Lesson 5: Generating and Encoding Text with Transformers

5.1 The Natural Language Processing Pipeline

5.2 Generative Models of Language

5.3 Generating Text with Transformers Pipelines

5.4 Deconstructing Transformers Pipelines

5.5 Decoding Strategies

5.6 Transformers are Just Latent Variable Models for Sequences

5.7 Visualizing and Understanding Attention

5.8 Turning Words into Vectors

5.9 The Vector Space Model

5.10 Embedding Sequences with Transformers

5.11 Computing the Similarity Between Embeddings

5.12 Semantic Search with Embeddings

5.13 Contrastive Embeddings with Sentence Transformers

5.1

The Natural Language Processing Pipeline

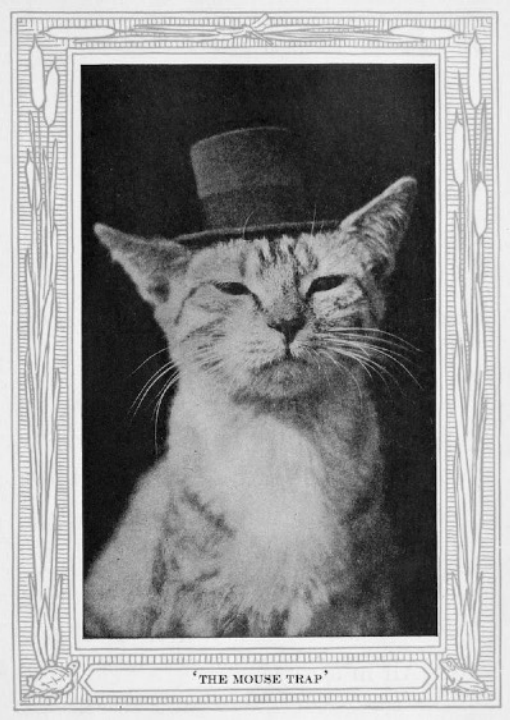
Natural Language Processing (NLP)

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

→

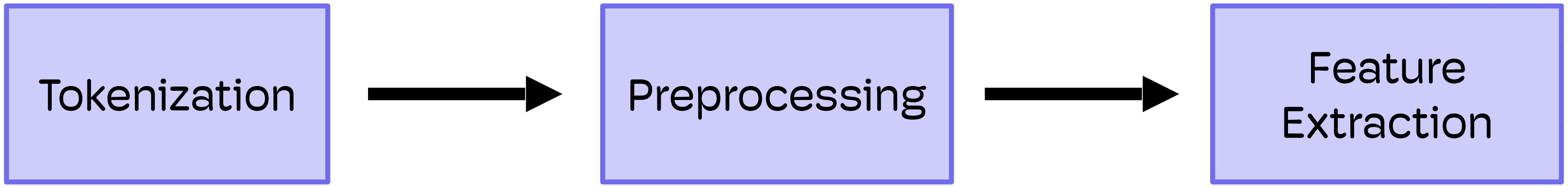
[1, 3, 1, 1, 2, 0, 1, 0]
[0, 1, 4, 0, 0, 1, 1, 1]
[3, 0, 1, 1, 2, 2, 3, 2]
[0, 1, 1, 1, 0, 3, 2, 3]
[1, 2, 1, 2, 2, 0, 0, 0]
[1, 0, 1, 1, 0, 1, 1, 1]
[0, 2, 0, 0, 2, 2, 0, 0]
[1, 1, 1, 1, 0, 1, 1, 1]

Image Vectorization



245	238	222	255
233	0	17	254
255	6	3	223
250	9	11	242
251	247	245	232

NLP Pipeline



NLP Pipeline

Tokenization → Preprocessing → Feature Extraction

- Sentence level
- Word level
- Character level
- Byte pair encoding

- Punctuation filtering
- Stop word removal
- Stemming
- Lemmatization

- Bag of words
- TF-IDF
- POS tagging
- word2vec

Terminology

- **Document:** Single row of data/corpus
- **Corpus:** Entire set of all documents
- **Vocabulary:** Set of all words in corpus
- **Vector:** Mathematical representation of document (counts of word occurrences)

Bag of Words

The brown fox → { "the" : 1, "brown": 1, "fox" : 1 } → [1,0,1,0,1,0,…]

brown   fox

original document → **Tokenization** → dictionary of word counts → **Vectorization** → feature vector

Bag of Words (Bernoulli)

	red	brown	jumps	the	fox	panda
doc0	0	1	0	1	1	0
doc1	1	0	0	1	1	0
doc2	1	0	0	1	0	1
doc3	0	0	1	1	1	0

Bag of Words (Multinomial)

	red	brown	jumps	the	fox	panda
doc0	0	2	0	4	2	0
doc1	1	0	0	1	1	0
doc2	2	0	0	2	0	2
doc3	0	0	1	2	2	0

Dedicated NLP Libraries

- spaCy: https://spacy.io
- fastText: https://fasttext.cc
- Gensim: https://radimrehurek.com/gensim/
- AllenNLP: https://allenai.org/allennlp
- flair: https://github.com/flairNLP/flair
- fairseq: https://github.com/facebookresearch/fairseq

5.2

Generative Models of Language

Probabilistic Model of Natural Language

Language Models

$$P(w_1, \ldots, w_m) = \prod_{t=1}^{m} P(w_t \mid w_{1:t-1})$$

Pearson

Language Models

History of words
that came before

$$P(w_t \mid w_{1:t-1})$$

Probability of
next word

Pearson

Language Models

Often not "full" history
(infinite context)

$$P(w_t \mid w_{t-5:t-1})$$

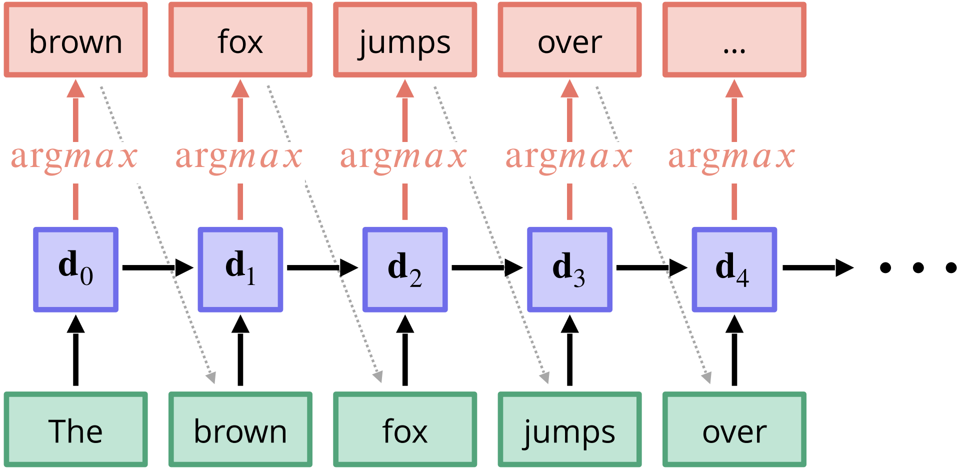
Probability of
next word

Pearson

Language Models

- **n-gram model:** Fixed window of previous *n* words
- **Neural/RNN:** Learned embeddings of *n* words
- **Large language model (LLM):** Large scale self- and semi-supervised pretraining of bidirectional models

