Zurich University
of Applied Sciences

School of
Management and Law

EQUIS
since 2023

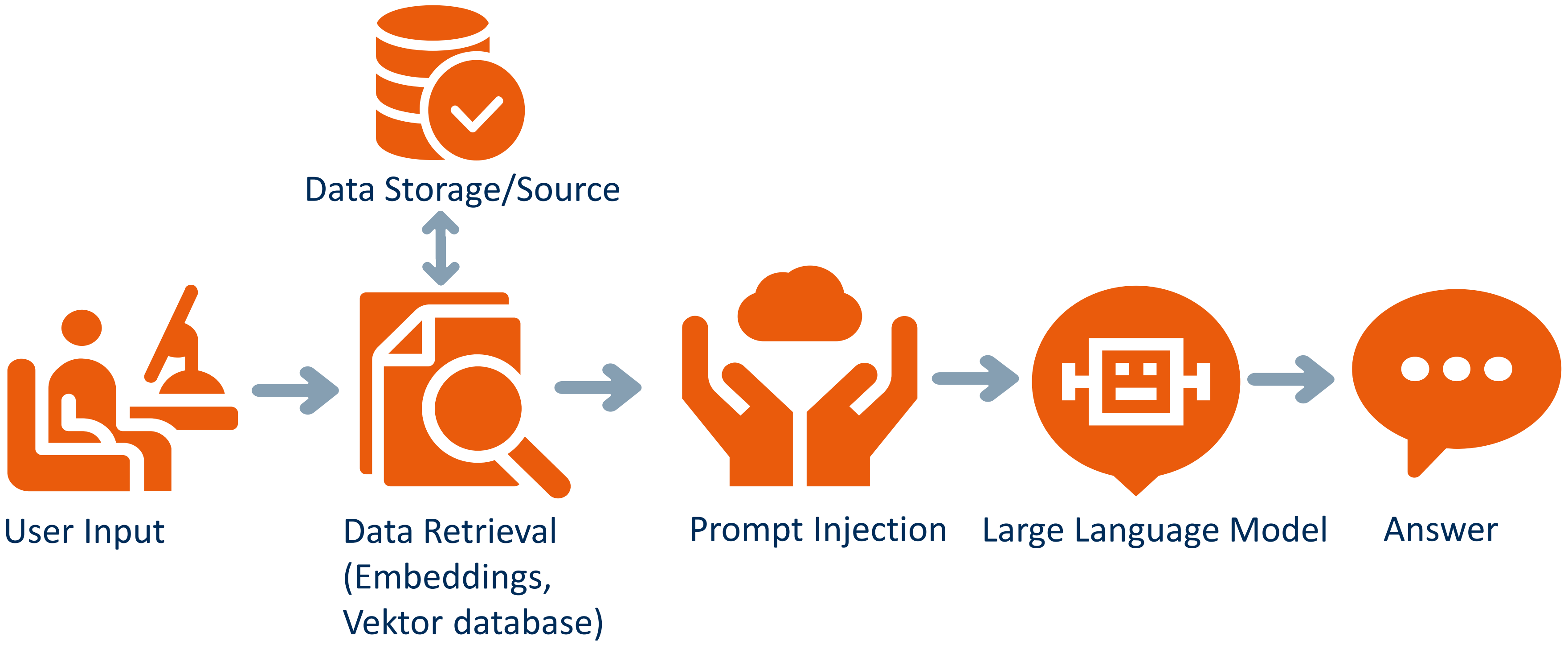
AACSB
since 2015

Retrieval Augmented Generation
Tokenization and Embedding

Building Competence. Crossing Borders.

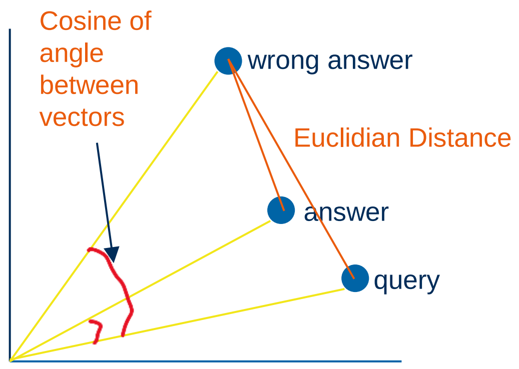
Jasmin Heierli
jasmin.heierli@zhaw.ch / 18.03.2024

RAG Architecture



New Problem

- Given our user query: *What are the first words spoken after landing on the moon?*
- Given a knowledge base (knowledge graph, database, collection of texts…)
- How can we identify relevant knowledge base entries?



