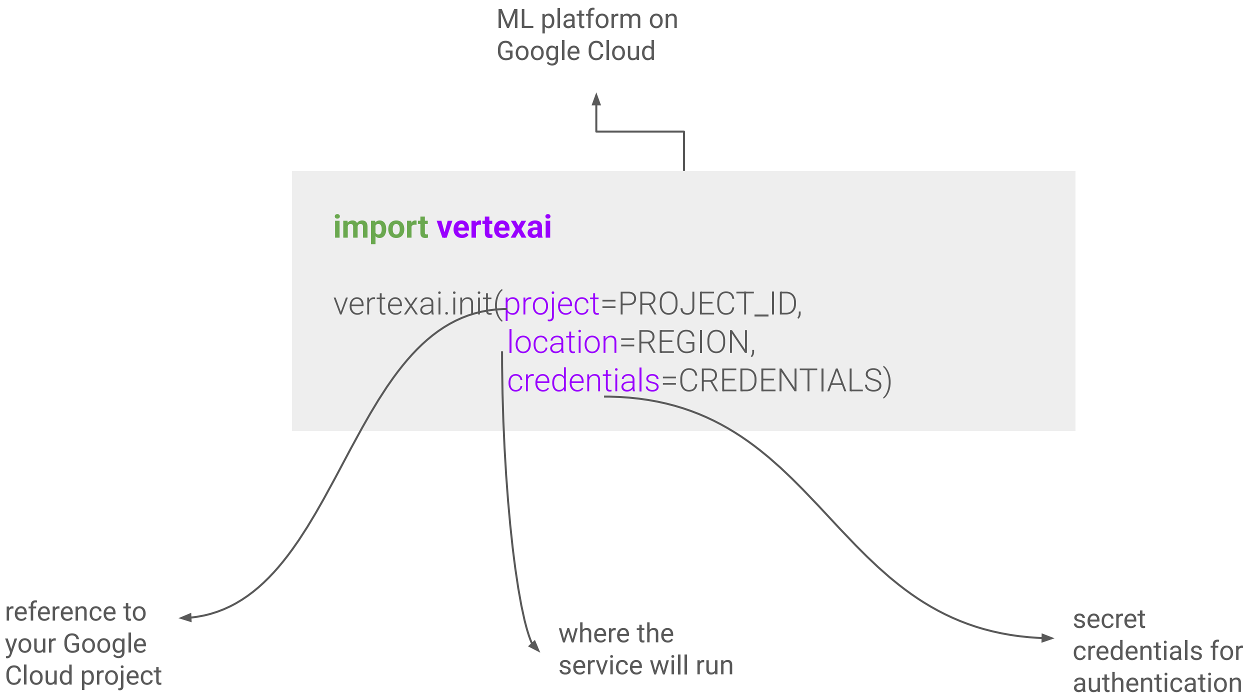
Understanding and Applying Text Embeddings

Set up Google Cloud



ML platform on Google Cloud

```python
import vertexai

vertexai.init(project=PROJECT_ID,
              location=REGION,
              credentials=CREDENTIALS)
```

reference to your Google Cloud project

where the service will run

secret credentials for authentication

Specify a model, Get an Embedding

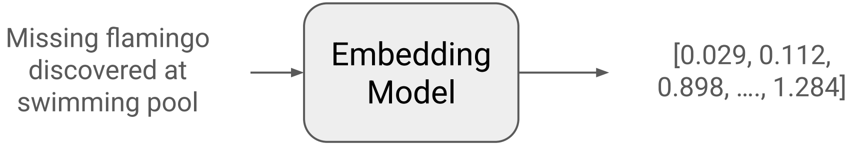
```
from vertexai.language_models import TextEmbeddingModel
```

```
embedding_model = TextEmbeddingModel.from_pretrained(
“text-embedding-005”)
```

```
embedding = embedding_model .get_embeddings(
[“What is the meaning of life?”])
```

