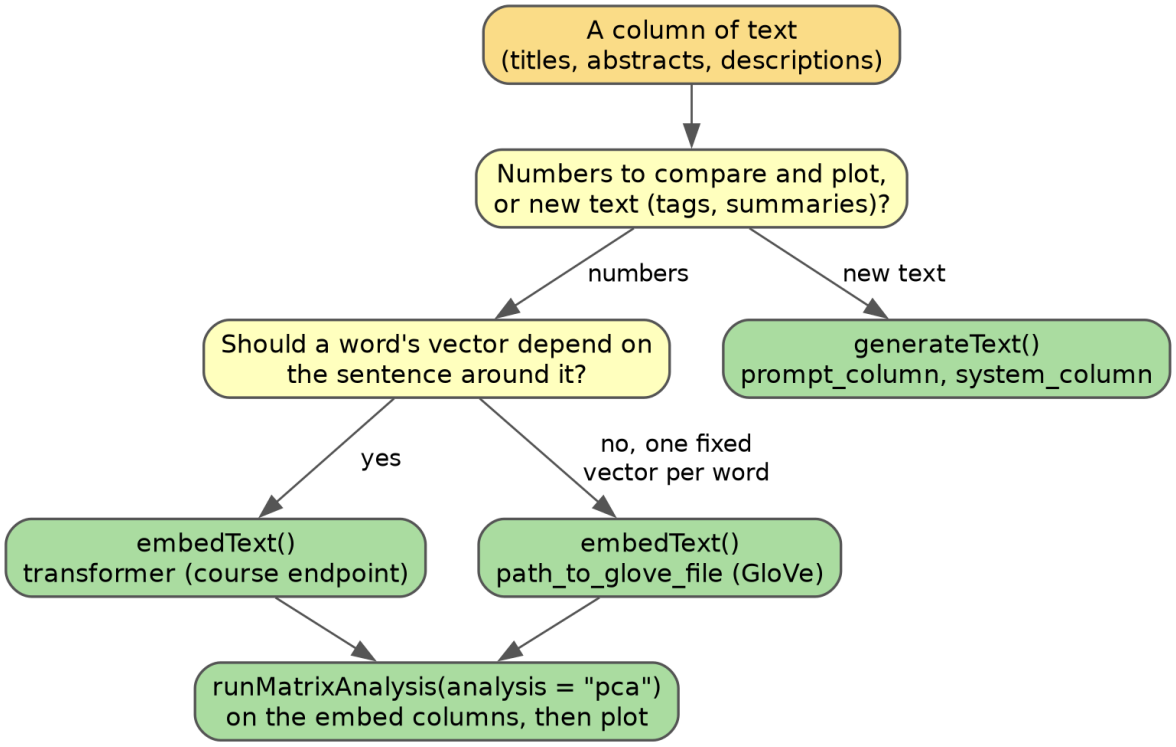
language models: summary

Integrated Bioanalytical Chemistry · thebustalab.github.io/integrated_bioanalytics



Language models turn text into numbers. An embedding model converts each title, abstract or description into a vector (384 numbers for the course transformer model), and texts with similar meanings get similar vectors, so familiar tools such as PCA can show which ones group together. The two embedding methods differ in how they treat context: GloVe gives every word one fixed vector and averages the word vectors over a sentence, whereas a transformer adjusts each word's vector to fit the sentence around it. A generative model builds on the same machinery to write new text in response to a prompt and a system message, and it is much slower and more expensive than embedding because each row is a separate request to a large model. Keys and tokens live in `.Renviron` and are read with `Sys.getenv()`, never typed into code.

Function	What it is for	Arguments varied in this chapter
`searchPubMed()`	retrieve PubMed records (title, abstract, journal) for a set of search terms	`search_terms`, `pubmed_api_key = Sys.getenv("PUBMED_TOKEN")`, `retmax_per_term`, `sort` (`"relevance"` or `"date"`)
`embedText()`, transformer	contextual embeddings from the course endpoint (BAAI/bge-small-en-v1.5, 384 columns)	`df`, `column_name` (the class token is picked up from `.Renviron`)
`embedText()`, GloVe	one fixed vector per word, averaged over each sentence	`column_name`, `path_to_glove_file = Sys.getenv("GLOVE_FILE")`
`runMatrixAnalysis()`	PCA on the embedding columns to see which texts group together	`analysis = "pca"`, `columns_w_values_for_single_analyte` (the columns found by `grep("embed", ...)`), `columns_w_sample_ID_info`
`generateText()`	new text (here, three tags per abstract) from the course generation endpoint	`df`, `prompt_column`, `system_column`