Word embeddings are a type of word representation that allows words to be represented as vectors in a continuous vector space. These vectors are learned in a way that captures semantic meanings, relationships, and context of words. Unlike traditional representations that might use one-hot encoding, where each word is represented as a unique vector in a very high-dimensional space (with dimensions equal to the size of the vocabulary), word embeddings map words into a much smaller, dense vector space.

The key characteristics of word embeddings include:

1. **Dimensionality Reduction**: Word embeddings reduce the dimensionality of the word representation, making the vector representation much more manageable compared to one-hot encoding.

2. **Semantic Similarity**: Words that are semantically similar are placed closer to each other in the embedding space. This means that the geometric distance (e.g., cosine similarity) between vectors of words like "king" and "queen" would be smaller than the distance between "king" and "apple".

3. **Contextual Understanding**: Many word embedding techniques capture contextual nuances, meaning that words used in similar contexts will have similar embeddings. However, the extent to which context is captured can vary significantly between different types of embeddings. Advanced models like BERT and ELMo generate context-dependent embeddings, where the representation of a word changes based on its sentence context.

4. **Training**: Word embeddings are usually obtained by training models on large text corpora. The training process involves predicting a word from its context (or vice versa), which forces the model to learn semantic and syntactic features of the language. Common models for generating word embeddings include Word2Vec, GloVe, and FastText.

5. **Applications**: Word embeddings are foundational in many natural language processing (NLP) tasks, such as text classification, sentiment analysis, machine translation, and more. They enable models to process text data in a more nuanced and efficient way.

Two popular algorithms for generating word embeddings are:

- **Word2Vec**: Developed by a team led by Tomas Mikolov at Google, it offers two architectures: Continuous Bag of Words (CBOW) and Skip-Gram. CBOW predicts a target word from a window of surrounding context words, while Skip-Gram does the inverse, predicting context words from a target word.

- **GloVe (Global Vectors for Word Representation)**: Developed by researchers at Stanford, GloVe is trained on global word-word co-occurrence statistics from a corpus. The model essentially captures the probabilities that two words appear together.

Word embeddings have significantly advanced the field of NLP, enabling more nuanced, efficient, and effective processing of text data.