What is an embedding

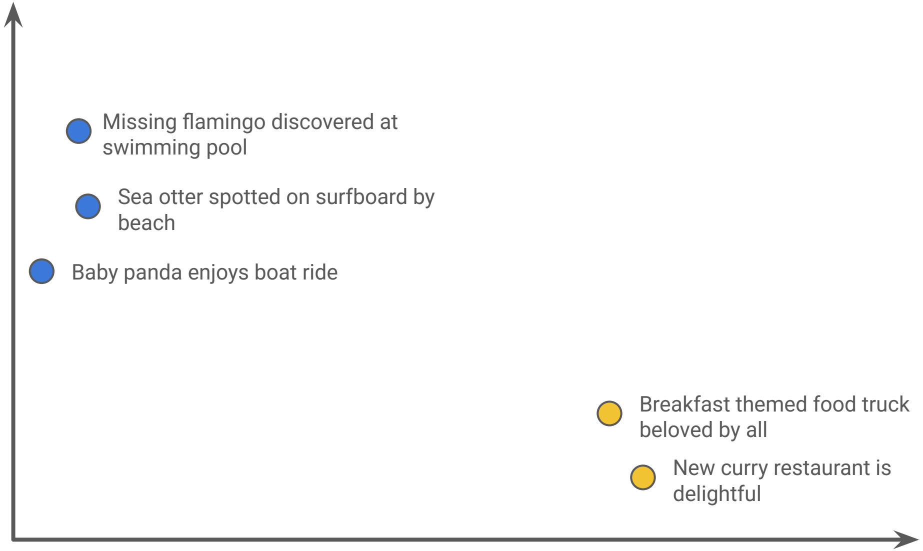
A way to represent data as points in space where the locations are semantically meaningful



Missing flamingo discovered at swimming pool

Embedding Model

[0.029, 0.112, 0.898, …., 1.284]



Missing flamingo discovered at swimming pool

Sea otter spotted on surfboard by beach

Baby panda enjoys boat ride

Breakfast themed food truck beloved by all

New curry restaurant is delightful

How are sentence embeddings computed

- **Simple Method:**
  - Embed each word separately, and take a sum or mean of all the word embeddings

- **Modern Embeddings:**
  - Use a transformer neural network to compute a context-aware representation of each word, then take an average of the context-aware representations
  - Compute embeddings for each token (e.g., sub-word) rather than word. Enables algorithm to work even for novel words and misspelt words (“Life the unverse and everything”).

- **Training the transformer network (contrastive learning)**
  - Given a dataset of pairs of “similar” sentences, tune neural network to move similar sentences embeddings together and dissimilar sentences’ embeddings apart.

Specify a mode, Generate Text

```
from vertexai.generative_models import GenerativeModel
```

```
model = GenerativeModel(“gemini-2.5-flash”)
```

```
prompt = “Recommend me a programming activity to improve my skills."
response = model.generate_content(contents=prompt)
```

Decoding Strategies

Text Input

The garden was full of beautiful

Probabilities over tokens

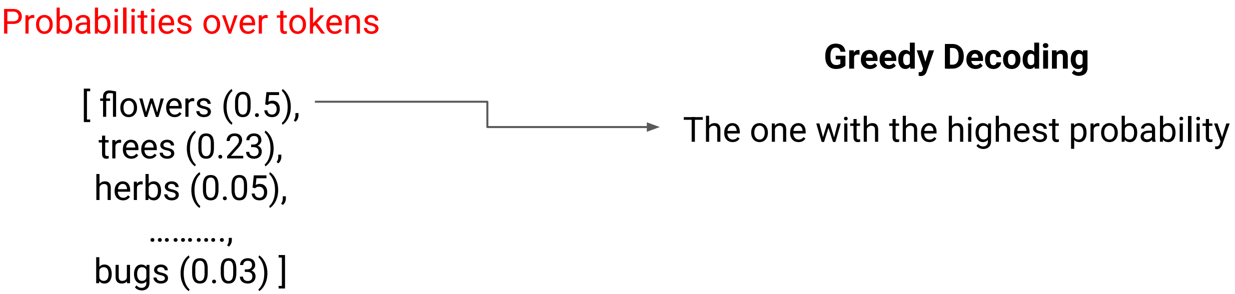
[flowers (0.5),
trees (0.23),
herbs (0.05),
……….,
bugs (0.03)]

Which one should you choose?