Causal Language Modeling



5.3

Generating Text with Transformers Pipelines

Live Coding

5.4

Deconstructing Transformers Pipelines

Live Coding

5.5

Decoding Strategies

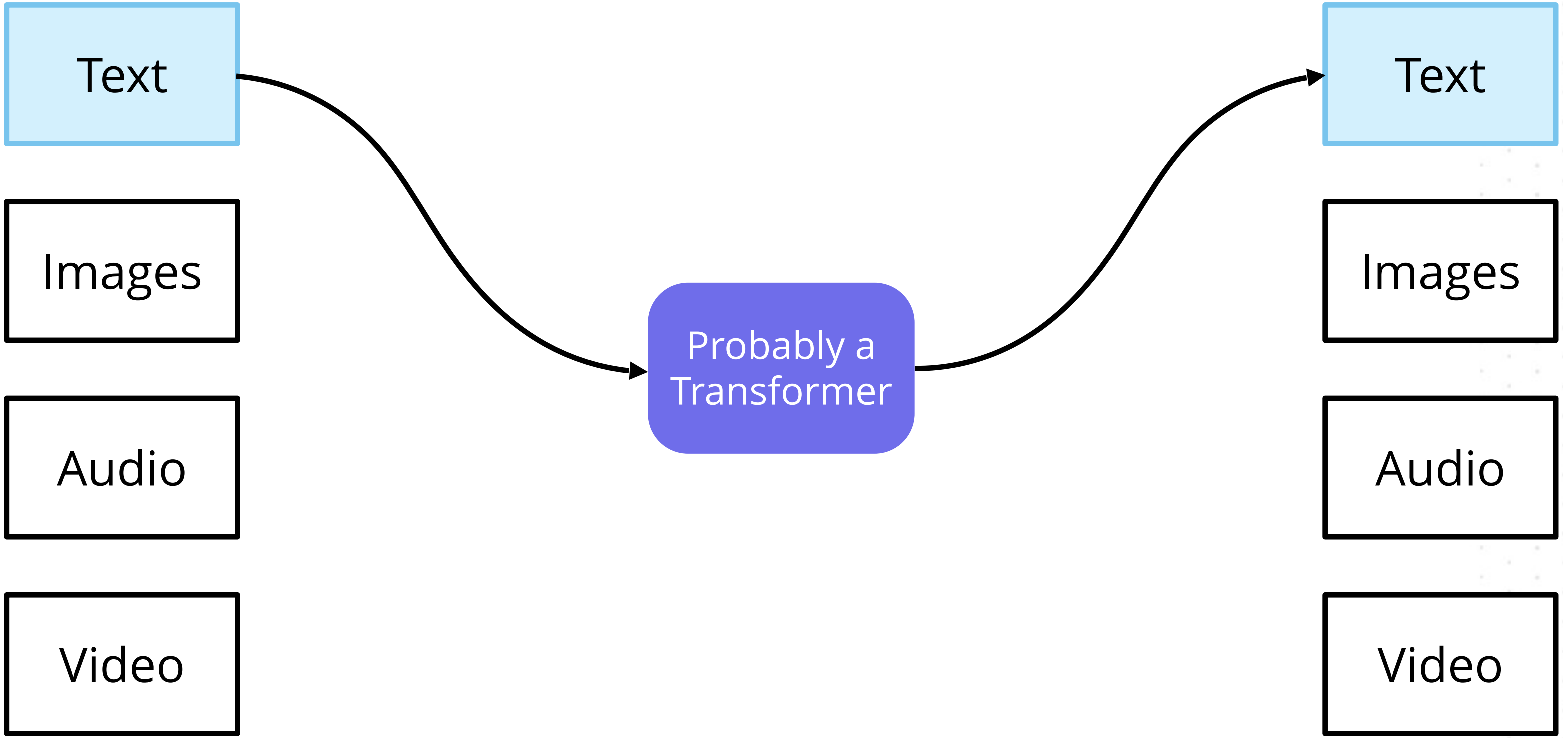
Live Coding

5.6

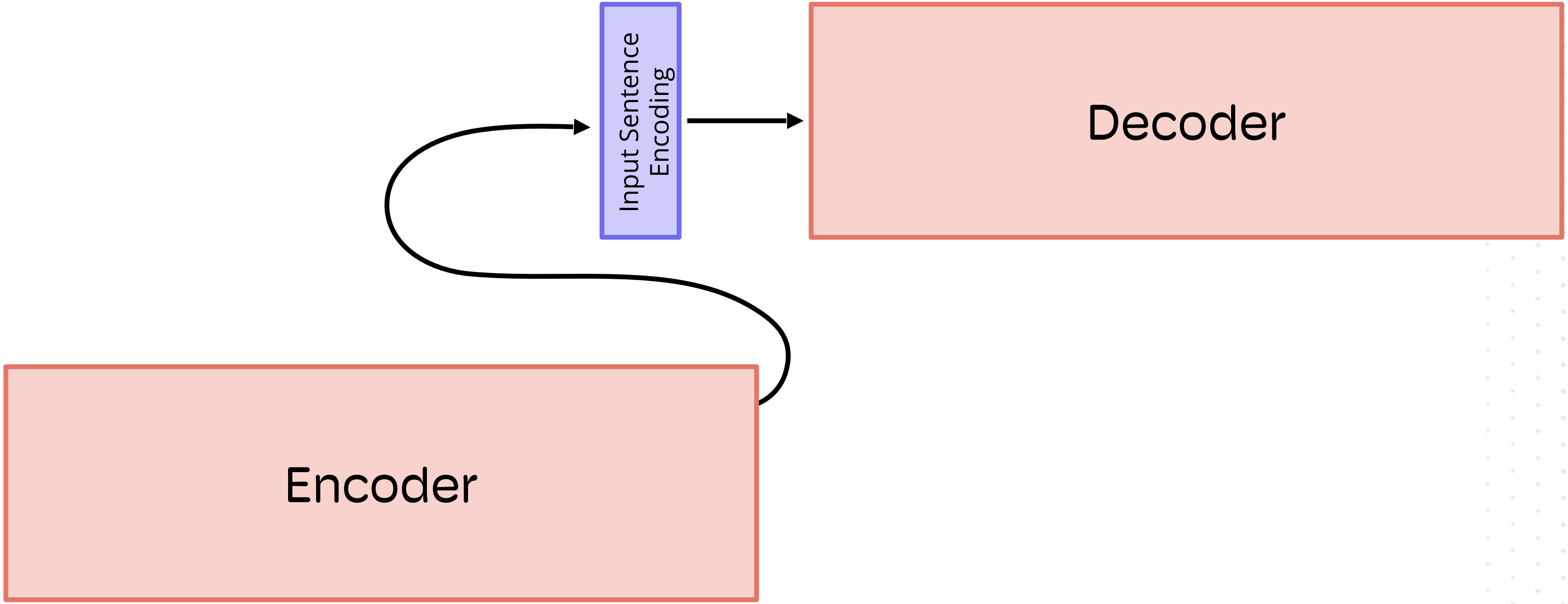
Transformers are Just Latent Variable Models for Sequences

Pearson

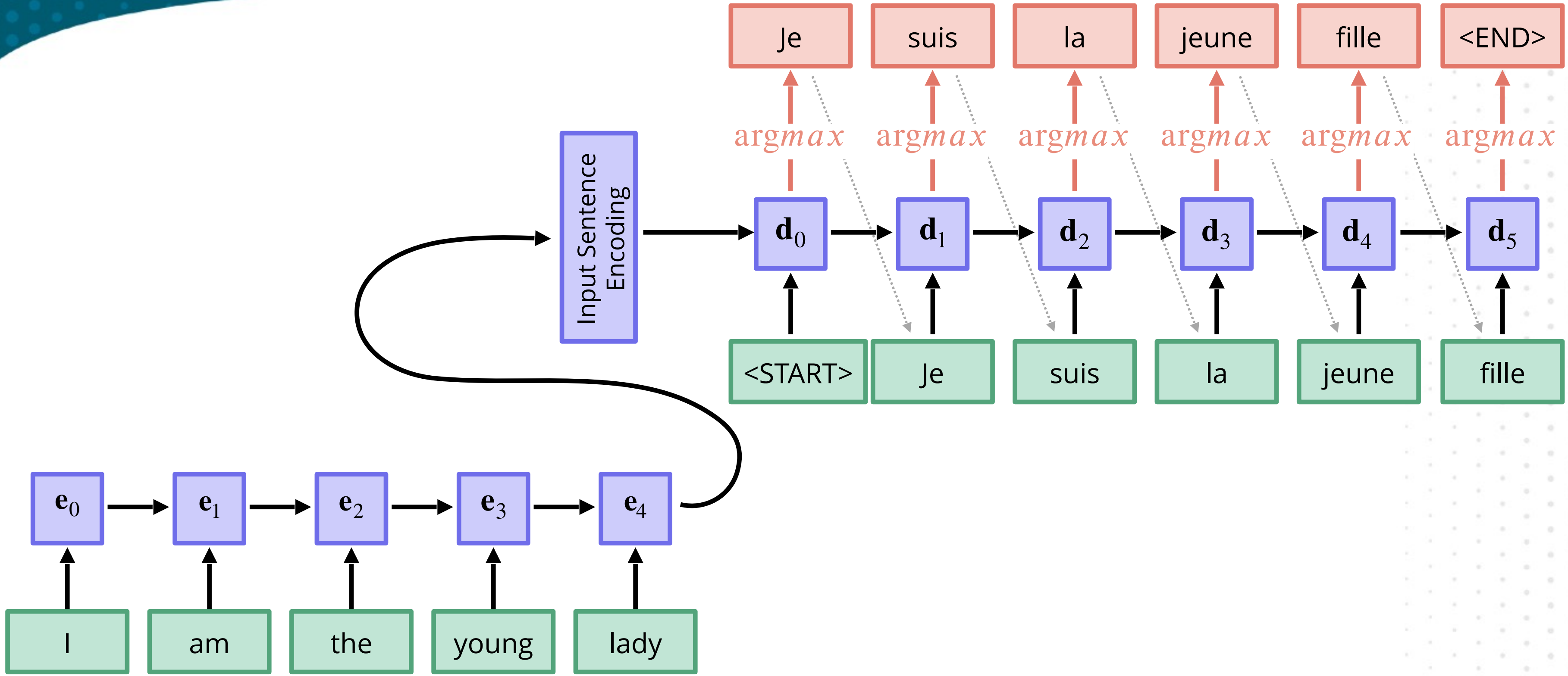
Text-to-Text (Translation, Summarization, Other)



