Justification for Using Sentence-Transformers

For the task of identifying duplicate or highly similar job postings based on their descriptions, **sentence-transformers** (specifically using a pre-trained model like all-MiniLM-L6-v2) is the recommended approach. Here's why:

1. **Optimized for Semantic Similarity:** Unlike word embedding models (GloVe, CBOW/Word2Vec) which represent individual words, sentence-transformers models are built upon transformer architectures (like BERT, RoBERTa) and are specifically fine-tuned to generate embeddings for entire sentences or paragraphs. These embeddings capture the semantic meaning of the text, considering word order and context. This is crucial for understanding the nuances of job descriptions, which are often longer than single sentences. Comparing these document-level embeddings using metrics like cosine similarity directly reflects semantic relatedness.
2. **Ease of Use:** The sentence-transformers library provides a very straightforward API. Loading a state-of-the-art pre-trained model and generating embeddings for a list of texts can be done in just a few lines of code. This contrasts with potentially more complex pipelines required for averaging GloVe/CBOW vectors or setting up some spaCy configurations.
3. **High Performance Pre-trained Models:** The library offers access to numerous models pre-trained on vast datasets and fine-tuned for similarity tasks. Models like all-MiniLM-L6-v2 offer a great balance between performance (quality of embeddings) and computational efficiency (speed and model size). They often outperform simple averaging of word embeddings or general-purpose language model embeddings that haven't been specifically tuned for similarity.
4. **Suitability for Job Descriptions:** Job descriptions contain specific terminology, requirements, and responsibilities. Transformer-based models used by sentence-transformers are better equipped to understand the context and relationships between these terms compared to methods that treat words more independently. This leads to more accurate similarity comparisons between postings.

Comparison to Alternatives:

- **GloVe/CBOW:** Require averaging word vectors, losing word order and contextual information. Less effective for capturing the overall meaning of a multi-sentence job description.
- **spaCy:** While powerful, using its basic word vectors has similar limitations to GloVe/CBOW. Using spaCy's transformer integration is possible but adds complexity compared to the direct sentence-transformers library for this specific

task.

Therefore, sentence-transformers provides the best combination of performance, ease of use, and suitability for generating meaningful embeddings for job description similarity within the project's scope and constraints.