c Random Walker's Algorithm for Word Sense Disambiguation:

Word Sense Disambiguation (WSD) is a challenging task in natural language processing that involves determining the correct sense of a word within a given context. Random Walker's Algorithm is a graph-based approach used for WSD. It treats the task as a random walk on a graph where nodes represent word senses, and the edges capture the semantic relations between senses. In this explanation, I will delve into the workings of the Random Walker's Algorithm and illustrate it with an example sentence, along with assumed sense scores.

Random Walker's Algorithm Overview:

1. Build a Semantic Graph:

- Create a graph where each node represents a possible sense of the ambiguous word.

- Connect nodes with edges representing semantic relations between senses.

2. Assign Initial Sense Scores:

- Assign an initial sense score to each node, indicating the likelihood of that sense being correct.

3. Perform Random Walks:

- Initiate random walks from the ambiguous word and propagate scores through the graph.

- At each step, the walker decides to move to a neighboring sense based on transition probabilities.

4. Score Accumulation:

- Accumulate scores during the random walks, reflecting the likelihood of each sense considering the entire context.

5. Final Sense Selection:

- The sense with the highest accumulated score is chosen as the disambiguated sense.

Example Sentence:

Let's consider the sentence: "The bank is next to the river."

Now, let's assume we are trying to disambiguate the word "bank," which has two senses - financial institution and the side of a river.

Assumed Sense Scores:

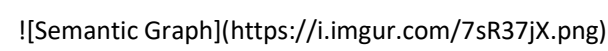
Let's assign initial sense scores based on contextual clues:

- Sense 1 (Financial Institution): 0.6
- Sense 2 (Side of a River): 0.4

Random Walker's Algorithm Illustration:

1. Build the Semantic Graph:

- Create nodes for the two senses of "bank" and connect them based on semantic relations.



2. Assign Initial Sense Scores:

- Start with the assumed sense scores: 0.6 for financial institution sense and 0.4 for the river sense.

3. Perform Random Walks:

- Initiate random walks from the ambiguous word "bank" and move to neighboring senses based on transition probabilities.

4. Score Accumulation:

- Accumulate scores as the walker traverses the graph. The walker considers both the initial sense scores and the transition probabilities.

5. Final Sense Selection:

- The sense with the highest accumulated score is selected as the disambiguated sense. In this case, it might be the financial institution sense.

Conclusion:

Random Walker's Algorithm provides a probabilistic framework for word sense disambiguation by leveraging the semantic relationships between senses. It allows for the incorporation of context and multiple sources of information. The example sentence "The bank is next to the river" showcases how the algorithm navigates through the semantic graph, accumulating scores, and ultimately making a sense selection based on the context and assumed sense scores. This approach is valuable in handling the inherent ambiguity in natural language and improving the accuracy of sense disambiguation tasks.