seq2seq

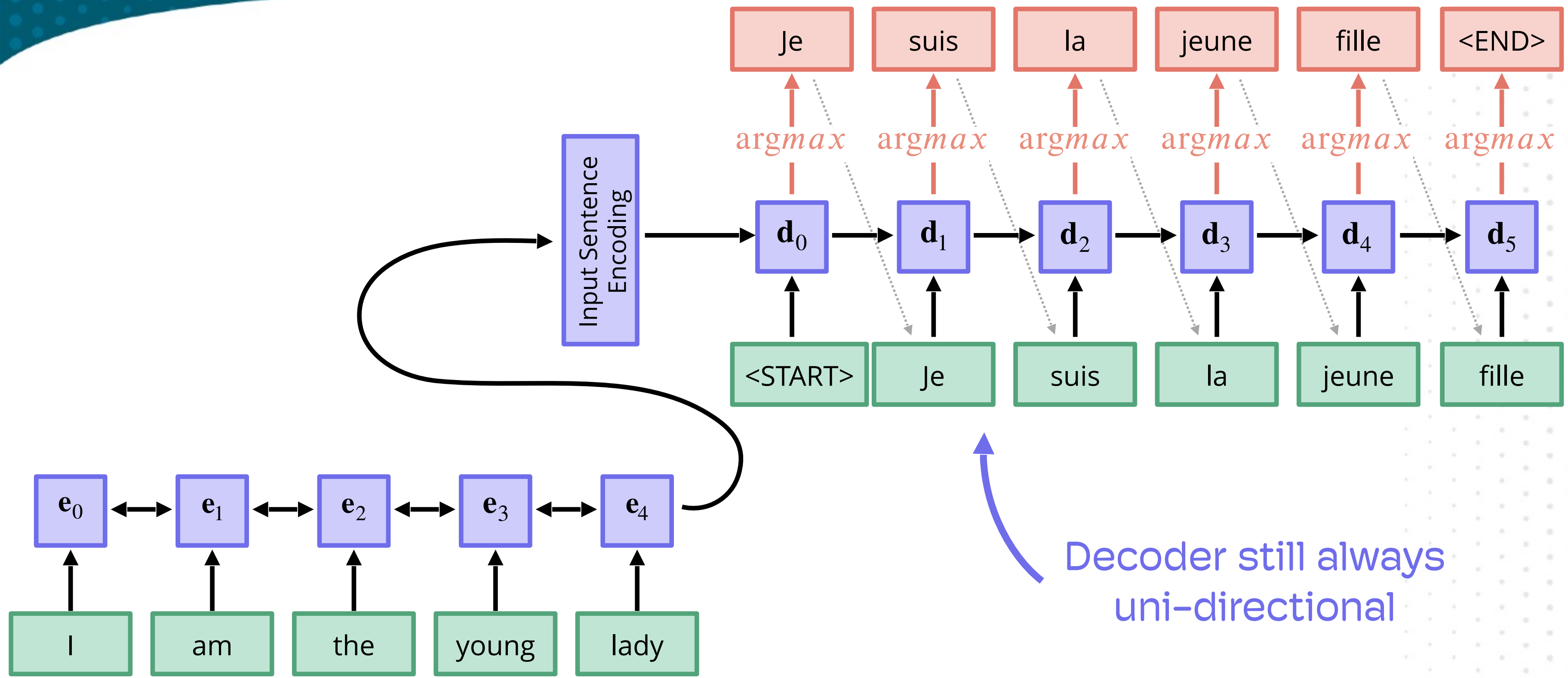


seq2seq

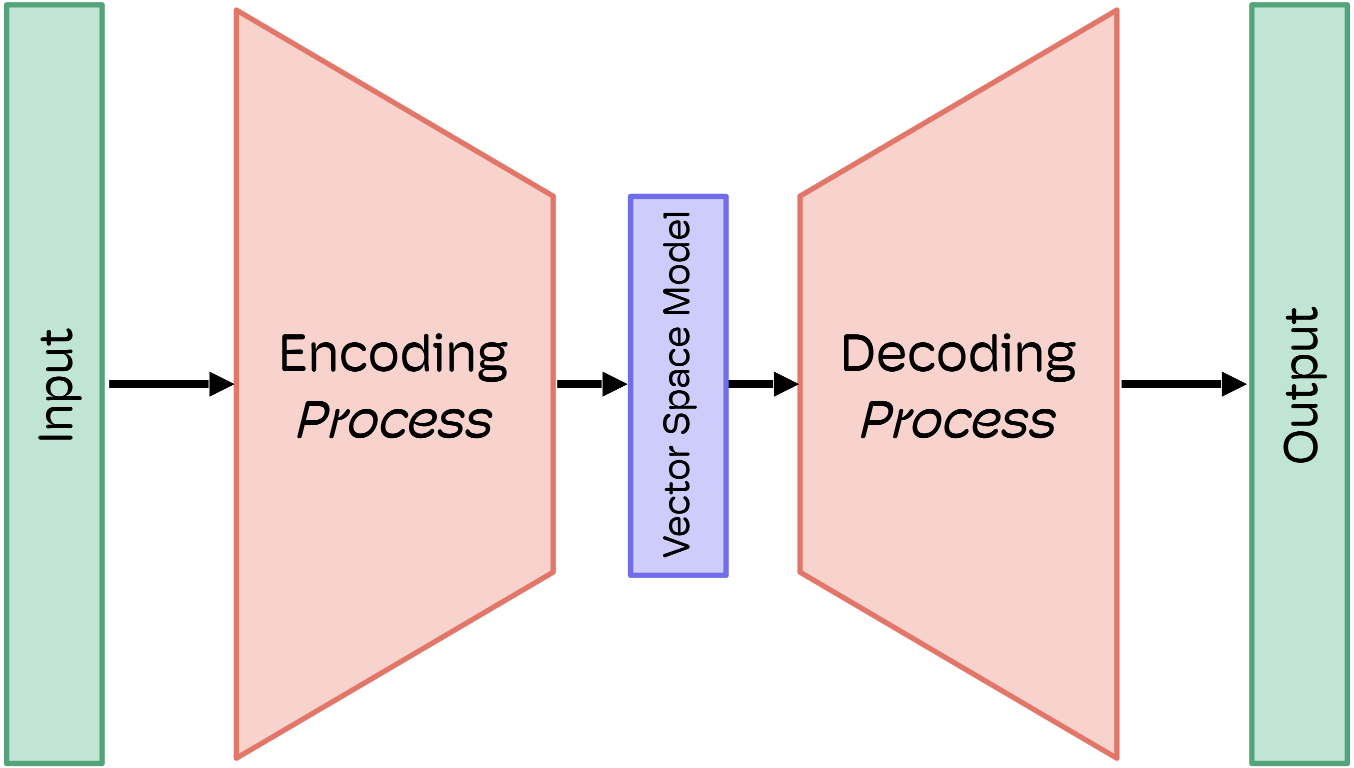


Je
suis
la
jeune
fille
<END>
argmax
argmax
argmax
argmax
argmax
argmax
Input Sentence Encoding
$\mathbf{d}_0$
$\mathbf{d}_1$
$\mathbf{d}_2$
$\mathbf{d}_3$
$\mathbf{d}_4$
$\mathbf{d}_5$
<START>
Je
suis
la
jeune
fille
$\mathbf{e}_0$
$\mathbf{e}_1$
$\mathbf{e}_2$
$\mathbf{e}_3$
$\mathbf{e}_4$
I
am
the
young
lady

