Webiks Home Assignment:

Link to the chosen model: https://spacy.io/models/en#en_core_web_lg

1. What does the output represent in regard to the input?

The output of the calculate_embedding() function is a vector embedding (special list of numbers) that numerically represents the input word's semantic and syntactic characteristics as understood by the word2vec model. This vector is a point in a high-dimensional space where similar words are located closer to each other.

Meaning, if two words are similar in meaning or use, their lists of numbers will be more alike and 'closer' to each other in this number space.

Here are three examples to illustrate this:

- Input: "cat"

  Output: Vector (e.g., [0.12, -0.49, ...])

  Explanation: The vector for "cat" captures features relevant to animals, pets, etc. The model likely associates "cat" with other domestic animals and common verbs or adjectives related to them.

- Input: "happy"

  Output: Vector (e.g., 2.13, -0.74, ...])

  Explanation: The vector for "happy" encodes its association with positive emotions, situations where happiness is typically mentioned, and possibly its synonyms.

- Input: "apple"

  Output: Vector (e.g., [-0.58, 0.76, ...])

  Explanation: The vector for "apple" represents features related to fruit, food, or even technology (due to the brand "Apple"). The model might relate "apple" to other fruits, healthy eating, or Apple products.

2. How does this representation illustrate semantic distance?

Semantic distance refers to the degree of semantic similarity between words, quantified as the distance between their vectors in the embedding space. So basically, semantic distance is like measuring how close or far apart the meanings of words are. It's done by looking at their word vectors. If two words mean similar things or are used in similar ways, their vectors are close to each other. If their meanings are very different, their vectors are farther apart. Words with similar meanings or used in similar contexts are closer together, while dissimilar words are farther apart.

For example:

"Cat" and "dog" are both pets and animals, so their vectors would be closer in the embedding space, indicating a smaller semantic distance.

"Cat" and "technology" are unrelated concepts, so their vectors would be farther apart, indicating a larger semantic distance.

This can be measured using metrics like cosine similarity.

This idea helps to understand small differences in meaning between words. For example, "large" and "big" are almost the same (they are synonyms), but "large" and "huge" are similar yet not exactly the same. "Huge" suggests something bigger than "large," showing a little difference in how big something is.

3. How do the vector embeddings enable semantic search?

Semantic search means searching for words based on their meaning, not just the exact words used. Word vectors help with this by letting the search tool understand how close the meanings of different words are to each other.

For example, in a semantic search:

If you search for "Japanese dish", you might get results that include "sushi" or "ramen", even if the exact phrase "Japanese dish" isn't used. This is because "sushi" and "ramen" are closely related to the concept of Japanese cuisine in the word vector space.

A search for "feline pet" could return documents containing "cat" because "feline" and "cat" are semantically close in the embedding space.

Searching for "happiness" might also surface documents containing "joy," "happy," or even "smiling," as these terms are semantically related.

This approach contrasts with traditional keyword-based search, which would only find exact matches. Semantic search enabled by vector embeddings is more flexible and effective in understanding the intent and context of the search query, leading to more relevant and comprehensive search results.