- Statistical based approaches treated words as atomic symbols:

  Animal Dog Trick

- Or in vector space:

  [0 0 0 0 1 0 0 0 0 . . . 0]

- Also known as 'one hot' representation.

1	2	3	4
1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1

Sparse representation capture each word; but very expensive.

BoW → term frequency
TF-IDF
] always sparse
can't understand context

Animal: [0 0 0 1 0 0 0 0 0 ..] = V_1

Dog: [0 0 0 0 0 0 1 0 0 0] = V_2

Truck: [0 0 0 0 1 0 0 0 0 0 ...] = V_3

* Distributional Representation — You shall know a word by company it keeps: John Rupert.

diff meaning in diff context.

→ One of the most successful ideas of modern NLP.

→ Linguistic units with similar distributions have similar meaning.

→ A bank is a financial institution that accepts deposits from the public & creates credit.

→ The northern bank of the river is flooded.

* Co-occurrence matrix

→ A co-occurrence matrix is a (terms × terms) matrix which captures the number of times a term appears in the context of another term.

→ The context is defined as a window of k words around the terms

Text Corpus (like n-gram) capturing in form of a matrix

1. "ChatGPT ~~is a~~ language model."
2. "Language models like ChatGPT ~~are~~ powerful"
3. "~~The~~ language model ChatGPT ~~is~~ based ~~on~~ GPT-3.5 arch."

window size = 1 (Take 1 NN) Before & After.

can capture context (too many 0's)

* **Distributed Representations**

→ Compact, dense, low-dimensional and real-valued representations

→ Also, known as word embeddings each single component of the vector representation does not have any meaning of its own.

→ The interpretable features are hidden & distributed among uninterpretable vector components

(Each word as a dense representⁿ)

$$\text{animals} = \begin{bmatrix} 0.286 \\ 0.792 \\ -0.112 \\ -0.193 \\ 0.341 \\ 0.512 \end{bmatrix}$$

each not interpretable

are they good repr of desired word

man ≡ woman (word embeddings)

what exactly some colors mean.

(closeness of two words)

* **Word Embeddings** (feed with neural network)

Bert, Glove

↳ They are numerical representations of words in a continuous vector space, where each word is mapped to a dense vector of real numbers.

↳ These vectors capture the semantic & contextual meaning of words based on their usage in a large corpus of a text.

↳ Transformer based: GPT

color coding of vectors help us interpret the similarity b/w 2 words.

* **Analogies**

→ You can add & subtract word embeddings and see the concept of analogies.

king − man + woman ≅ queen.

model.most_similar(positive= ["king", "woman"], negative= ("man"))

T-SNE visualization

word embeddings

Gave all travel documents to a model

50 dim $\xrightarrow[\text{reduction}]{\text{dim}}$ 2 dim

most words related to

each word has 50 features

city travel feeling relative food

info captured for a word

* **vector Embedding of Words**

1. latent Semantic Analysis/ Indexing (1988)
   - Term weighing based model.
   - Consider occurrences of terms at document level.
2. Word2Vec (2013)
   - Prediction-based model.
   - Consider occurrences of terms at context level.

* ## Prediction Based Models - word2vec

→ word2vec is a neural network-based model created by Google researchers in 2013 to learn word embeddings (ie; vector representations of words)

→ These embeddings capture semantic relationships b/w words, enabling similar words to have similar vector representations in a high-dimensional space.

→ In 2013, proposed word2vec can be trained using two new models.

* ## CBOWM

→ CBOWM focusses on guessing a word based on its context.

→ Predicting n^{th} word given the context words.

Ex: The quick brown fox jumped over the lazy dog.

The ↓ 100...
quick ↓ 010...
brown ↓ 0010...

(network is learning that wherever there is quick fox it should be followed by fox).

classmate

Date ______

Page ______

→ The input (first) layer is represented by a one-hot encoded vector and consists of the context words surrounding the target word.

→ Training data: All n-word windows in the corpus.

→ The hidden (~~sec~~ second) layer is where the word embeddings are learned and the size depends on the dimensionality.