Decoding Strategies

Text Input

The garden was full of beautiful flowers



Probabilities over tokens

[flowers (0.5),
trees (0.23),
herbs (0.05),
……….,
bugs (0.03)]

Greedy Decoding

The one with the highest probability

Decoding Strategies

Text Input

The garden was full of beautiful bugs

Probabilities over tokens

[flowers (0.5),
trees (0.23),
herbs (0.05),
……….,
bugs (0.03)]

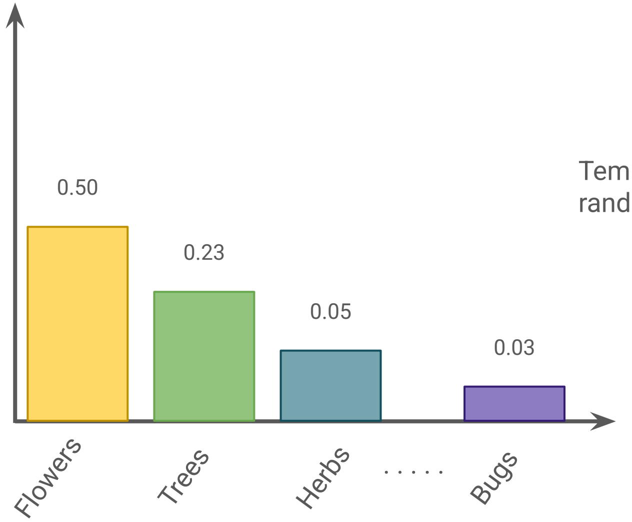
Random sample

Use the probabilities to sample a random token

Temperature

Text Input

The garden was full of beautiful



Temperature controls the randomness

Temperature

Word	Logits	Softmax	Softmax with temperature 0.1
flowers	20	**0.881**	**1.000**
trees	18	**0.119**	0.000
herbs	5	0.000	0.000
bugs	2	0.000	0.000

Softmax

$$Softmax_{\theta}(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{n} e^{z_i}}$$

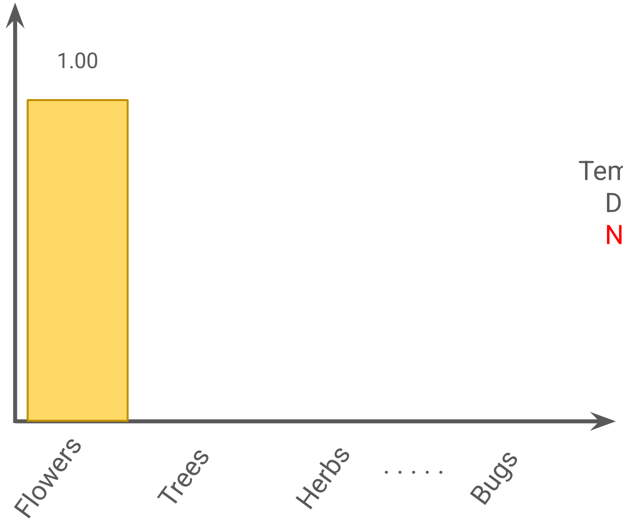
Softmax with temperature θ

$$Softmax_{\theta}(z_i) = \frac{e^{\frac{z_i}{\theta}}}{\sum_{j=1}^{n} e^{\frac{z_i}{\theta}}}$$

