1. What are the different kinds of similarity search that can be used to find the answer closest to the question in meaning. What similarity metrics does each of them use for comparison?

Before I dive right into the answer, let me talk about the title(Semantic Search). Semantic search is a data searching technique that focuses on understanding the contextual meaning and intention behind a user's search query, rather than only matching keywords. Instead of justlooking for literal matches between search queries and indexed content, it focuses on delivering more relevant search results by considering different factors, such as the relationships between words, the searcher's location, any previous searches, and the context of the search.

Now that we've looked at Semantic Search, let's turn our gears and look at some other types of searches and what type of metrics they use.

1. **Keyword Search**: is a searching technique that focuses more so on finding exact matches between the keywords in a query and the keywords in a document rather than the intent and the context behind it.

2. **Lexical search**: is similar to Keyword Search in a way that they both focus on the similarity of the words rather than the intent or context behind it. Lexical Search looks for and matches similarities in the words(not only keywords) in their literal form.

3. **Contextual Search**: expands upon traditional search by taking into account the user's context, such as their location, and past interactions. It is similar to Semantic Search but rather than deciphering the intrinsic meaning of the query it focuses on the external clues about the user.

2. What is the solution to mitigate the slowness of nearest neighbor similarity calculation.

- **Approximate Nearest Neighbor (ANN) Algorithms**
  Use algorithms like Locality-Sensitive Hashing (LSH) or KD-Trees to quickly find approximate nearest neighbors instead of exact matches. This method is very significant in that it speeds up search time with a trade-off in accuracy.

- **Index Structures**
  Utilizing spatial data structures like R-Trees, Ball Trees, or VP-Trees to organize data points efficiently. This method allows for faster retrieval of nearest neighbors by limiting the search space.
- **Batch Processing**
  Instead of processing one query at a time, process multiple queries in batches. This in turn will reduce overhead and can leverage parallel processing.

3. How can we fix the potential problem of searching with similarity and how can we make the search better?

To make the search better we can use different Techniques and Methods.

- **Better Data Representation**
  Use embeddings (e.g., Word2Vec, BERT) to capture semantic meanings more effectively than traditional methods. Or for example using vector representations instead of literal words and reducing words to their root form. This will result in more relevant similarity comparisons based on meaning rather than surface-level features.

- **Hybrid Models**
  Combine multiple similarity metrics to leverage their strengths. The weakness of one metric can be covered by the other metrics so, together we can have more relevant search results. The benefit of this technique is that it provides a more nuanced understanding of similarity, improving overall search quality.

- **Feedback Loops**
  Implementing user feedback mechanisms to refine and adjust the search results based on user interactions. Collecting user feedback will greatly help us in our future search results. We can have a more refined or give a result that the user needed in the first place by learning from the feedback. This will help because daptive learning from user behavior can continuously improve the relevancy of results.