To understand the quality of the produced knowledge graph as you increase the number of documents considered, let's first examine the different implementations provided in the code snippets.

Implementation 1 - Extracting Relations from Short Text: In the first implementation, the code processes a short text by tokenizing it and then generating relations from the model's output using a transformer-based language model (seq2seq). The function **from_small_text_to_kb** takes a short text as input, generates possible relations using the model, and constructs a Knowledge Base (KB) containing the extracted relations.

Implementation 2 - Split Spans from Long Text to Knowledge Base: The second implementation extends the approach to handle longer texts by splitting them into smaller spans. This allows the model to generate relations for each span, and the resulting KB is built by merging relations with the same head, type, and tail while preserving the metadata about the spans.

Implementation 3 - Extract Knowledge Base from Web Article: The third implementation goes a step further by extracting knowledge from a web article. It uses the same approach as the previous implementation but adds an extra step of fetching an article from a given URL using the newspaper library. The extracted article's text is then processed similarly to the previous implementation to generate relations and construct the KB. Additionally, this implementation introduces entities, which are extracted from Wikipedia to enrich the KB with more information about the entities involved in the relations.

Now, I will discuss how to improve the quality of the Knowledge Base using specific ideas:

Entity Linking and Disambiguation: One way to improve the quality of the Knowledge Base is to enhance the entity extraction process. Currently, the code fetches information about entities from Wikipedia based on their titles. However, entity linking and disambiguation can be used to accurately identify entities and link them to their corresponding Wikipedia articles. This would prevent the occurrence of unrelated entities with similar titles and improve the overall quality and accuracy of the Knowledge Base.

Relation Extraction from Context: The current approach generates relations independently for each span, which may result in incomplete or fragmented relations. A better approach would be to consider the context of the entire text when extracting relations. This means generating relations that involve entities across multiple spans to capture the full context of relationships in the text. It would lead to a more comprehensive Knowledge Base with relations that reflect the interactions between entities in a coherent manner.

Coreference Resolution: Coreference resolution can be employed to improve the representation of entities and relations in the Knowledge Base. In some cases, entities may be referred to by different expressions or pronouns throughout the text. Coreference resolution would identify these references and link them to the appropriate entities, ensuring that all occurrences of an entity in the text are associated with the correct entry in the Knowledge Base. This would improve the KB's consistency and reduce redundancy.

By incorporating these ideas, the overall quality and accuracy of the Knowledge Base can be significantly improved. Entity linking and disambiguation, relation extraction from context, and coreference resolution are essential techniques to ensure that the Knowledge Base captures the most relevant and coherent information from the input documents.