Semantics

Time flies like an arrow; fruit flies like a banana

fly: moving through the air.
fly: a small insect with small wings

Semantics in a nut shell: A fundamental aspect of semantics is the recognition that there is not always a direct connection between words and their meanings. This concept challenges the intuitive belief that words are inherently tied to what they represent.

https://web.stanford.edu/~jurafsky/slp3/6.pdf

https://web.stanford.edu/~jurafsky/slp3/G.pdf

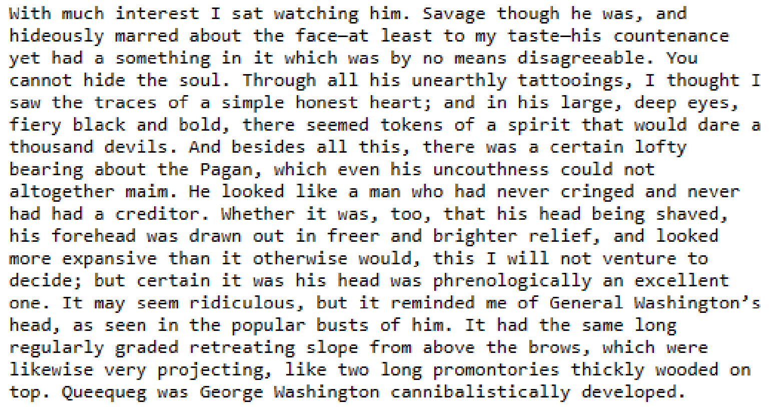
Word Representation

- Machine Learning Models cannot read raw text 😳
- LMs seem to be able to 😭

Why then do we store embeddings and not raw text in a vector store?

→ Indexing and search happens based on text similarity

→ Neither can be sufficiently computed based on characters

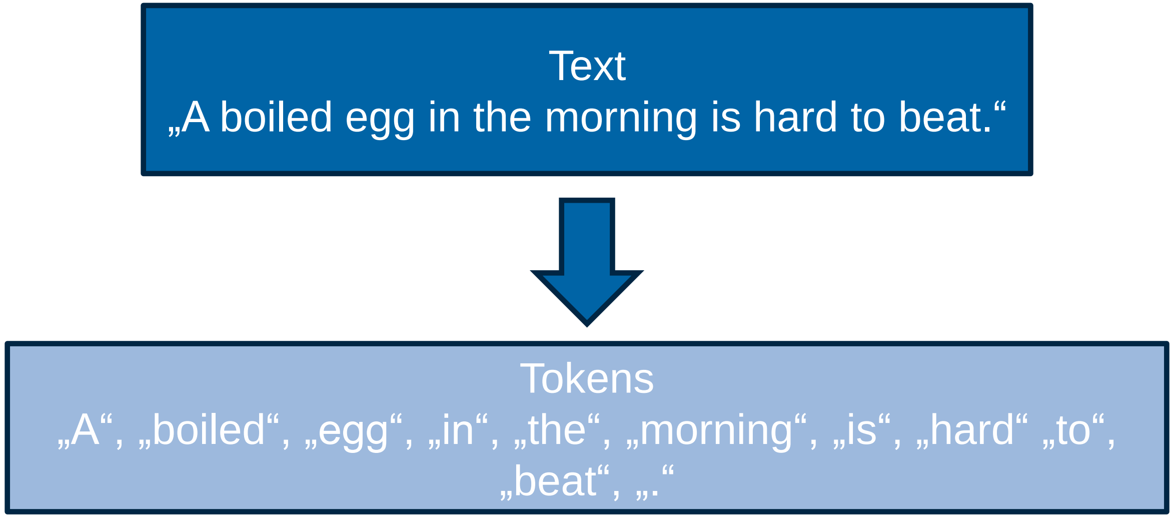


```
With much interest I sat watching him. Savage though he was, and
hideously marred about the face—at least to my taste—his countenance
yet had a something in it which was by no means disagreeable. You
cannot hide the soul. Through all his unearthly tattooings, I thought I
saw the traces of a simple honest heart; and in his large, deep eyes,
fiery black and bold, there seemed tokens of a spirit that would dare a
thousand devils. And besides all this, there was a certain lofty
bearing about the Pagan, which even his uncouthness could not
altogether maim. He looked like a man who had never cringed and never
had had a creditor. Whether it was, too, that his head being shaved,
his forehead was drawn out in freer and brighter relief, and looked
more expansive than it otherwise would, this I will not venture to
decide; but certain it was his head was phrenologically an excellent
one. It may seem ridiculous, but it reminded me of General Washington’s
head, as seen in the popular busts of him. It had the same long
regularly graded retreating slope from above the brows, which were
likewise very projecting, like two long promontories thickly wooded on
top. Queequeg was George Washington cannibalistically developed.
```

