Task: Simplifying Numbers for Easy Reading

Implement a function in Python that takes a string of raw text (`raw_text`) as input and returns the text with numbers simplified according to the given rules.

Key Details

1. **Main Function Requirement:**
    - Your main function must be named **`simplify_numbers`** .
    - This function should take a single argument, `raw_text` , and return the simplified text.
    - Please stick closely to the requirements to ensure your code works smoothly with the internal test function.
2. **Approach:**
    - Start by using **regex** or similar techniques to identify and replace numbers in the text.
    - You can use additional methods to ensure better accuracy and alignment with the rules.
    - **Note:** You do not need to focus on grammar or perfect sentence structure for now. The priority is implementing the number simplification rules correctly.
    - Your solution should be **smart enough to handle a variety of cases**, as we will be testing it using **internal test cases** in addition to the ones provided.
3. **Simplification Steps:**
    - **Start simple:** Begin with **rounding large numbers** and **interpreting percentages** as described in points 1 and 2 of the German rule.
    - **Gradual refinement:** Once the basic steps are implemented, incorporate **contextual explanations** (point 4) and **visual comparisons** (point 5) to make the simplifications more advanced and relatable.

    Rule: Simple representation of numbers and quantities

    - **1. Round large numbers:**
      Simplify by rounding large numbers to a nearby approximation.
        - Example:
            - `324.620,22 Euro → etwa 325.000 Euro`
            - `38,7 → etwa 40`

- **2. Use comparisons for percentages:**
  Replace percentages with simple comparisons.
  - Example:
    - `25 Prozent → jeder Vierte`
    - `50 Prozent → die Hälfte`
    - `75 Prozent → drei von vier`
- **3. Use simple descriptions for quantities:**
  Replace percentages with descriptive phrases.
  - Example:
    - `60 Prozent → mehr als die Hälfte`
    - `14 Prozent → wenige`
    - `90 Prozent → fast alle`
- **4. Explain the meaning of numbers:**
  Add contextual explanations to help readers understand large numbers better.
  - Example:
    - `1 Million Euro → So viel Geld, dass man 100 Autos kaufen könnte`
    - `10.000 Besucher → So viele Menschen, wie in ein großes Fußballstadion passen`
- **5. Use figurative comparisons:**
  Use visual comparisons to make abstract quantities more relatable.
  - Example:
    - `30 Prozent der Fläche → Ein Drittel, also ein Stück von drei gleich großen Teilen`
    - `250 Kilogramm → So schwer wie ein großer Kühlschrank`

3. **Testing Your Function:**
   Test your function with the following input cases:

```
test_cases = [
    "324.620,22 Euro wurden gespendet.",
    "1.897 Menschen nahmen teil.",
    "25 Prozent der Bevölkerung sind betroffen.",
    "90 Prozent stimmten zu.",
    "14 Prozent lehnten ab.",
    "Bei 38,7 Grad Celsius ist es sehr heiß.",
    "denn die Rente steigt um 4,57 Prozent.",
    "Im Jahr 2024 gab es 1.234 Ereignisse.",
    "Am 1. Januar 2024 waren es 5.678 Teilnehmer.",
    "Im Jahr 2025 gab es 2018 Ereignisse."
]
```

The **expected output** for these test cases is:

```
etwa 325.000 Euro wurden gespendet.
etwa 2.000 Menschen nahmen teil.
jeder Vierte der Bevölkerung sind betroffen.
fast alle stimmten zu.
wenige lehnten ab.
Bei etwa 39 Grad Celsius ist es sehr heiß.
denn die Rente steigt um wenige.
Im Jahr 2024 gab es etwa 1.000 Ereignisse.
Am 1. Januar 2024 waren es etwa 6.000 Teilnehmer.
Im Jahr 2025 gab es etwa 2000 Ereignisse.
```

Additional Information:

- The task must be implemented in **Python**.
- Once completed, deliver your solution by hosting it in a **GitHub repository** and providing the link to abdullah@klao.eu.
- **Internal test cases:** We will evaluate your solution using additional test cases beyond the ones provided. Ensure your function is robust and can handle various edge cases.
- **README File**: It is recommended to create a well-structured **README file** in your GitHub repository that explains your solution, how to set it up, usage instructions, and any other relevant details to help understand your implementation.

Goal:
The goal is to ensure that the output text is clear, simple, and adheres to the provided rules. Use all the examples in the rule to guide your simplifications and ensure consistency. Remember:

- **It is okay if the grammar is not perfect for now.** Focus on correctly implementing the number simplifications.
- **Start with the basics (points 1 and 2), then gradually implement more advanced features (points 4 and 5).**