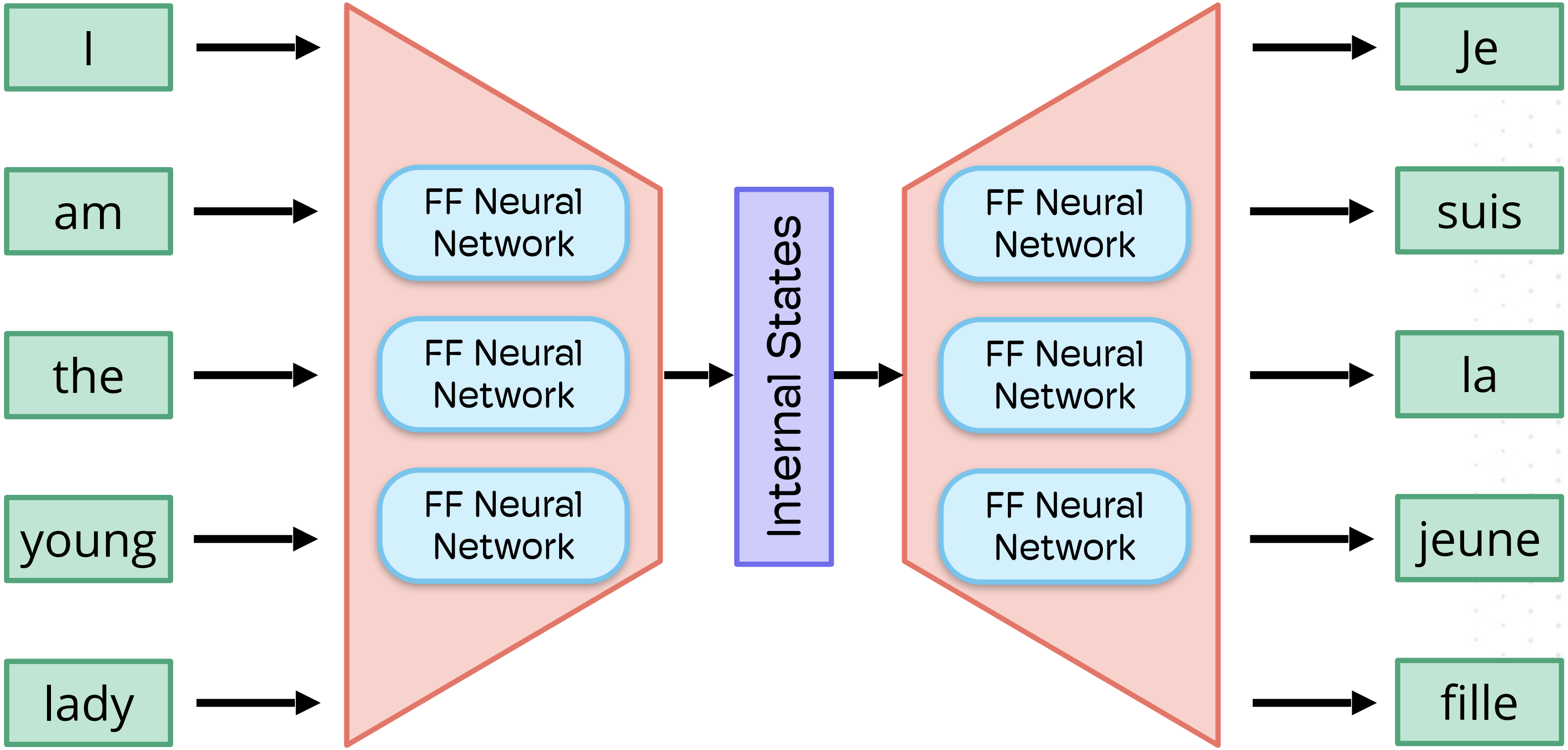
Bidirectional seq2seq



Encoding Natural Language

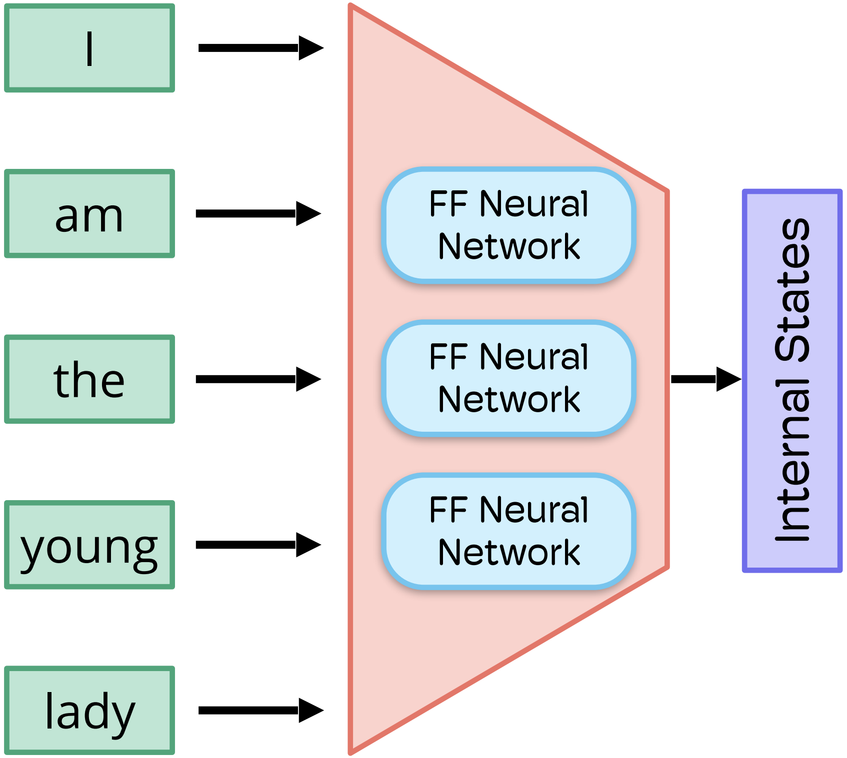


Transformer



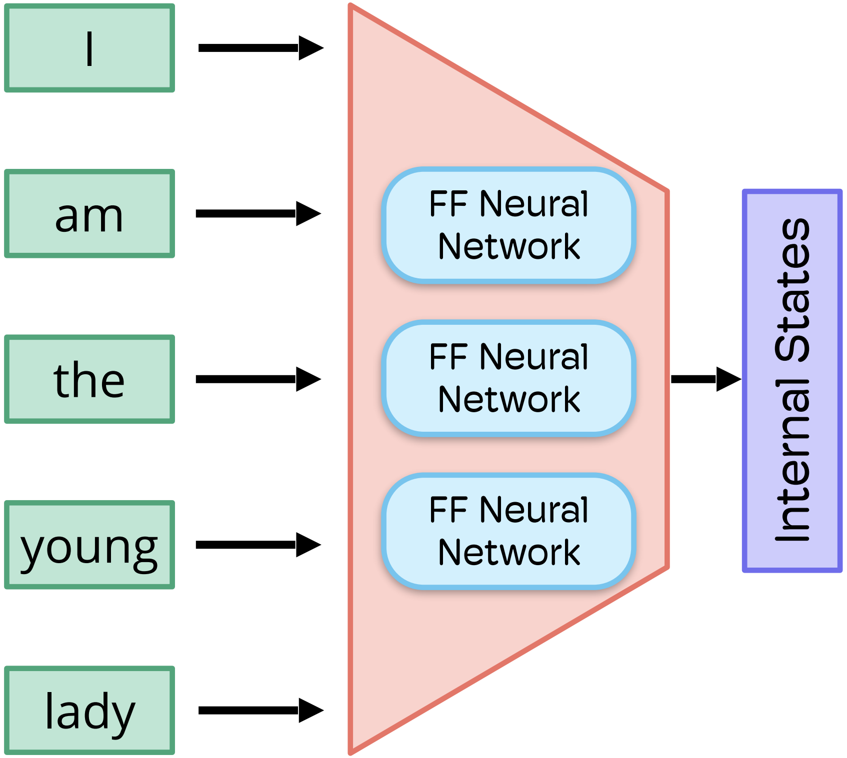
Pearson

Encoder Only Transformer



Can “see” the entire input

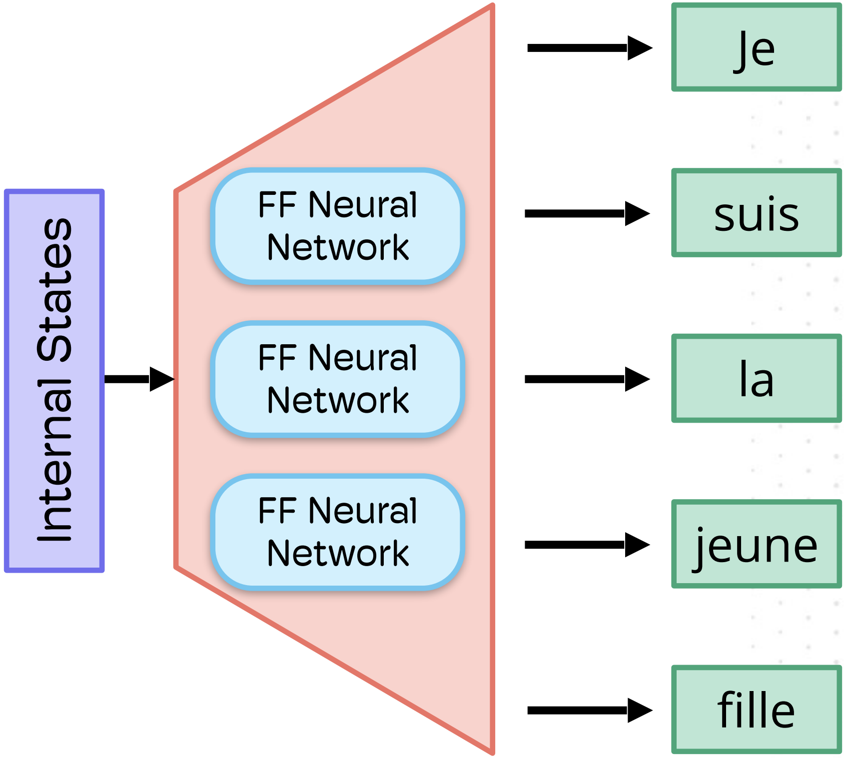
Encoder Only Transformer



Typically for downstream tasks (i.e. BERT)

Decoder Only Transformer

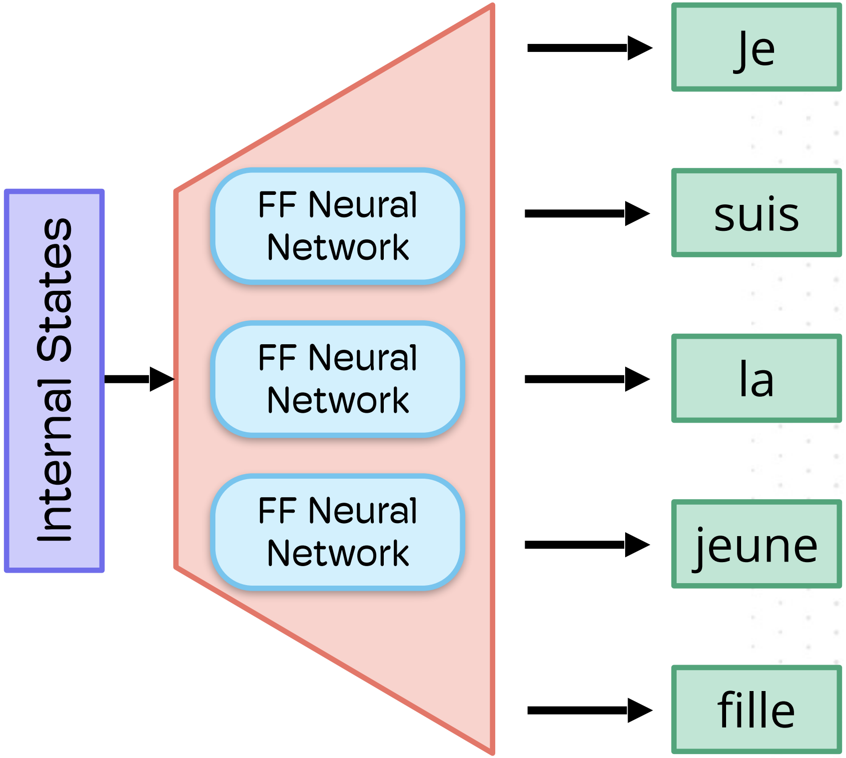
Only has access to the previous words



Pearson

Decoder Only Transformer

Typically for generation (i.e. GPT)