https://www.gutenberg.org/ebooks/2701

Tokenization

- Originally the task of segmenting a text into words
- Tokenization: The process of segmenting documents into paragraphs, sentences, and words (tokens) per sentence.



Text
„A boiled egg in the morning is hard to beat.“

Tokens
„A“, „boiled“, „egg“, „in“, „the“, „morning“, „is“, „hard“ „to“, „beat“, „.“

Space-based tokenization: simplest heuristic to split text for ***some*** languages

Source: Carstensen et al., *Computerlinguistik und Sprachtechnologie: Eine Einführung*, Spektrum Akademischer Verlag Heidelberg

Problem: New Words

Problem: You built your perfect linguistic tokenizer on a training corpus and your actual data that you work with, contains a word it has not seen before.

Example
Training corpus contains: low, high, cat, stairs
Test corpus contains: dog 🥴

Do you still think that linguistic word tokenizations is a good solution to build real world applications?

Solution: Subword tokenisation

Modern tokenizers induce ideal tokens that are often smaller than words, called subwords. They may correspond to morphemes[1], but they don‘t have to.

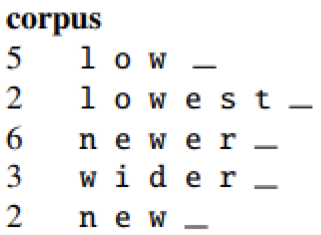
Byte-Pair Encoding: Bottom-up approach, we use our data to tell us what the ideal token should be 🤯

https://web.stanford.edu/~jurafsky/slp3/2.pdf

[1] smallest unit of meaning in a word

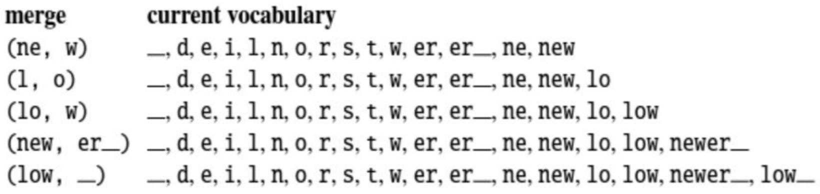
Byte-Pair Encoding

- Token-learner begins with a set of tokens that is just a set of all individual characters
- Then it starts to merge the two most frequent co-occurring characters together.
- It continues to count and merge together sequences, creating more and longer sequences
- The alogorithm usually respects word boundaries if existing
- Often ends up learning entire words from the training corpus.
- A token segmenter then greedily splits each test sentence into characters and then starts to apply the merging rules.



corpus

5	l o w _
2	l o w e s t _
6	n e w e r _
3	w i d e r _
2	n e w _



merge	current vocabulary
(ne, w)	_, d, e, i, l, n, o, r, s, t, w, er, er_, ne, new
(l, o)	_, d, e, i, l, n, o, r, s, t, w, er, er_, ne, new, lo
(lo, w)	_, d, e, i, l, n, o, r, s, t, w, er, er_, ne, new, lo, low
(new, er_)	_, d, e, i, l, n, o, r, s, t, w, er, er_, ne, new, lo, low, newer_
(low, _)	_, d, e, i, l, n, o, r, s, t, w, er, er_, ne, new, lo, low, newer_, low_

https://web.stanford.edu/~jurafsky/slp3/2.pdf

From Tokens to Numbers

A traditional way to encode texts as numbers: one-hot-encoding

Example: for web search you want to match “London cab” in documents containing “London taxi”

Result:
cab = [0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0]
taxi = [0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0]