Temperature

Text Input

The garden was full of beautiful

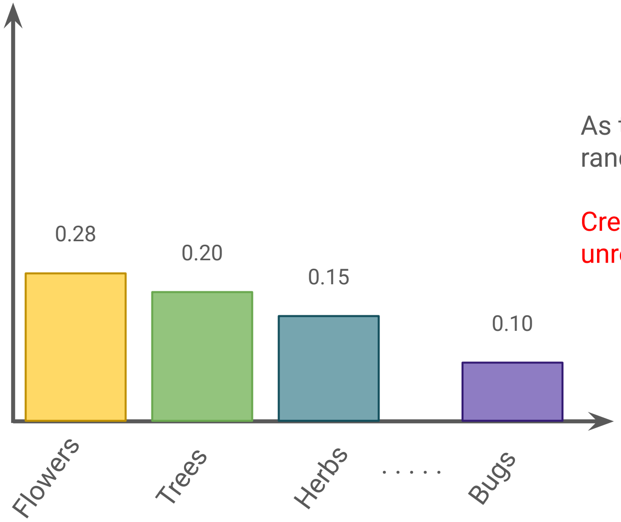


Temperature of 0:
- Deterministic choice
- Not creative but reliable

Temperature

Text Input

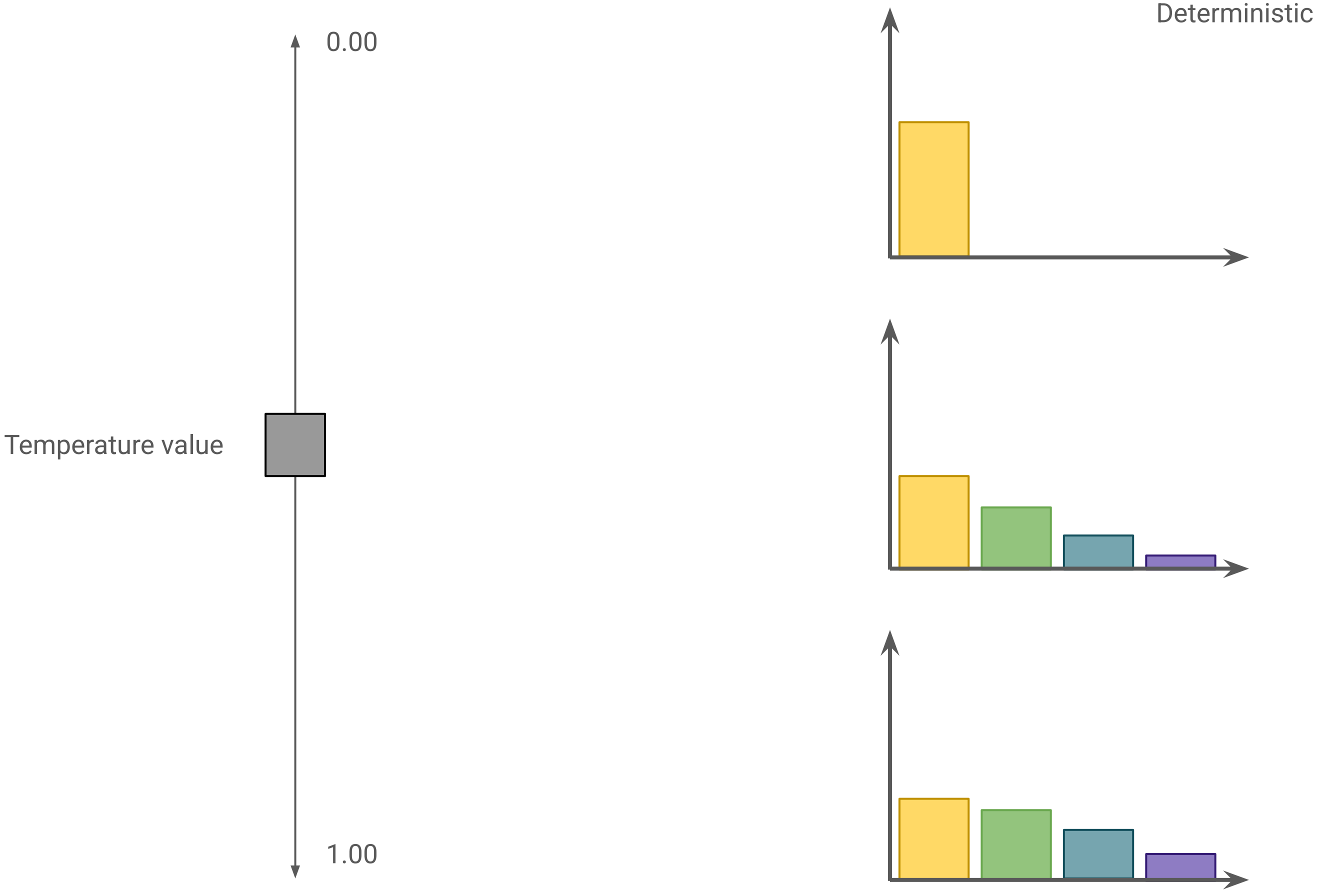
The garden was full of beautiful



As temperature increases so does randomness

Creative but sometimes unreasonable

Temperature



Top K

Text Input

The garden was full of beautiful

Probabilities over tokens

[flowers (0.5),
trees (0.23),
herbs (0.05),
……….,
bugs (0.03)]

Sample from tokens with top k probabilities

Top K

Text Input

The garden was full of beautiful

Probabilities over tokens

[flowers (0.5),
trees (0.23),
herbs (0.05),
……….,
bugs (0.03)]