5.7

Visualizing and Understanding Attention

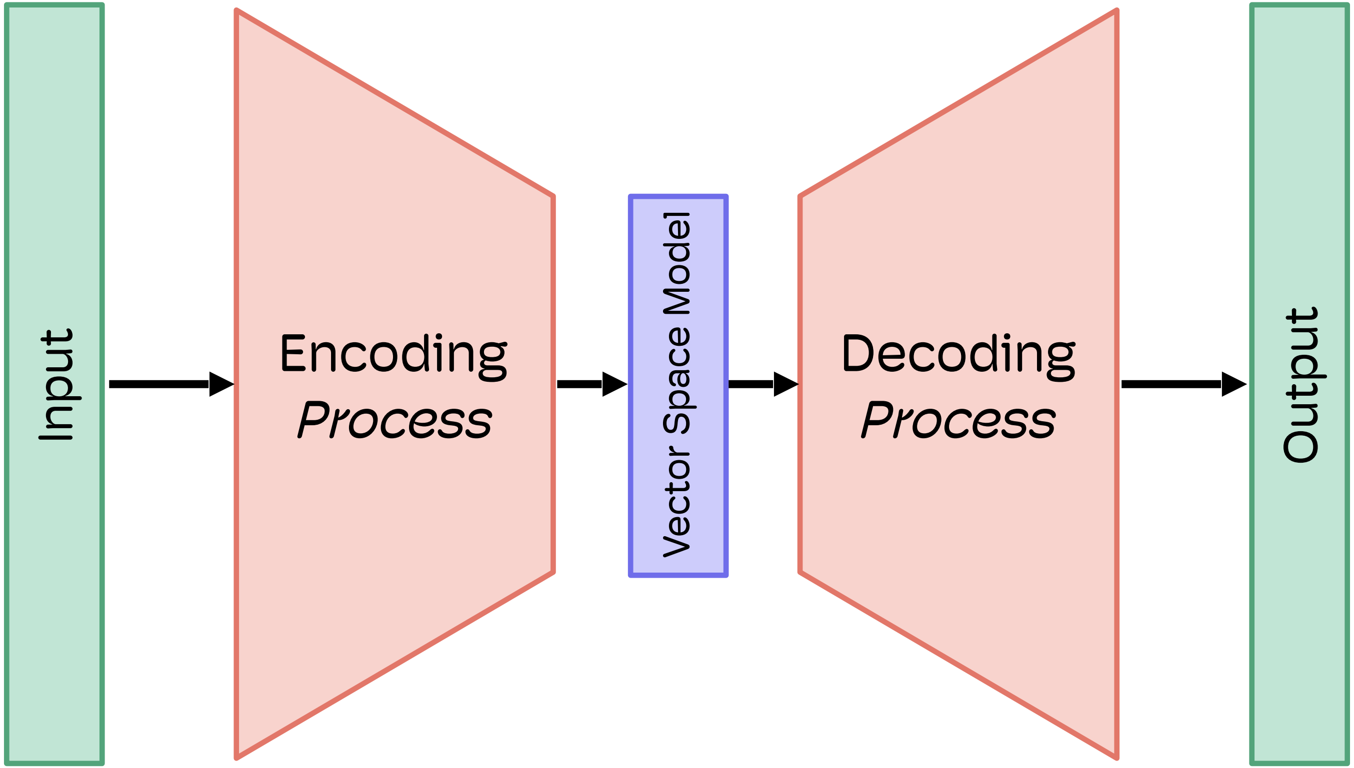
Pearson

Live Coding

5.8

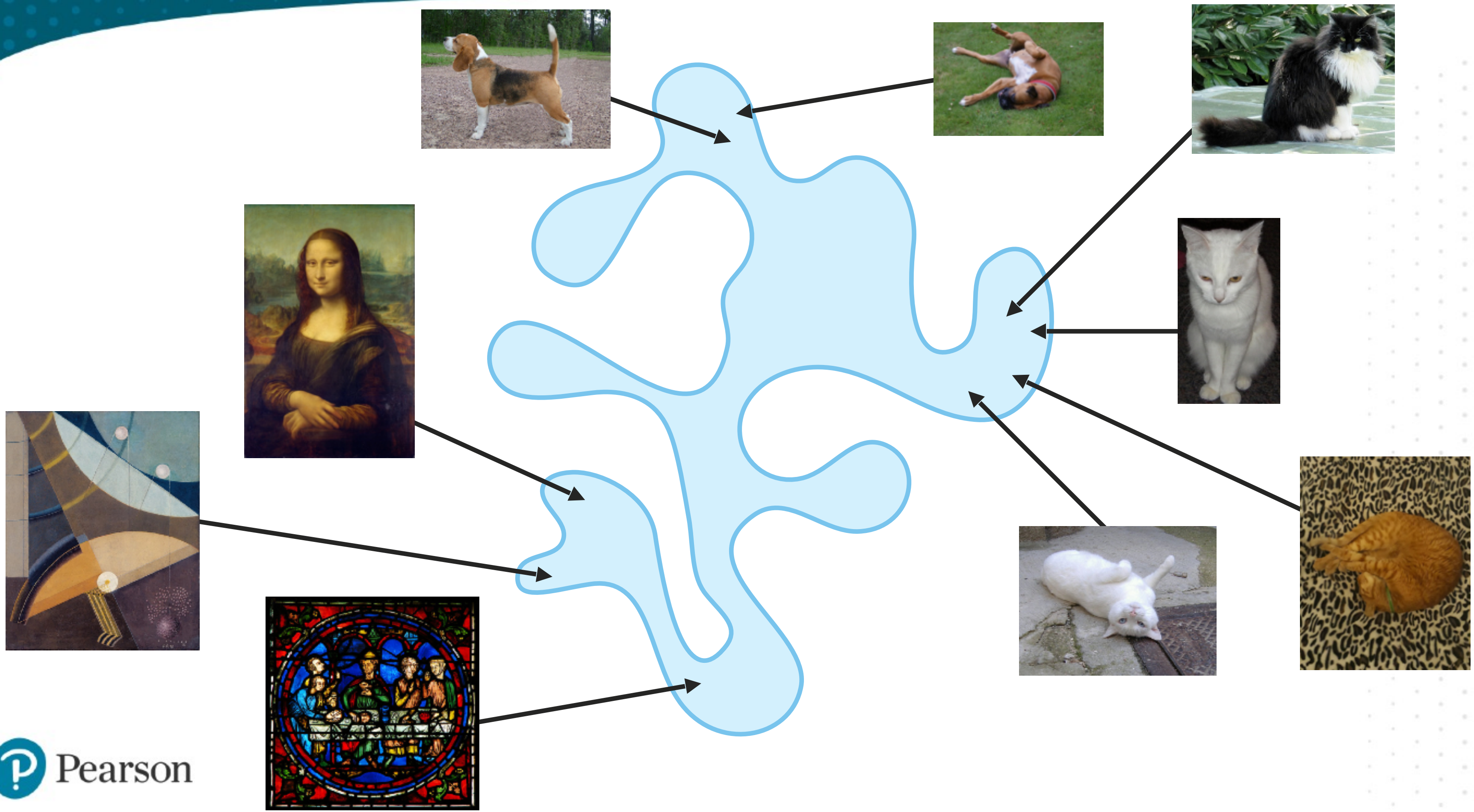
Turning Words into Vectors

Encoding Natural Language



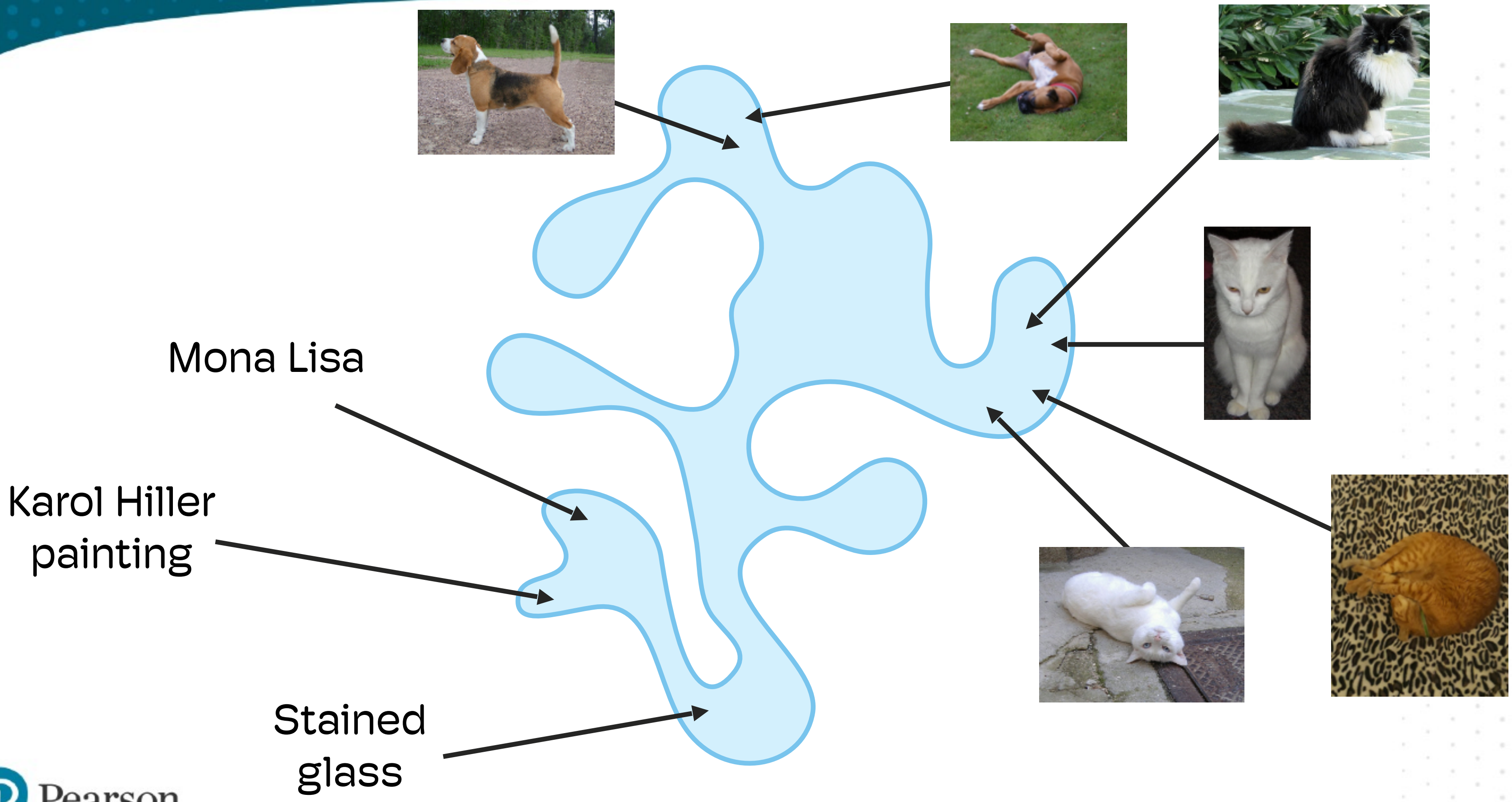
Pearson

Natural Image Manifold

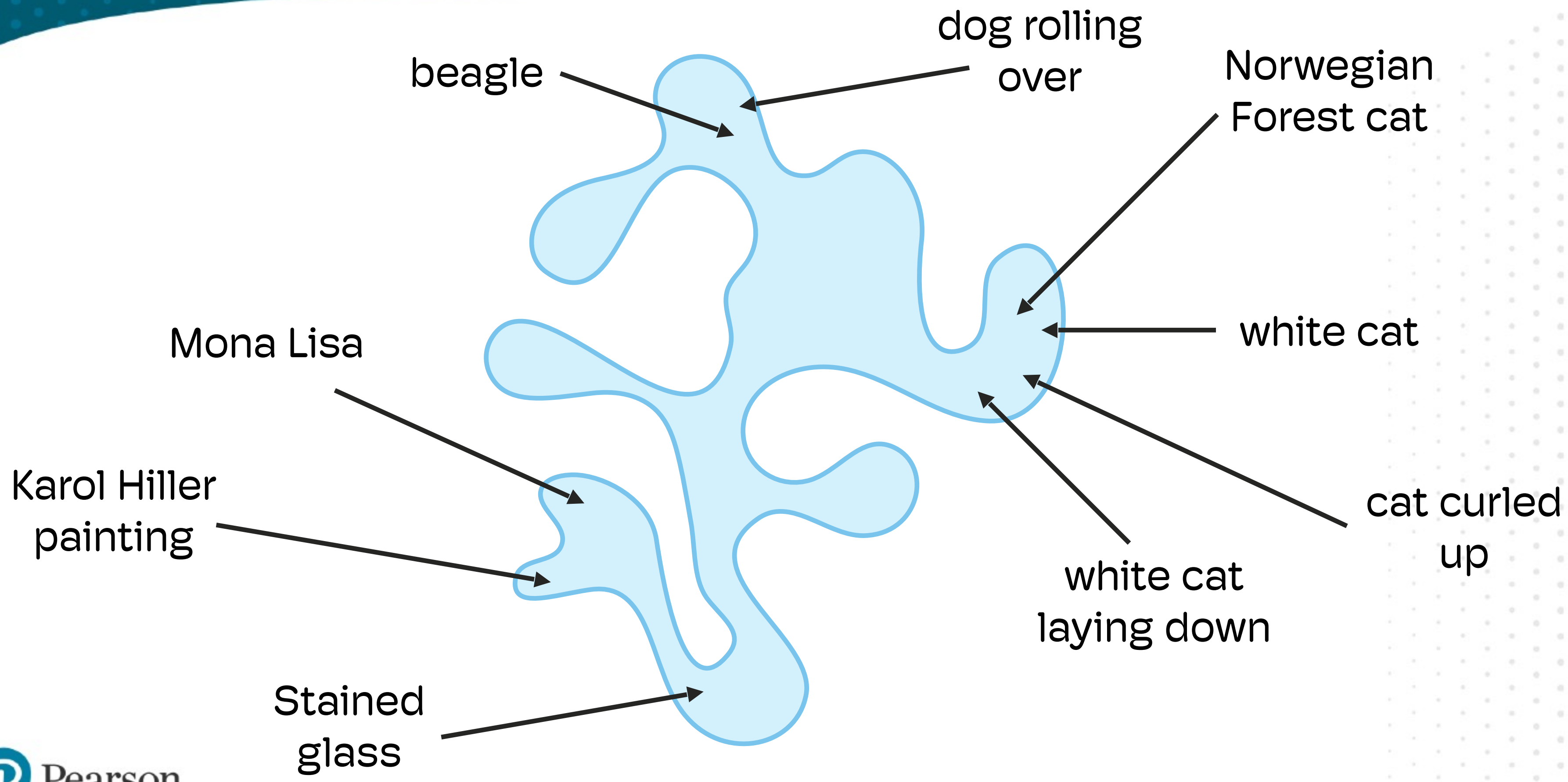


Pearson

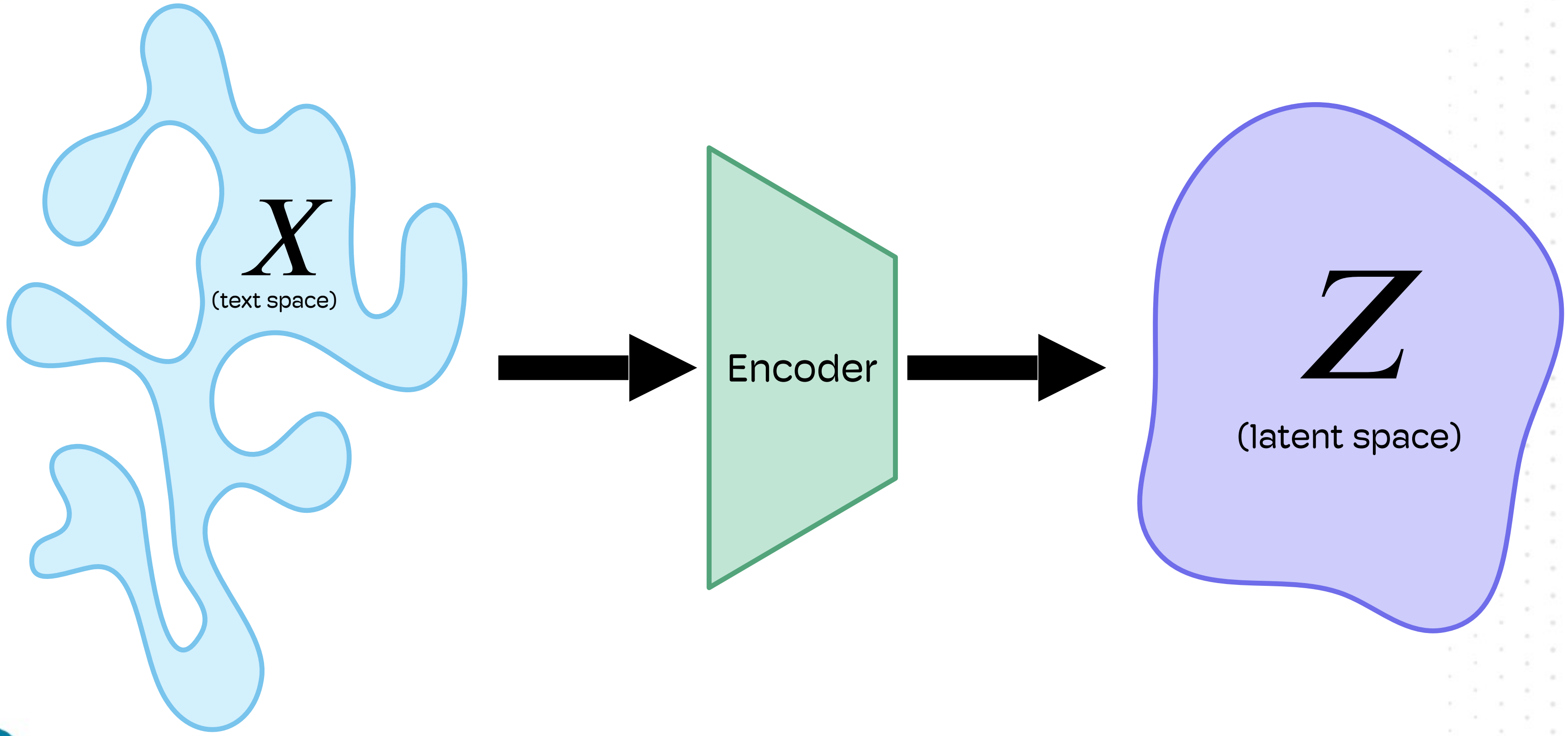
Natural Language Manifold



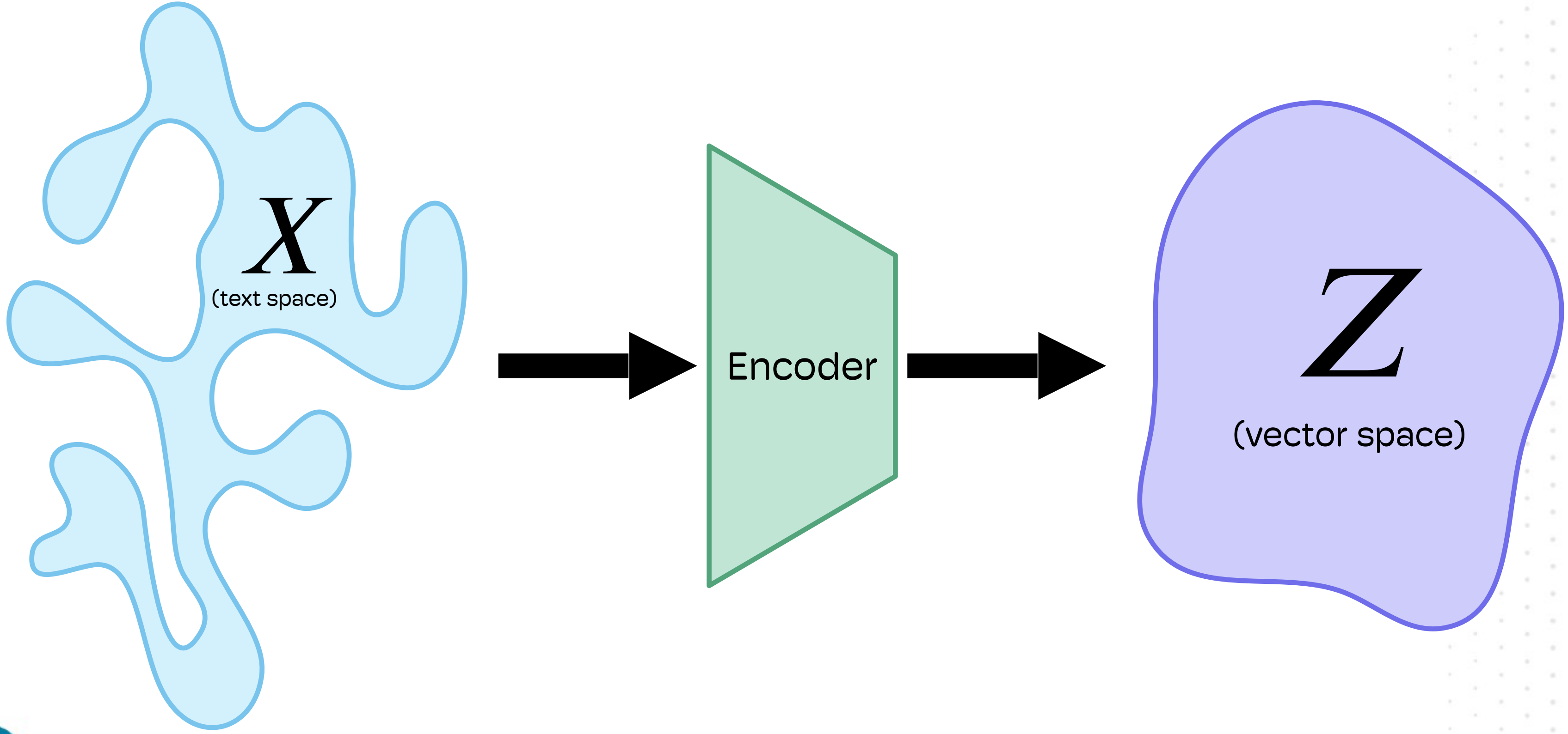
Natural Language Manifold



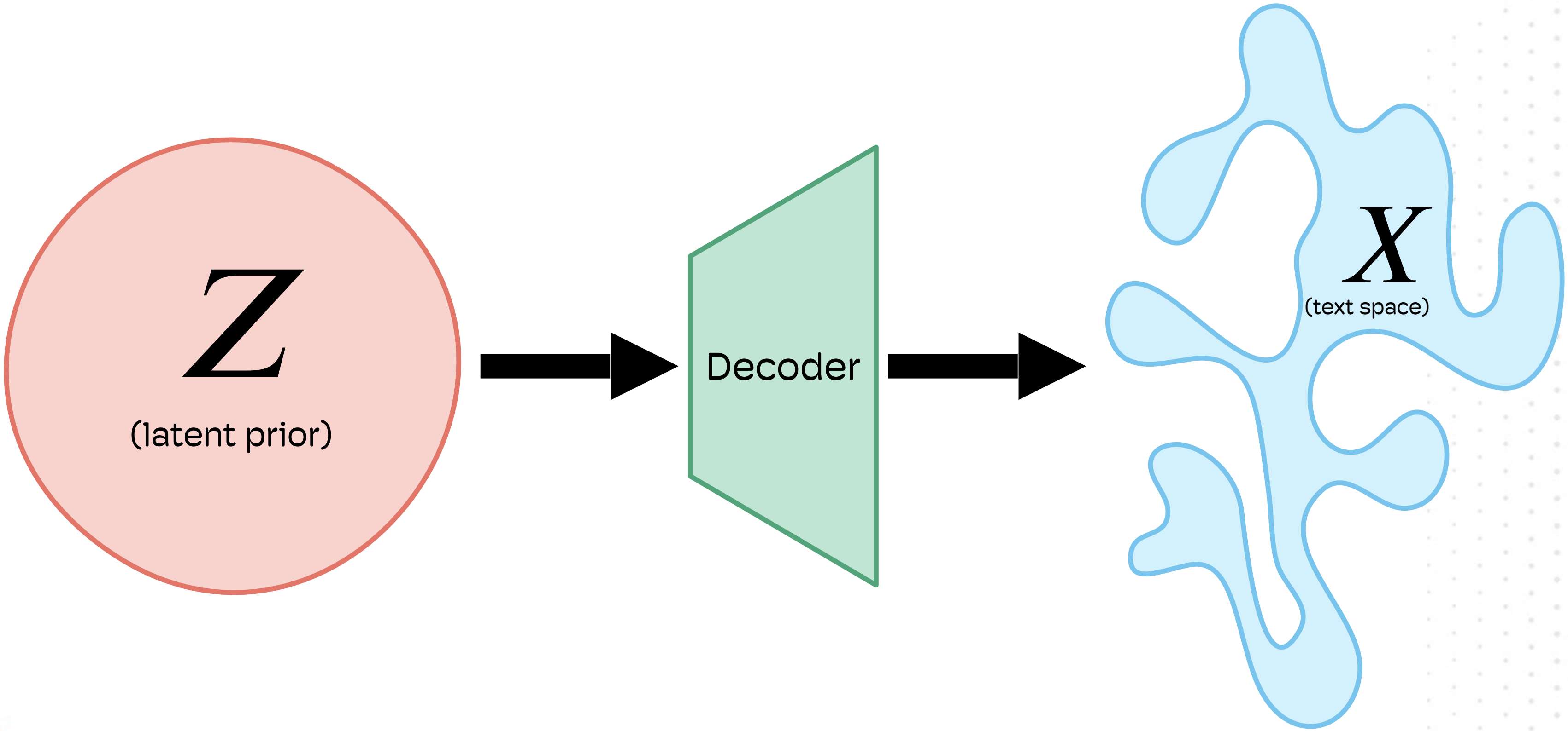
Natural Language Manifold



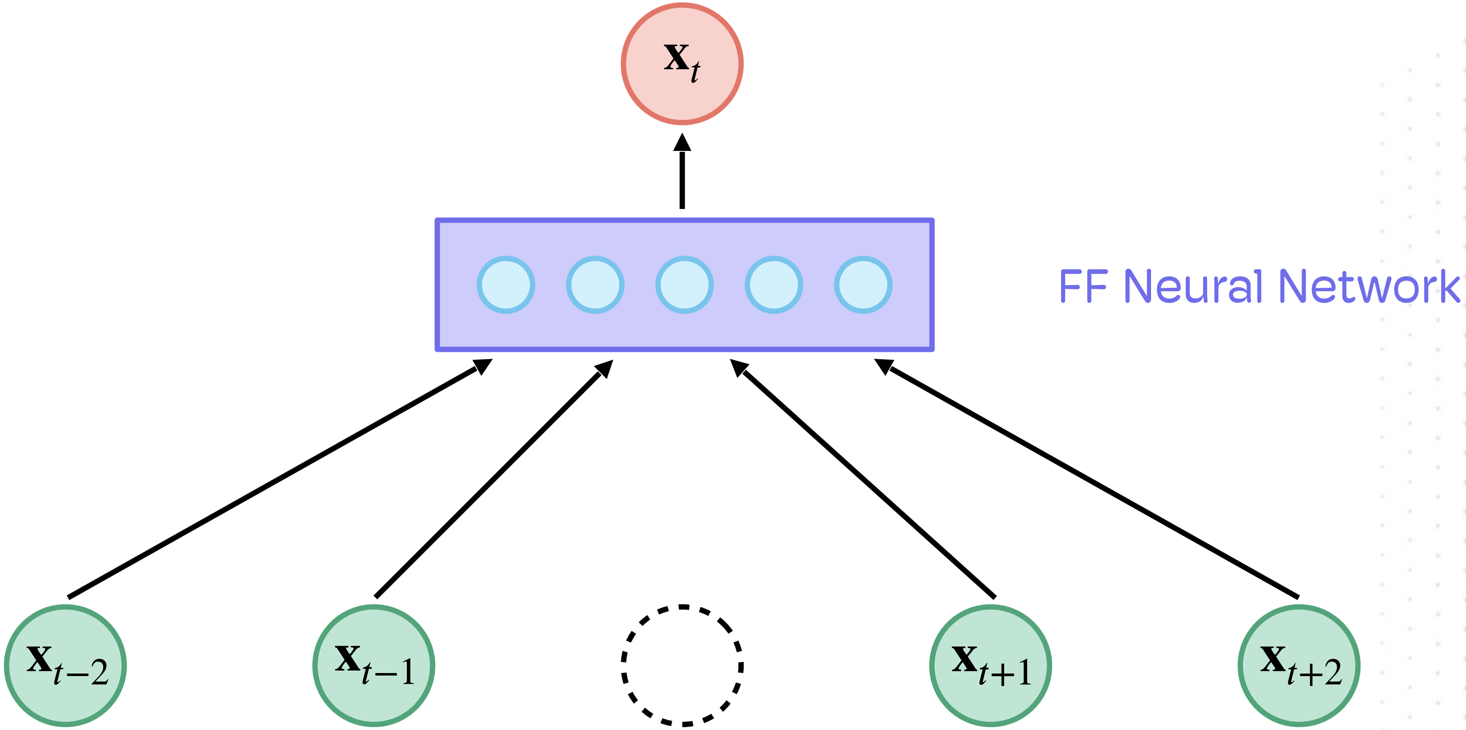
Natural Language Manifold



Natural Language Manifold

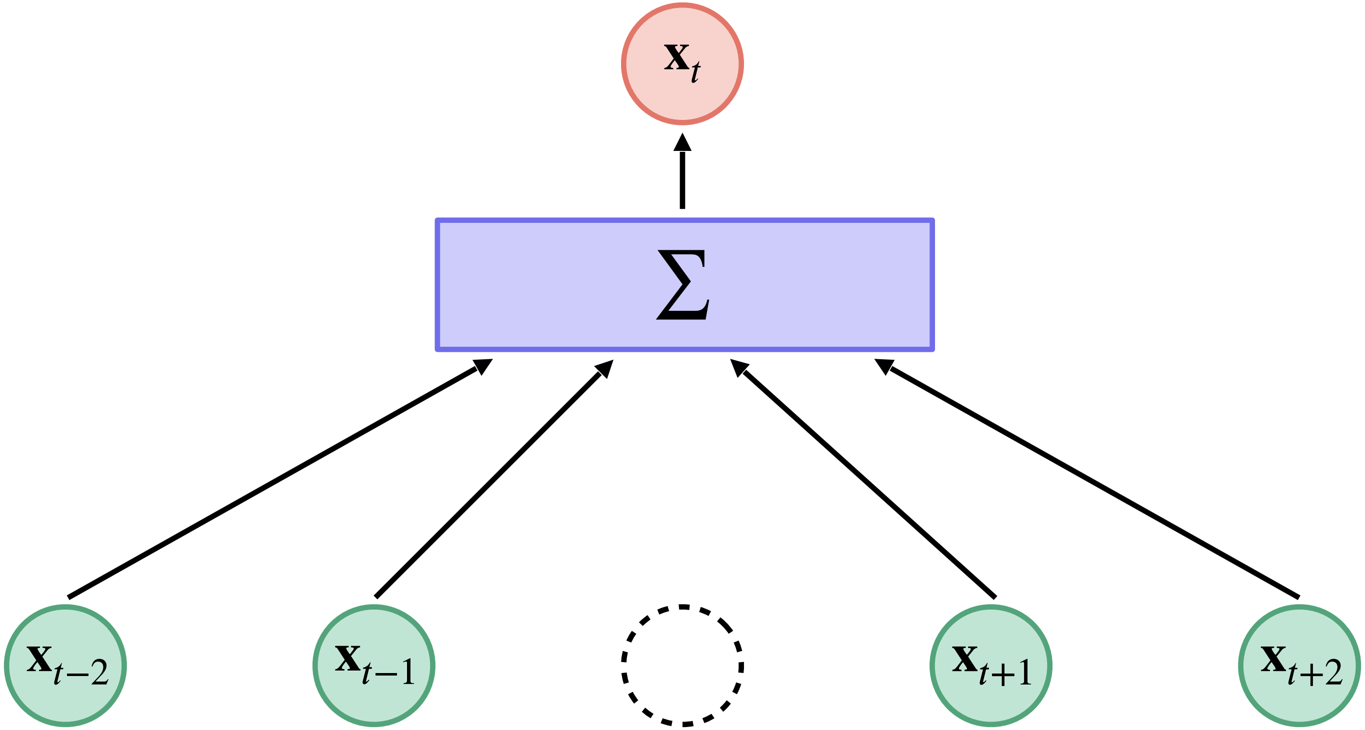