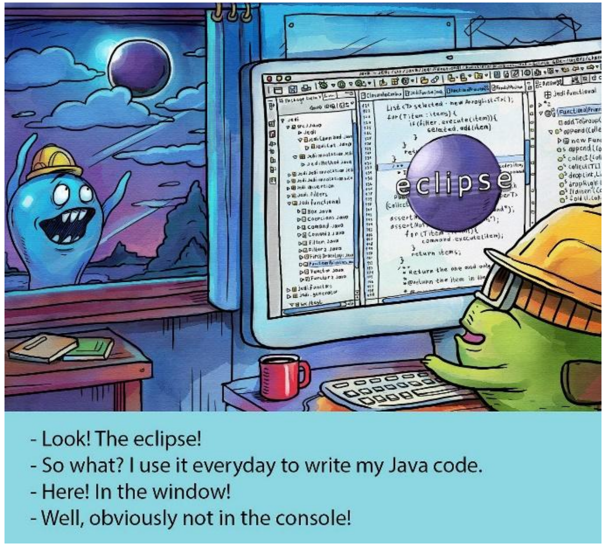
Problem
- The vectors are orthogonal
- There is no natural notion of similarity for one-hot-encoded vectors
- They become immensely sparse with a large vocabulary
- The integer-encoding does not capture any relationship between words

From Tokens to Numbers with Context

(6.1) Ongchoi is delicious sauteed with garlic.
(6.2) Ongchoi is superb over rice.
(6.3) ...ongchoi leaves with salty sauces...

A word's meaning can be inferred by the context it appears in. Synonyms (words with similar meanings) often appear in similar contexts.

We can use all the available contexts of a given word to build up
A representation of it.


eclipse
- Look! The eclipse!
- So what? I use it everyday to write my Java code.
- Here! In the window!
- Well, obviously not in the console!

https://twitter.com/data_monsters

https://web.stanford.edu/~jurafsky/slp3/6.pdf

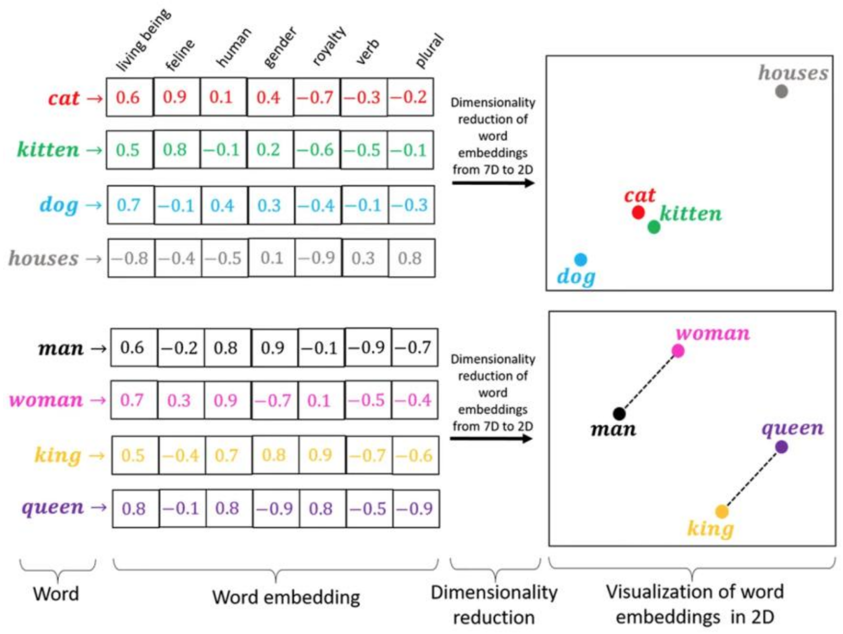
Word Embeddings

An embedding is a short dense vector of a fixed size consisting of floating point numbers. Similar words have a similar encoding.

Static embeddings: the same token has always the same embedding

Contextual embeddings: different representations for a token depending on its context.

The floating point values of the embedding vector are trainable parameters and the weights can be learned by a neural network. You just need to specify the amount of dimensions (typically between 100 and 2000)



