[Semantic Chunking](https://www.linkedin.com/in/pramodith?miniProfileUrn=urn%3Ali%3Afs_miniProfile%3AACoAABtfu2YBCtU4w9ZkjLKdSLI2oY1qzP05m8o" \t "_self)

[Pramodith B.Pramodith B.• 2.• 2.AI Engineer @ LinkedIn | Posts about AIAI Engineer @ LinkedIn | Posts about AI](https://www.linkedin.com/in/pramodith?miniProfileUrn=urn%3Ali%3Afs_miniProfile%3AACoAABtfu2YBCtU4w9ZkjLKdSLI2oY1qzP05m8o)

[4 Std. • 4 Std. •](https://www.linkedin.com/in/pramodith?miniProfileUrn=urn%3Ali%3Afs_miniProfile%3AACoAABtfu2YBCtU4w9ZkjLKdSLI2oY1qzP05m8o" \t "_self)

Folgen

[🎆 Semantic Chunking is a new chunking strategy in LangChain that can be used in your RAG workflows.  
  
🔍 What’s the problem?  
  
Chunking is the process of splitting a large document into multiple segments so that we can embed each segment and retrieve the most relevant segments in RAG. However, most chunking strategies are static in terms of character length or using a pre-defined list of separators.  
  
🧩 Semantic Chunking  
  
In semantic chunking, the goal is to leverage how the content in a document changes over a sliding window of text and identify the major change points.  
  
🖥 Algorithm  
  
1. Choose a window size, let’s say W = 3. We now split the document into groups of 3 sentences, with a sliding window stride of 1 (similar to CNN’s). So if my document had 5 sentences in total, we’d have 3 groups A = (1,2,3), B = (2,3,4), and C = (3,4,5).  
  
2. Compute the embedding of each group.  
  
3. Compute semantic similarity scores of each consecutive pair. So you’d compute sim(A,B), sim(B,C).  
  
4. Now here’s where you can use multiple strategies to identify the split points for your document. A couple of them are percentile-based and standard deviation-based.  
  
5. Since you have a list of scores, you can identify the pth percentile of the highest distance scores or the points at which the standard deviation exceeds a certain value.  
  
6. If in our example we see that the distance between group B and C meets the above criterial, we'll choose to split our document on sentence 2. So our vector store will save the embeddings for Chunk C1 = (1,2) and Chunk C2= (3,4,5).  
  
Wrap Up 👏  
  
The idea was originally explored by Gregory Kamradt in his notebook here, which covers a bunch of other chunking strategies too: [Retrieval Tutorials Notebook](](https://www.linkedin.com/feed/update/urn:li:activity:7169676809263915008/)<https://lnkd.in/e5HjKcRf>)  
  
Langchain docs: [Semantic Chunker Documentation](<https://lnkd.in/etEKBHiS>)  
  
The image is from the notebook.  
  
[#rag](https://www.linkedin.com/feed/hashtag/?keywords=rag&highlightedUpdateUrns=urn%3Ali%3Aactivity%3A7169725984529666048) [#generativeai](https://www.linkedin.com/feed/hashtag/?keywords=generativeai&highlightedUpdateUrns=urn%3Ali%3Aactivity%3A7169725984529666048)

Übersetzung anzeigen

Größere Bilddarstellung aktivieren,

