effectiveness for attribute extraction. Finally, during the attribute segmentation phase, the optimized prompt was applied by the LLMs to segment the 161 medication descriptions into the predefined attributes, producing a structured CSV file for analysis.

Then, we applied a normalization procedure to standardize the models' responses, facilitating fair and accurate metric comparisons. This normalization involved manipulating each column separately, including removing spaces, converting uppercase letters to lowercase and vice versa, and standardizing the representation of measurements. The objective was to mitigate any potential biases introduced by variations in input formatting or the writing nuances of the models.

Next, for metric calculation, we've adopted a binary approach to assess the performance of LLMs in the segmentation task. For example, considering the following string as an item description: `Dipyrone Monohydrate 500mg/ml Drops 10ml`, and the following structure as the model's desired result:

*model's expected return*

| Description; | Volume; | Concentration; | Type; | Active Ingredient; | Batch; | Expiration date; |
|---|---|---|---|---|---|---|
| Dipyrone Monohydrate 500mg/ml Drops 10ml; | 10 ml; | 500 mg/ml; | Drops; | Dipyrone Monohydrate; | N/A; | N/A; |

The binary scheme assigns a value of 1 to attributes present in the input description and 0 to those absent, including cases where the attribute is labeled as "N/A". For instance, given the description provided earlier, the corresponding binary representation would be $[1, 1, 1, 1, 1, 0, 0]$.

We extended this binary representation to encompass both the input descriptions and the LLMs' responses. Consequently, we could define the following:

- **True Positive ($T_P$):** This occurs when both the input and the LLM output a 1, indicating that the LLM correctly identified the attribute. It also includes cases when both the input and the LLM output a 0, signifying that the LLM correctly recognized the absence of the attribute or labeled it as "N/A".
- **False Positive ($F_P$):** This arises when the input is 0 ("N/A"), but the LLM outputs a 1, suggesting that the LLM generated information not present in the input.
- **False Negative ($F_N$):** This emerges when the input is 1, but the LLM outputs a 0 ("N/A"), indicating that the LLM either failed to identify the attribute, provided an incomplete response, or incorrectly labeled it as "N/A". It also includes cases where the LLM outputs incorrect or incomplete information for a present attribute.

Employing these definitions, we computed the relevant metrics to assess the LLMs' performance in string segmentation.

### 4.6. Results

In examining the outputs generated by the seven models against the expected attributes, we've conducted a thorough evaluation of each model's performance. This comprehensive assessment not only provides insights into segmentation efficacy but also highlights areas for potential improvement or optimization in our experimental setup. The resultant comparative analysis, detailed in Table 2, serves as a valuable resource for understanding the strengths and limitations of each model in handling segmentation tasks.

| *Model* | *Accuracy* | *Precision* | *Recall* | $F_1$***-score*** |
|---|---|---|---|---|
| *Command R+* | 0.149 | 0.913 | 0.620 | 0.738 |
| *Llama-3* | 0.161 | 0.922 | 0.670 | 0.776 |
| *Mixtral 8x22B* | 0.279 | 0.953 | 0.695 | 0.804 |
| *Sabiá-2-Medium* | 0.304 | 0.904 | 0.756 | 0.823 |
| *GPT-3.5 Turbo* | 0.403 | 0.947 | 0.747 | 0.835 |
| *GPT-4* | 0.422 | 0.953 | 0.787 | 0.862 |
| *Claude 3 Opus* | 0.633 | 0.992 | 0.836 | 0.907 |

**Table 2. Metrics-based Comparative Analysis of Large Language Models for Attribute Segmentation Task**

The highlight among the models was Claude, with an $F_1$-score of approximately 90%. This result underscores the efficiency of the model in segmentation, indicating its ability to accurately extract attributes from descriptions. Similarly, GPT-4 also demonstrated strong performance with an $F_1$-score of 86.2%. Our decision to evaluate the models using a zero-shot prompting approach aimed to provide an impartial assessment of their inherent capabilities without additional training or data supplementation. By refraining from providing specific information to the models, we sought to evaluate their raw ability to comprehend and segment attributes, a crucial aspect in real-world applications where access to comprehensive training data may be limited.

Furthermore, Claude's performance is evidenced by its precision of 0.992 and recall of 0.836. In this context, precision reflects the model's ability to correctly identify relevant attributes with minimal false positives, meaning it successfully avoided generating information not present in the input (i.e., $F_P$). The recall metric indicates Claude's capability to capture the majority of relevant attributes, with few instances where it failed to identify an attribute or incorrectly labeled it as "N/A" (i.e., $F_N$).