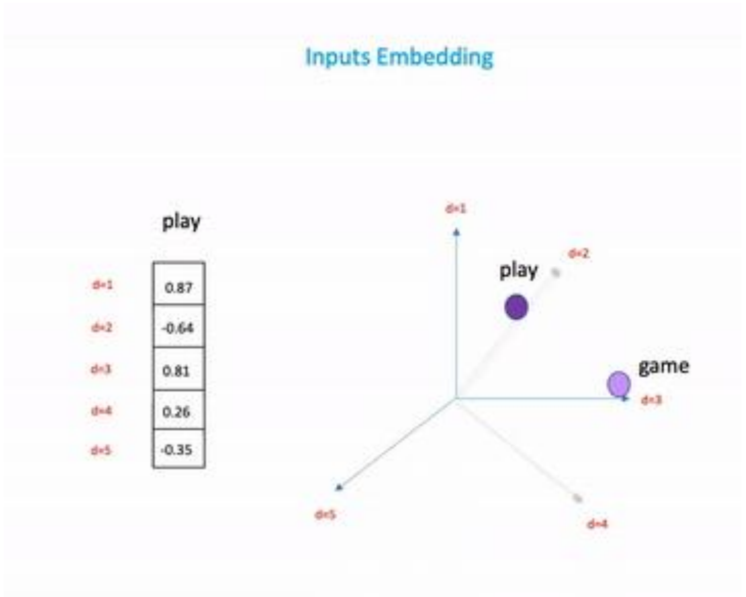
	living being	feline	human	gender	royalty	verb	plural
cat →	0.6	0.9	0.1	0.4	−0.7	−0.3	−0.2
kitten →	0.5	0.8	−0.1	0.2	−0.6	−0.5	−0.1
dog →	0.7	−0.1	0.4	0.3	−0.4	−0.1	−0.3
houses →	−0.8	−0.4	−0.5	0.1	−0.9	0.3	0.8

man →	0.6	−0.2	0.8	0.9	−0.1	−0.9	−0.7
woman →	0.7	0.3	0.9	−0.7	0.1	−0.5	−0.4
king →	0.5	−0.4	0.7	0.8	0.9	−0.7	−0.6
queen →	0.8	−0.1	0.8	−0.9	0.8	−0.5	−0.9

Dimensionality reduction of word embeddings from 7D to 2D
houses
cat
kitten
dog
Dimensionality reduction of word embeddings from 7D to 2D
woman
man
queen
king
Word
Word embedding
Dimensionality reduction
Visualization of word embeddings in 2D

https://web.stanford.edu/~jurafsky/slp3/6.pdf

Word Embeddings



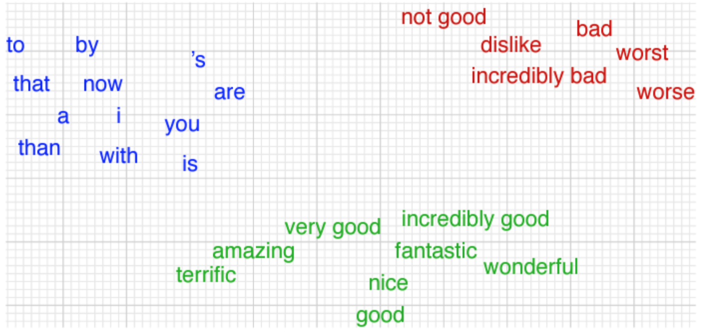
https://medium.com/womenintechnology/transformers-for-dummies-word-embeddings-1c9ade5ff500

- Beginning: Each word is assigned a vector with random numbers
- During training vectors are modified in such a way that they better represent each word in relation to all other words.
- Example: Play and games start out with random embeddings and converge during training
- Unsupervised!

Word Embeddings

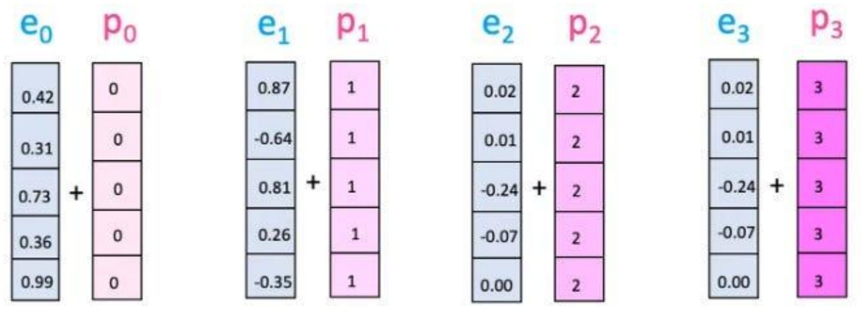
- Dense vector for each word, calculated so that words that appear in similar contexts have similar vectors.
- Word embeddings are a distributed representation
- Since word embeddings are vectors, you can calculate: distance, sum, difference, product

→ Cosine similarity is most commonly used to calculate the similarity between two vectors