word2vec

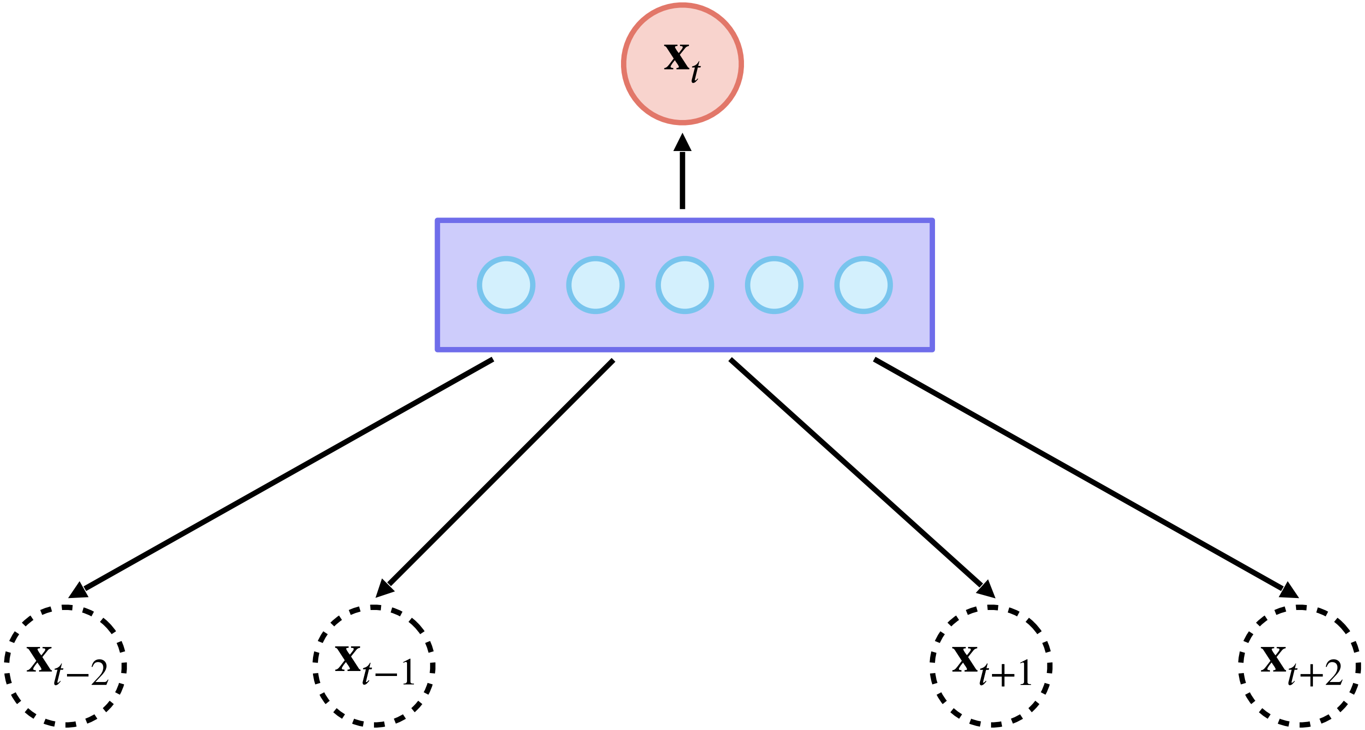


$\mathbf{x}_t$
FF Neural Network
$\mathbf{x}_{t-2}$
$\mathbf{x}_{t-1}$
$\mathbf{x}_{t+1}$
$\mathbf{x}_{t+2}$

Skip-gram



Continuous bag of words (CBOW)



Pearson

Embedding Visualizer Screen Sharing

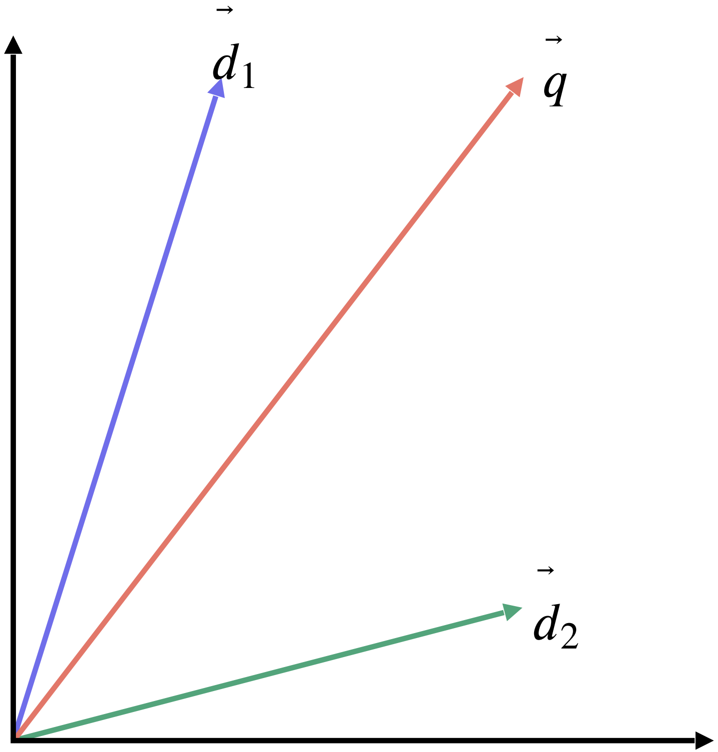
5.9

The Vector Space Model

Unsupervised Learning

- No need for labels
- Discovers latent features (hidden patterns in data)
- Often exploratory in nature
- Since there is no “gold standard” often difficult to validate model (especially with stochastic algorithms)

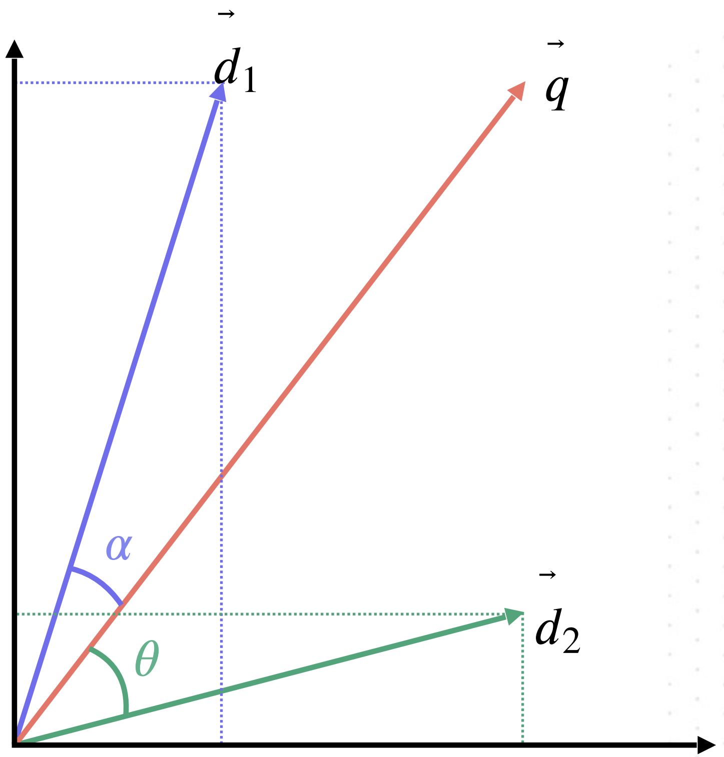
Vector Space Model



Pearson

Vector Space Model

