Deep Learning: Automatic Text Summarization & Concept Generation

There are broadly two approaches to automatic text summarization: ***extractive*** *and* ***abstractive***

Extractive approaches select passages from the source text, then arrange them to form a summary

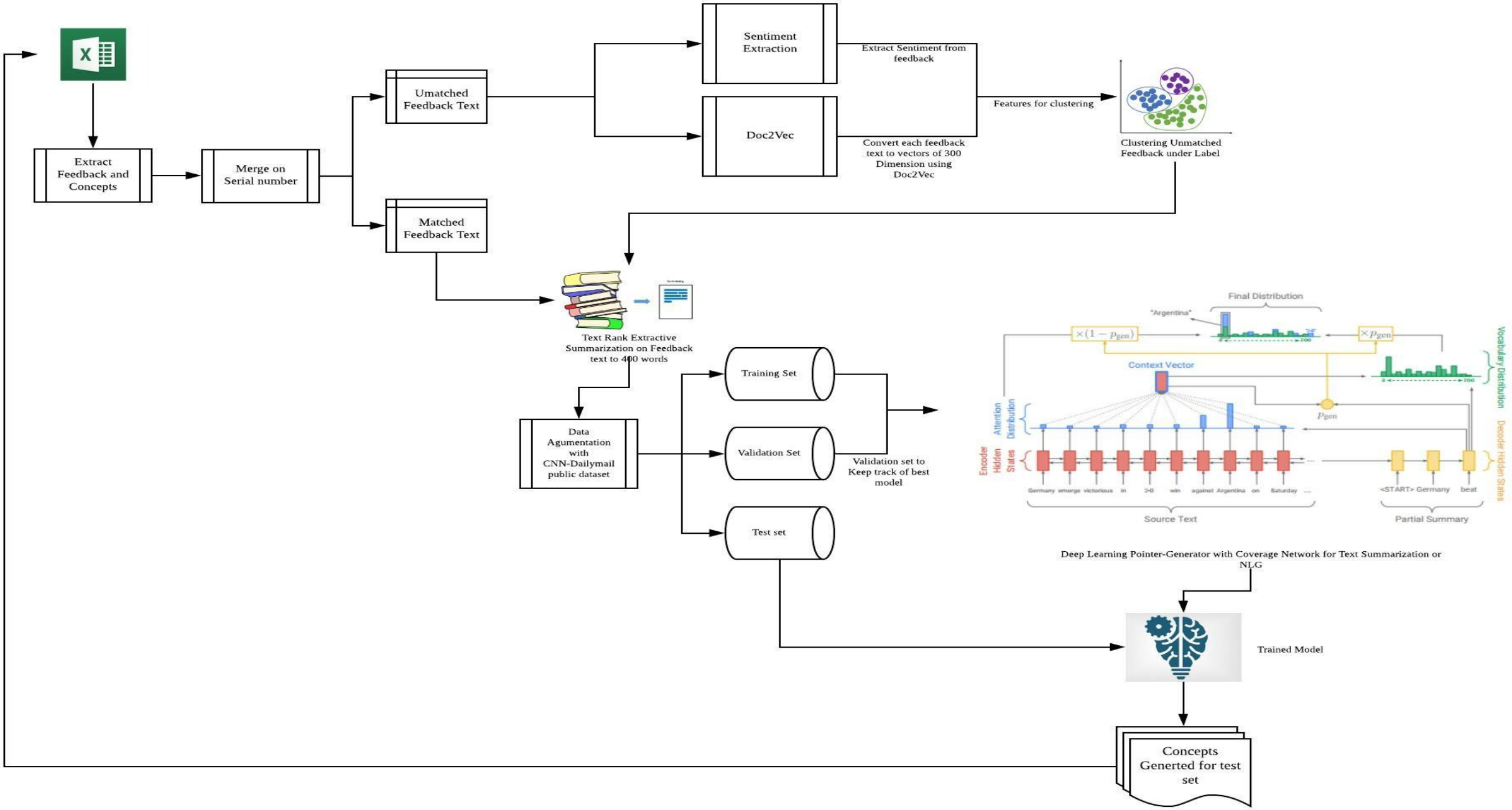
You might think of these approaches as like a highlighter.

Abstractive approaches use natural language generation techniques to write novel sentences.

By the same analogy, these approaches are like a very smart pen.

Concept Generation Process Flow

Environment: GCP VM with 64 GB RAM, 8 vCPU and 1 Tesla P100 GPU



Concept Generation Process Flow

Preprocessing	Extractive Summarization	Data Augmentation	Translation	Model Training/Evaluation	Model Scoring
Transpose concepts into row format I.e. one concept per serial number.	Text Rank Extractive summarization to reduce voluminous feedback text to 400 words.	Add CNN/Daily-mail 300K publicly available article-summary files to private feedback data to train a summarization model.	Convert tokens to word ids	Build a Pointer-Generator Deep Learning Summarization model	Using the best model from a series of iterations of pointer-generator models, Generate an abstractive summary from the unseen new feedback text.
Merging Feedback and Concepts on Serial Number to build a Feedback-Concept pair.		Tokenize the text	Translate word ids to Glove 300 dimension word embeddings using word lookup	Train the model using the text converted to numbers until the model is converged.	
Aggregate feedback text on label and concepts. Aggregate feedback text on label alone where there is no concept generated.		Build vocabulary file on the all of the feedback text.		Evaluate the model parallelly, using the model trained so far to keep track of best model.	
Cluster feedback text under label to form clusters of feedback for each label that don't have concepts generated.					