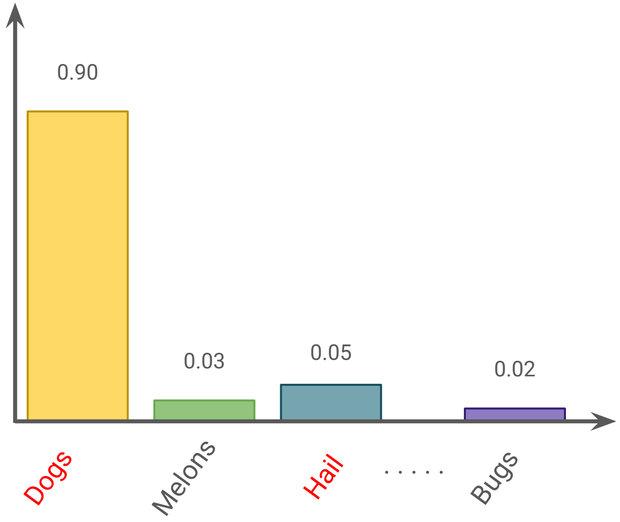
k = 2

Sample from tokens with top k probabilities

Top K

Text Input

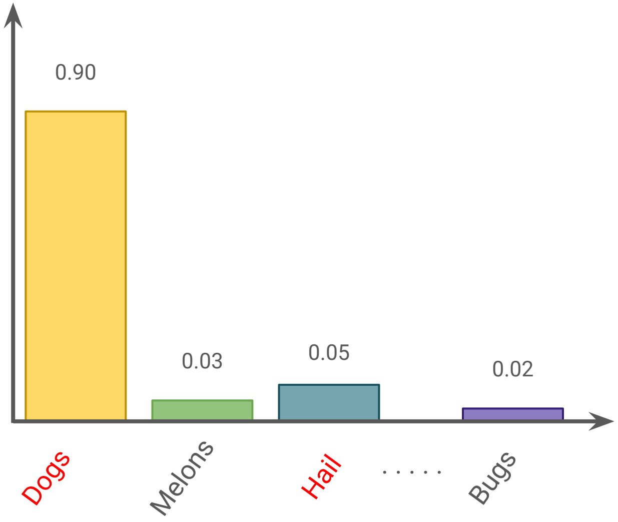
It is raining cats and ______



Top K

Text Input

It is raining cats and hail



Top P

Text Input

The garden was full of beautiful

Probabilities over tokens

[flowers (0.5),
trees (0.23),
herbs (0.05),
……….,
bugs (0.03)]

Sample from minimum set of tokens whose cumulative probability is greater or equal to P

Top P

Text Input

The garden was full of beautiful

Probabilities over tokens

[flowers (0.5),
trees (0.23),
herbs (0.05),
………..,
bugs (0.03)]

P = 0.75

Sample from minimum set of tokens whose cumulative probability is greater or equal to P

Putting it all together

Text Input

The garden was full of beautiful

[flowers, trees, herbs, plants, _ _ _, weeds, bugs]

Top K

[flowers, trees, herbs, plants]

Top P

[flowers, trees]

Temperature

trees

Putting it all together

```
from vertexai.generative_models import GenerativeModel, GenerationConfig
```

```
model = GenerativeModel(“gemini-2.5-flash”)
```

```
config= GenerationConfig(temperature=0.9, top_p=0.7, top_k=20)
```

```
prompt = “Recommend me a programming activity to improve my skills."
response = model.generate_content(contents=prompt, generation_config=config)
```

Grounding LLMs

Out-of-the-box LLM aren’t connected to real world

```
from vertexai.generative_models import GenerativeModel, GenerationConfig
```

```
prompt = “How to concat dataframes in pandas”
response = model.generate_content(contents=prompt, generation_config=config)
```