Despite these positive results, it is important to address the challenges encountered during the evaluation process. A noteworthy issue observed in the models was their tendency to exhibit "laziness" when dealing with longer descriptions. Instead of annotating the entire content meticulously, the models opted for truncated annotations, adding "..." at the end of descriptions. This behavior, while understandable from a computational standpoint, introduced inaccuracies in segmentation, primarily affecting the accuracy metric, which considers a prediction as correct only when all attributes are accurately identified. As we can see in the Table 3 below, we measured the accuracy of each model concerning each attribute:

| ***Models*** | *Description* | *Volume* | *Concentration* | *Type* | *Active Ingredient* | *Batch* | *Expiration Date* |
|---|---|---|---|---|---|---|---|
| *Command R+* | 0.242 | 0.931 | 0.770 | 0.844 | 0.788 | 0.751 | 0.826 |
| *Llama-3* | 0.322 | 0.975 | 0.745 | 0.745 | 0.844 | 0.844 | 0.900 |
| *Mixtral 8x22B* | 0.434 | 0.987 | 0.757 | 0.801 | 0.844 | 0.869 | 0.881 |
| *Sabiá-2-Medium* | 0.714 | 0.875 | 0.776 | 0.795 | 0.838 | 0.782 | 0.857 |
| *GPT-3.5 Turbo* | 0.664 | 0.968 | 0.788 | 0.813 | 0.869 | 0.788 | 0.869 |
| *GPT-4* | 0.701 | 0.981 | 0.906 | 0.801 | 0.801 | 0.863 | 0.888 |
| Claude 3 Opus | 0.763 | 0.981 | 0.925 | 0.919 | 0.925 | 0.875 | 0.894 |

**Table 3. Model Accuracy for Each Attribute**

Analyzing Table 3, some significant insights can be drawn. Firstly, there is a notable variation in the models' performance across different attributes, highlighting their distinct segmentation abilities. The Claude 3 Opus model demonstrates remarkable accuracy across most attributes, indicating its overall capability for precise segmentation. Specifically, it achieved accuracies of 0.763 for Description, 0.981 for Volume, 0.925 for Concentration, 0.919 for Type, 0.925 for Active Ingredient, 0.875 for Batch, and 0.894 for Expiration Date.

The Mixtral 8x22B achieved the highest accuracy in Volume, with a score of 0.987, underscoring its particular strength in this attribute. Llama-3 also demonstrated notable performance in the Expiration Date attribute, with an accuracy of 0.900. Moreover, both GPT-4 and Sabiá-2-Medium exhibited competitive performance for the Description attribute, reflecting their potential in this area. However, a common challenge across all models was the tendency to truncate annotations in longer descriptions. This limitation impacted accuracy for attributes that require a comprehensive understanding of the full content, highlighting the need for further refinement to enhance consistency, especially in more complex scenarios.

In addition, the models exhibited an overall good average regarding the attributes: volume, concentration, type, and active ingredient. These attributes are key characteristics for the identification and classification of medications. This suggests that the models possess a solid understanding of these fundamental attributes, which are crucial for effective medication analysis and classification tasks.

Finally, our evaluation methodology was designed to provide a comprehensive assessment of the models' capabilities in handling diverse attributes rather than focusing solely on individual attribute accuracy. While accuracy is a crucial factor, we also considered the models' overall effectiveness in managing various types of information and their ability to accurately segment attributes. This holistic approach enabled us to gain valuable insights into each model's overall performance and identify areas for potential

enhancement, ensuring a more nuanced understanding of their strengths and limitations in real-world applications.

## 5. Conclusion and Future Work

This paper presented an evaluation of the ability of LLMs to segment strings in attributes of medication descriptions using a zero-shot prompt approach. The results yielded valuable insights into the performance of different models, with Claude standing out by achieving an impressive $F_1$-score of approximately $90\%$. This score underscores the model's efficiency in segmentation, indicating its capability to accurately extract information from descriptions. However, challenges such as the models' tendency to exhibit "laziness" when dealing with longer descriptions were observed, negatively impacting the accuracy metric.

For future work, we intend to apply our segmentation pipeline to larger datasets, addressing the context window limitations of current LLMs, which affect performance on tasks involving over a million descriptions. To overcome these constraints, we will explore parallel computing strategies, dividing segmentation tasks into smaller, concurrently processed sub-tasks. This approach aims to enhance scalability, optimize the segmentation process, and ensure reliable performance on large-scale data, enabling more robust applications.

## 6. Acknowledgements

The authors thank the Santa Catarina Government Agency for Law Enforcement and Prosecution of Crimes (MPSC) for the financial support to this research, through the project *Céos: Data Intelligence for the Society*.