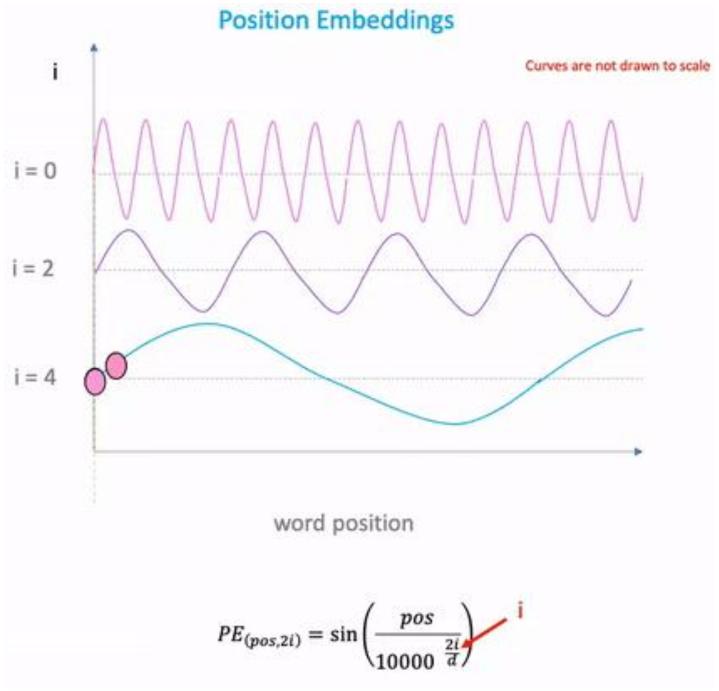
https://web.stanford.edu/~jurafsky/slp3

Word Embeddings



- We need to add position in order to capture nuances in meaning
- Naive way: Add sentence position to vector
  → No: can significantly distort emebeddings, especially, for words at the end of long sentences/chunks
- Fractions? A word at the position 2 has a value of 2/30 in a 30 word sentence and 2/50 in a 50 word sentence…

https://medium.com/womenintechnology/transformers-for-dummies-positional-embeddings-bec8474ed6f5



$$PE_{(pos,2i)} = \sin\left(\frac{pos}{10000^{\frac{2i}{d}}}\right)$$

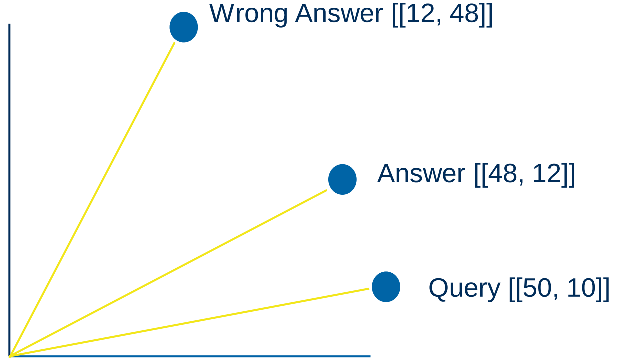
- Use of Sinus and Cosinus functions!
- They are smooth and continuous, meaning small position changes lead to small changes in the values.
- The values are independent of sentence length because the frequency of the waves can be adjusted.
- Each wave corresponds to a different index of the embedding vector
- The **x-axis represents the word position** in the sentence, and each **wave corresponds to a different index i of the positional embedding**.

https://medium.com/womenintechnology/transformers-for-dummies-positional-embeddings-bec8474ed6f5

Cosine Similarity

Let‘s close the circle and go back to our original Problem:

We can now calculate the cosine similarity between two vectors



$$cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\Sigma_{i=1}^{n} A_i \cdot B_i}{\sqrt{\Sigma_{i=1}^{n} A_i^2} \cdot \sqrt{\Sigma_{i=1}^{n} B_i^2}}$$

$$\mathbf{A} \cdot \mathbf{B} = 50 \times 48 + 10 \times 12 = 2400 + 120 = 2520$$

$$\|\mathbf{B}\| = \sqrt{48^2 + 12^2} = \sqrt{2304 + 144} = \sqrt{2448} \approx 49.5$$

$$\|\mathbf{A}\| = \sqrt{50^2 + 10^2} = \sqrt{2500 + 100} = \sqrt{2600} \approx 51.0$$

$$cosine\ similarity = \frac{2520}{51.0 \times 49.5} \approx 0.998$$

$$cos(2.73°) \approx 0.998$$

Visualisation of a Word Embedding Space

https://projector.tensorflow.org/

Thank you.