Response would depend on the knowledge cut-off

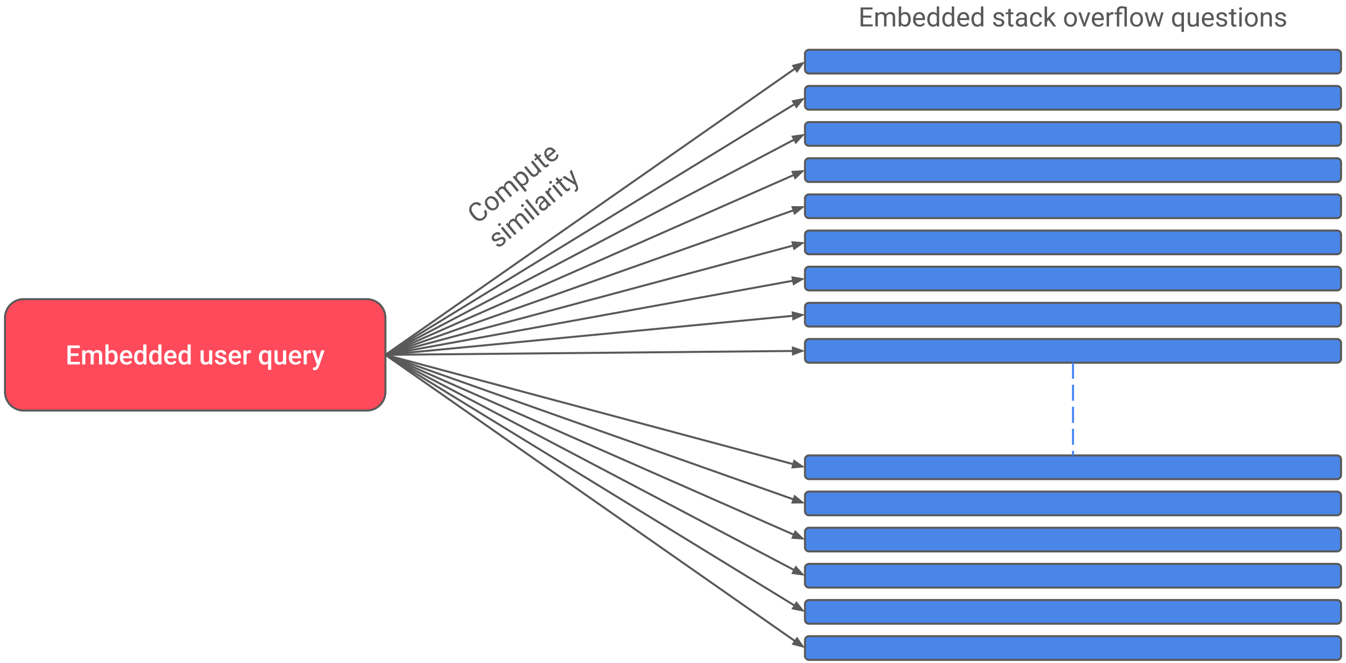
Grounding LLMs:

- **Access information outside of training data**
- **Integrate with existing IT systems, databases, and business data**
- **Mitigate risk of hallucinations**

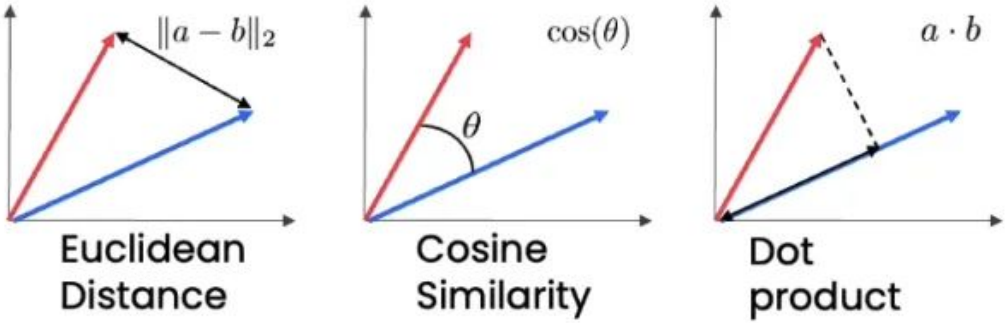
Q&A over Stack Overflow data

Question: Combine 3 dataframes in pandas. I have a dataframe like this

Accepted Answer: You can use pd.concat here. Just define and pass all 3 dataframes to . . .