## References

Almeida, T. S., Abonizio, H., Nogueira, R., and Pires, R. (2024). Sabiá-2: A new generation of portuguese large language models. *ArXiv*, abs/2403.09887.

Aumiller, D., Almasian, S., Lackner, S., and Gertz, M. (2021). Structural text segmentation of legal documents. In *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Law*, ICAIL '21. ACM.

Borkar, V., Deshmukh, K., and Sarawagi, S. (2001). Automatic segmentation of text into structured records. *SIGMOD Rec.*, 30(2):175–186.

Boukhers, Z., Ambhore, S., and Staab, S. (2019). An end-to-end approach for extracting and segmenting high-variance references from pdf documents. In *2019 ACM/IEEE JCDL*, pages 186–195.

Chang, Y., Wang, X., Wang, J., Wu, Y., Yang, L., Zhu, K., Chen, H., Yi, X., Wang, C., Wang, Y., Ye, W., Zhang, Y., Chang, Y., Yu, P. S., Yang, Q., and Xie, X. (2023). A survey on evaluation of large language models.

Chen, X., Marazopoulou, K., Lee, W., Agarwal, C., Sukumaran, J., and Hofleitner, A. (2023). Binary classifier evaluation on unlabeled segments using inverse distance weighting with distance learning. *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*.

Chen, Z., Meng, W., and Dragut, E. C. (2022). Web record extraction with invariants. *Proc. VLDB Endow.*, 16:959–972.

Cruz, P., Vanneschi, L., Painho, M., and Rita, P. (2021). Automatic identification of addresses: A systematic literature review. *ISPRS Int. J. Geo Inf.*, 11:11.

Dorneles, C. F., Gonçalves, R., and dos Santos Mello, R. (2011). Approximate data instance matching: a survey. *Knowledge and Information Systems*, 27(1):1–21.

Géron, A. (2019). *Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O'Reilly.

Haider, W. and Yeşilada, Y. (2022). Classification of layout vs. relational tables on the web: Machine learning with rendered pages. *ACM Transac. on the Web*, 17:1 – 23.

Hoffart, J., Seufert, S., Nguyen, D. B., Theobald, M., and Weikum, G. (2012). Kore: keyphrase overlap relatedness for entity disambiguation. In *Proceedings of the 21st CIKM*, page 545–554, New York, NY, USA. Association for Computing Machinery.

Kayed, M., Dakrory, S., and Ali, A. A. (2021). Postal address extraction from the web: a comprehensive survey. *Artificial Intelligence Review*, 55:1085 – 1120.

Lerman, K., Getoor, L., Minton, S., and Knoblock, C. (2004). Using the structure of web sites for automatic segmentation of tables. In *Proceedings of the 2004 ACM SIGMOD*, page 119–130, New York, NY, USA. Association for Computing Machinery.

Misra, H., Yvon, F., Cappé, O., and Jose, J. (2011). Text segmentation: A topic modeling perspective. *Information Processing & Management*, 47(4):528–544.

Peng, F. and McCallum, A. (2006). Information extraction from research papers using conditional random fields. *Information Processing & Management*, 42(4):963–979.

Rea, L. and Parker, R. (2012). *Designing and Conducting Survey Research: A Comprehensive Guide*. Wiley.

Simon, K. and Lausen, G. (2005). Viper: augmenting automatic information extraction with visual perceptions. In *International Conference on Information and Knowledge Management*.

Uppalapati, V. K. and Nag, D. S. (2024). A comparative analysis of ai models in complex medical decision-making scenarios: Evaluating chatgpt, claude ai, bard, and perplexity. *Cureus*, 16.

Varma, M., Orr, L., Wu, S., Leszczynski, M., Ling, X., and Ré, C. (2021). Cross-domain data integration for entity disambiguation in biomedical text. In *EMNLP*.

Yao, Y., Duan, J., Xu, K., Cai, Y., Sun, Z., and Zhang, Y. (2024). A survey on large language model (llm) security and privacy: The good, the bad, and the ugly. 4(2):100211.

Yoon, J., Gupta, A., and Anumanchipalli, G. K. (2024). Is bigger edit batch size always better? – an empirical study on model editing with llama-3. *ArXiv*.

Zhang, P., Shao, N., Liu, Z., Xiao, S., Qian, H., Ye, Q., and Dou, Z. (2024). Extending llama-3's context ten-fold overnight. *ArXiv*.

Zhang, X., Zou, J., Le, D., and Thoma, G. (2011). A structural svm approach for reference parsing. *BMC bioinformatics*, 12 Suppl 3:S7.