Both the manual and Copilot versions achieve the same goal — sorting a list of dictionaries by a key — but differ in robustness and readability. The manual function is straightforward, easy to read, and perfectly suited for clear, small-scale data manipulation. It assumes that every dictionary contains the specified key and that values are comparable, which keeps it simple and fast.

The Copilot-generated function, however, introduces **error handling** and **optional reverse sorting**, making it slightly more general and fault-tolerant. By using `.get(key, 0)`, it avoids `KeyError`s if the key is missing, which is useful for messy real-world data. Its `reverse` argument improves usability, and the `try/except` block adds resilience.

From an efficiency standpoint, both rely on Python's built-in `sorted()` function, whose complexity is **O(n log n)**, so performance differences are negligible. The key advantage of the Copilot version lies in **faster prototyping and improved versatility** — the AI auto-added practical safeguards that a human might forget under time pressure. Nevertheless, developers must always **verify Copilot's logic and error handling** to ensure it aligns with project requirements and data constraints.