Measuring Similarity using Embeddings

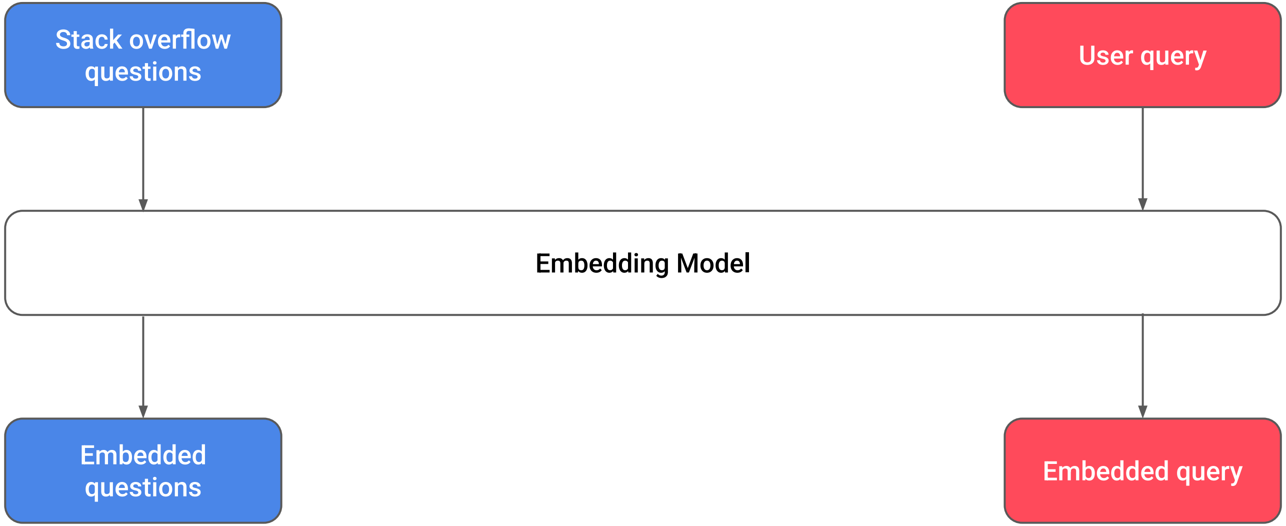


Euclidean Distance (L2 Distance): Distance between ends of vectors

Cosine Similarity: Cosine of the angle between vectors

Dot Product: Cosine multiplied by lengths of both vectors

Q&A with Semantic Search



Stack overflow questions

