

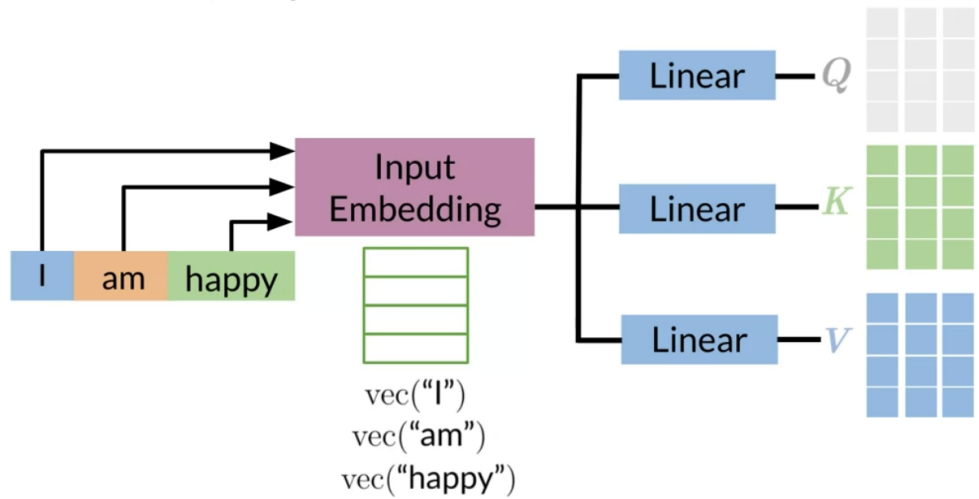
## Text Summarization

- ✓ **Video:** Transformers vs RNNs  
6 min
  - ✓ **Reading:** Transformers vs RNNs  
10 min
  - ✓ **Video:** Transformer Applications  
8 min
  - ✓ **Reading:** Transformer Applications  
10 min
  - ✓ **Video:** Dot-Product Attention  
7 min
  - ✓ **Reading:** Dot-Product Attention  
10 min
  - ▶ **Video:** Causal Attention  
4 min
  - 📖 **Reading:** Causal Attention  
10 min
  - ▶ **Video:** Multi-head Attention  
6 min
  - 📖 **Reading:** Multi-head Attention  
10 min
  - 📖 **Lab:** Attention  
1h
  - ▶ **Video:** Transformer Decoder  
5 min
  - 📖 **Reading:** Transformer Decoder  
10 min
  - ▶ **Video:** Transformer Summarizer  
4 min
  - 📖 **Reading:** Transformer Summarizer  
10 min
  - 📖 **Lab:** The Transformer Decoder  
1h
  - 📖 **Reading:** Content Resource  
10 min
- Assignment**
- 📄 **Programming Assignment:** Transformer Summarizer  
3h

## Dot-Product Attention

Dot product attention could be summarized as follows:

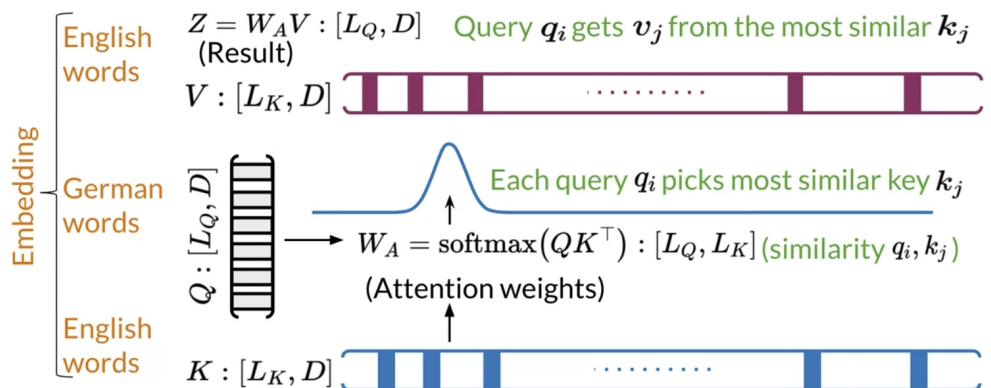
## Queries, Keys and Values



Given an input, you transform it into a new representation or a column vector. Depending on the task you are working on, you will end up getting queries, keys, and values. Each column corresponds to a word in the figure above. Hence, when you compute the following:

$$\begin{aligned}
 & \begin{matrix} \text{Matrix } Q & \times & \text{Matrix } K^T \\ \hline Q & & K^T \end{matrix} \\
 & \text{softmax}(QK^T) \\
 & \left( \begin{matrix} \text{Matrix } QK^T \\ \hline \end{matrix} \right) \times \begin{matrix} \text{Matrix } V \\ \hline V \end{matrix}
 \end{aligned}$$

This concept implies that similar vectors are likely to have a higher score when you dot them with one another. You transform that score into a probability by using a softmax function. You can then multiply the output by  $V$ .



You can think of the **keys** and the **values** as being the same. Note that both  $K, V$  are of dimension  $L_k, D$ . Each query  $q_i$  picks the most similar key  $k_j$ . Queries are the German words and the keys are the English words. Once you have the attention weights, you can just multiply it by  $V$  to get a weighted combination of the input.