sult in residual issues. To address these, we use CDM's rendering techniques [41] to identify unrenderable elements. These elements are then reviewed and corrected by three researchers to ensure accuracy in the final annotations.

3.3. Dataset Statistics

Page Diversity. OmniDocBench comprises a total of 981 PDF pages across 9 distinct types. Each page is annotated with global attributes, including text language, column layout type, and indicators for blurred scans, watermarks, and colored backgrounds.

Annotation Diversity: OmniDocBench contains over 100,000 annotations for page detection and recognition: (1) More than 20,000 block-level annotations across 15 categories, including over 15,979 text paragraphs, 989 image boxes, 428 table boxes, and so on. All document components except headers, footers, and page notes are labeled with reading order information, totaling over 16,000 annotations. (2) The dataset also includes more than 70,000 span-level annotations across 4 categories, with 4,009 inline formulas and 357 footnote markers represented in LaTeX format, while the remaining annotations are in text format.

Annotation Attribute Diversity: (1) *Text Attributes:* All block-level annotations, except for tables and images, include text attribute tags. In addition to standard Chinese and English text, there are over 2,000 blocks with complex backgrounds and 493 with rotated text. (2) *Table Attributes:* In addition to standard Chinese and English tables, there are 142 tables with complex backgrounds, 81 containing formulas, 150 with merged cells, and 7 vertical tables.

4. OmniDocBench Evaluation Methodology

To provide a fair and comprehensive evaluation for various models, we proposed an end-to-end evaluation pipeline consisting of several modules, including extraction, matching algorithm, and metric calculation, as shown in Figure 4. It ensures that OmniDocBench automatically performs unified evaluation on document parsing, thereby producing reliable and effective evaluation results.

4.1. Extraction

Preprocessing. The model-generated markdown text should be preprocessed, which includes removing images, eliminating markdown tags at the beginning of the document, and standardizing the number of repeated characters.

Elements Extraction. Extraction is primarily carried out using regular expression matching. To ensure that the extraction of elements does not interfere with each other, it is necessary to follow a specific order. The extraction sequence is as follows: LaTeX tables, HTML tables, display formulas, markdown tables (which are then converted into HTML format), and code blocks.

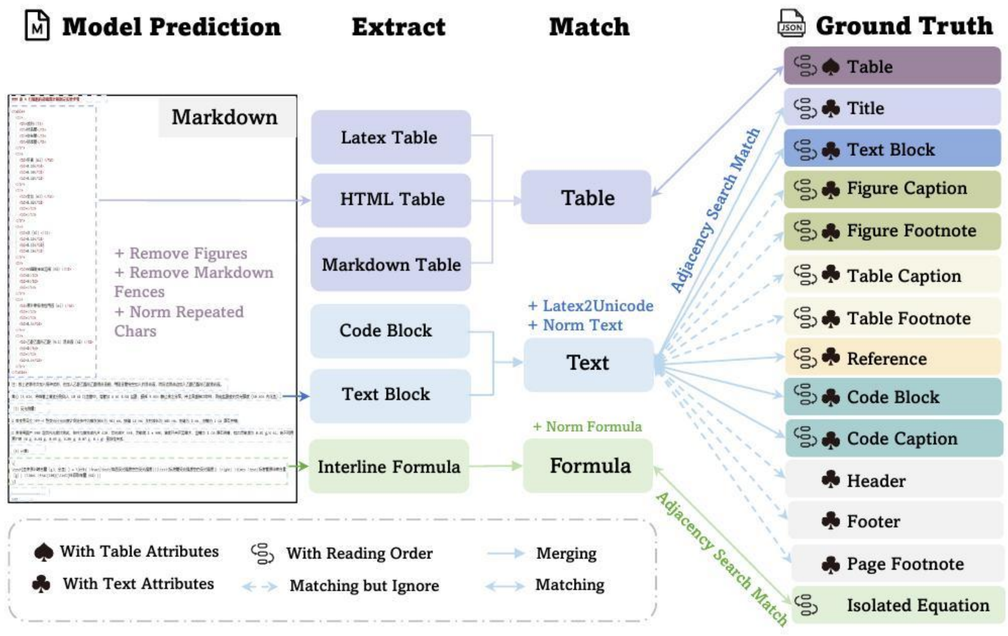

Figure 4. OmniDocBench Evaluation Pipeline.

Pure Text Extraction. After extracting special components, the remaining content is considered pure text. Paragraphs are separated by double line breaks, allowing them to participate in subsequent matching processes, thus aligning with reading order annotation units in the GTs. If no double line break exists, single line breaks are used for paragraph separation. Additionally, previously extracted code blocks are merged into the text category for processing.

Inline Formula Format Converting. We standardized inline formulas within paragraphs to Unicode format. This was necessary because different models produce inconsistent outputs for inline formulas. For formulas originally written in Unicode, it is hard to extract them using regular expressions. Therefore, to ensure a fair comparison, we do not extract inline formulas for separate evaluation. Instead, we include them in their Unicode format alongside the text paragraphs for evaluation.

Reading Order Extraction. Upon completion of the extraction, the start and end positions of the extracted content in the original markdown are recorded for subsequent reading order calculation.

4.2. Matching Algorithm

Adjacency Search Match. To avoid the impact of paragraph splitting on the final results, we proposed Adjacency Search Match, that merges and splits paragraphs in both GTs and Preds to achieve the best possible match. The specific strategy involves: i) Calculate a metrix of Normalized Edit Distance between GTs and Preds. The Pred and GT pairs whose similarity exceeds a specific threshold are considered as successful match. ii) For the rest, we apply fuzzy matching to determine whether one string is a subset of another string. If so, we further apply the merging algorithm which would try to merge adjacent paragraph. This process would continue to merge more paragraph until the Normalized Edit Distance starts to decrease. After this process, the best match will be found for GTs and Preds.

[4] https://mathpix.com/