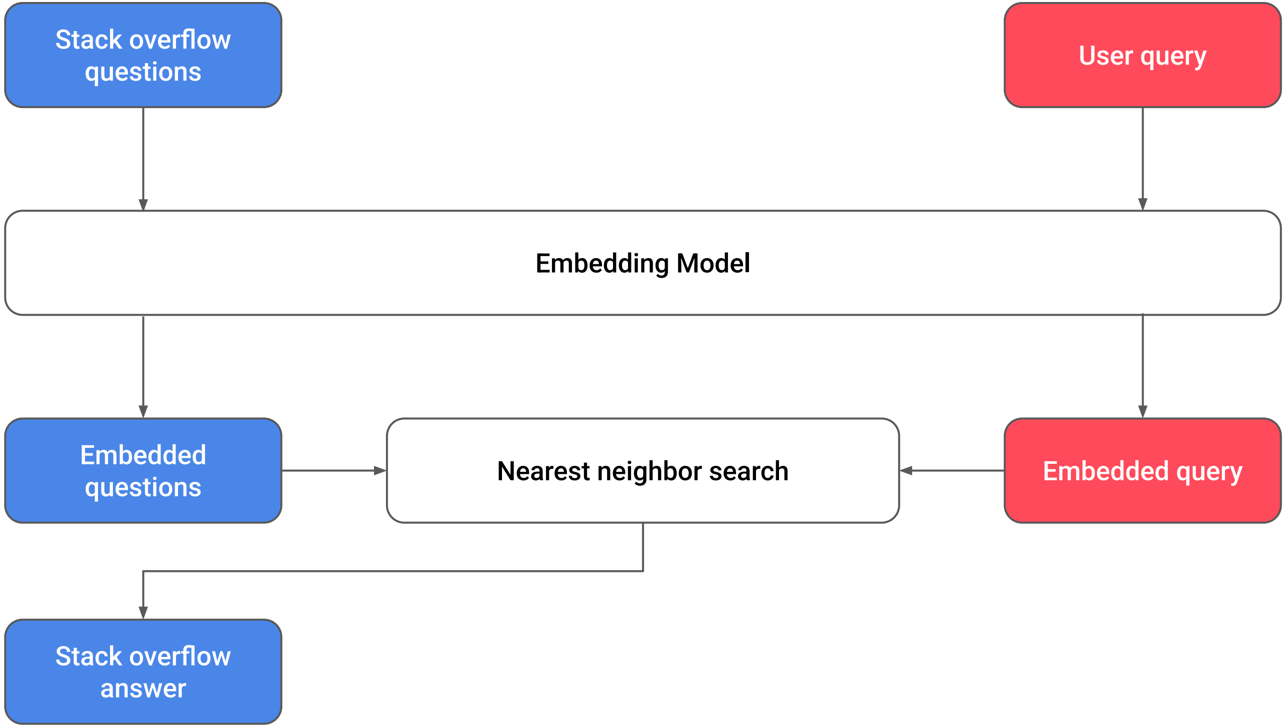
User query

Embedding Model

Embedded questions

Embedded query

Q&A with Semantic Search



Stack overflow questions

User query

Embedding Model

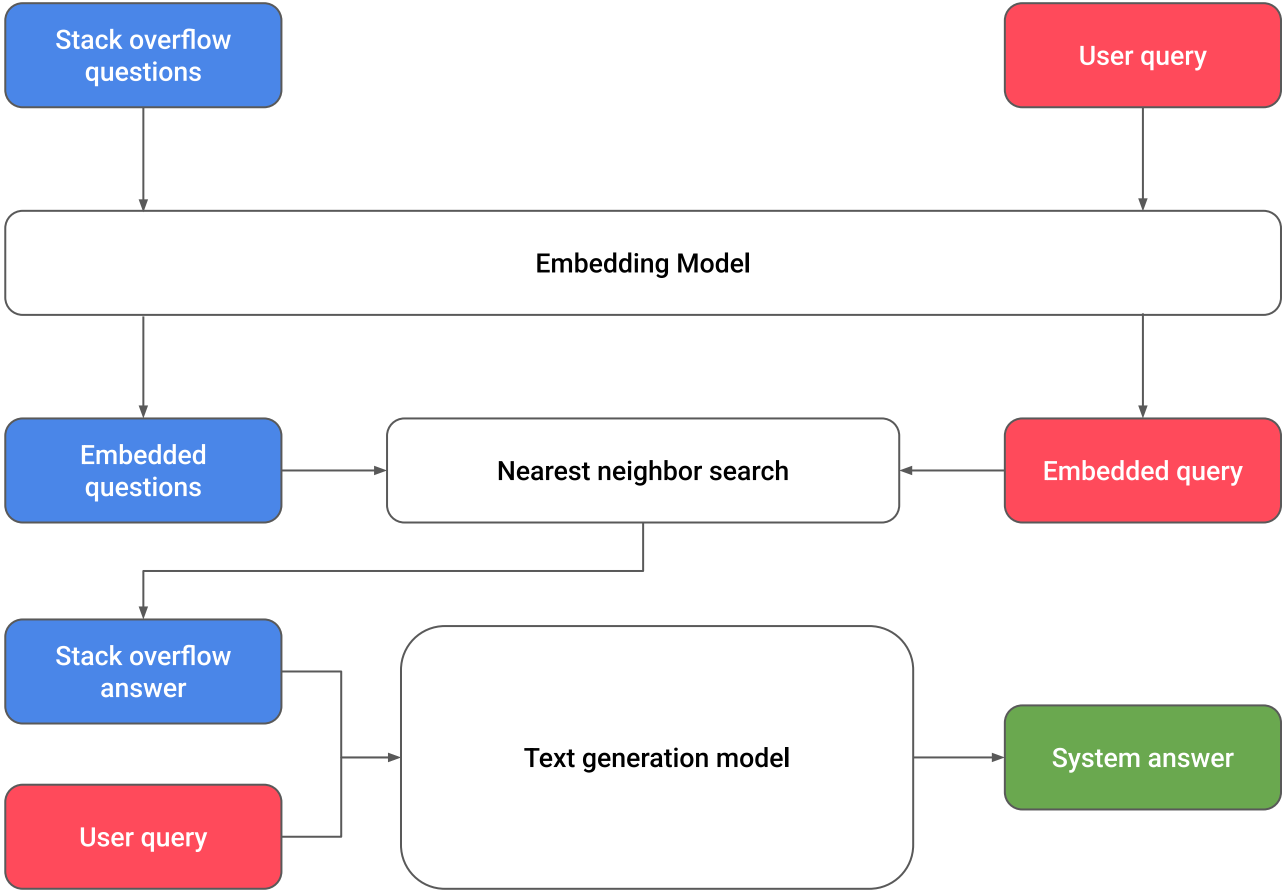
Embedded questions

Nearest neighbor search

Embedded query

Stack overflow answer

Q&A with Semantic Search



THANK YOU