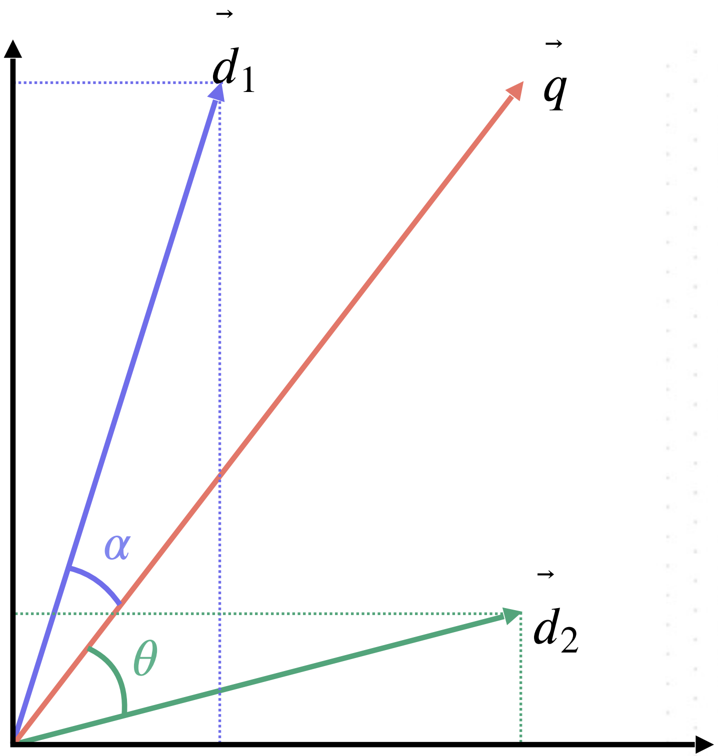
Similarity is a measure of "distance"



Pearson

Vector Space Model

$$\cos\theta = \frac{\vec{d_2} \cdot \vec{q}}{\| \vec{d_2} \| \| \vec{q} \|}$$



It’s all (almost) the same

K-means ≈ PCA ≈ LDA ≈ SVD ≈ NMF

It’s All (Almost) the Same

Huang, Heng, et al. "Simultaneous tensor subspace selection and clustering: the equivalence of high order svd and k-means clustering." *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2008.

Ding, Chris, Xiaofeng He, and Horst D. Simon. "On the equivalence of nonnegative matrix factorization and spectral clustering." *Proceedings of the 2005 SIAM International Conference on Data Mining*. Society for Industrial and Applied Mathematics, 2005.

Ding, Chris, and Xiaofeng He. "K-means clustering via principal component analysis." *Proceedings of the Twenty-first International Conference on Machine learning*. 2004.

Corrochano, Eduardo Bayro, et al. "Eigenproblems in pattern recognition." *Handbook of Geometric Computing: Applications in Pattern Recognition, Computer Vision, Neuralcomputing, and Robotics* (2005): 129-167.

5.10

Embedding Sequences with Transformers

Live Coding

5.11

Computing the Similarity Between Embeddings

Live Coding

5.12

Semantic Search with Embeddings

Live Coding

5.13

Contrastive Embeddings with Sentence Transformers

Pearson

Live Coding