Natural language processing

Lecture 20 of “Mathematics and AI”

Outline

1. Text and sequence data
2. Natural language processing

   applications, approaches
3. Statistical models

   bag of words, n-grams, latent semantic analysis
4. Neural models

   Word2vec, seq2seq

Sequence data

Sequence data

- Each observation is a sequence of objects (e.g., words)
- The order of objects within the sequence is meaningful (e.g., “Bob likes the cat.” vs “The cat likes Bob.”)
- Same value in different positions of a sequence can indicate similar meanings (e.g., “Bob hates the rain.”)
- ***How do we design a machine that can distinguish and use these subtle connections for its predictions?***
- Applications: Natural language, computer programming languages, video to text, speech, music, DNA/RNA sequences, large molecules

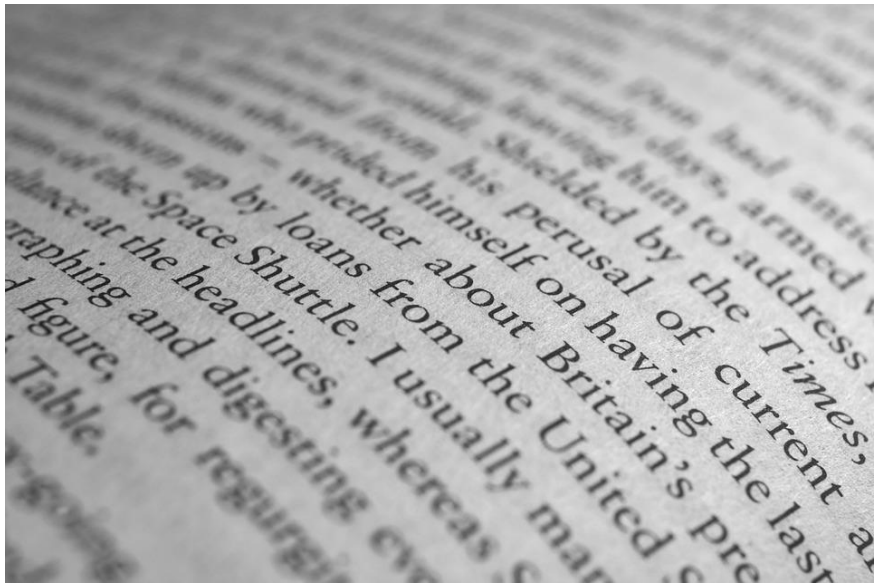


Image source: https://pixabay.com/photos/page-book-words-letters-read-3380967

Natural language processing

Natural language processing (NLP)

- Machine translation
- Text and document classification
- Sentiment analysis
- Language modeling
- Information retrieval
- Q&A systems
- …

Word cloud for “The Wizard of Oz”

Image source: https://workspace.google.com/marketplace/app/word_cloud_for_docs/58662699010

Natural language processing

- Symbolic approach (ca. 1950+)
  - Build and use sets of rules
- Statistical approach (ca. 1990+)
  - Use statistical models
- Neural approach (ca 2010+)
  - Use neural networks
- Combinations of different approaches ...

Word cloud for “The Wizard of Oz”

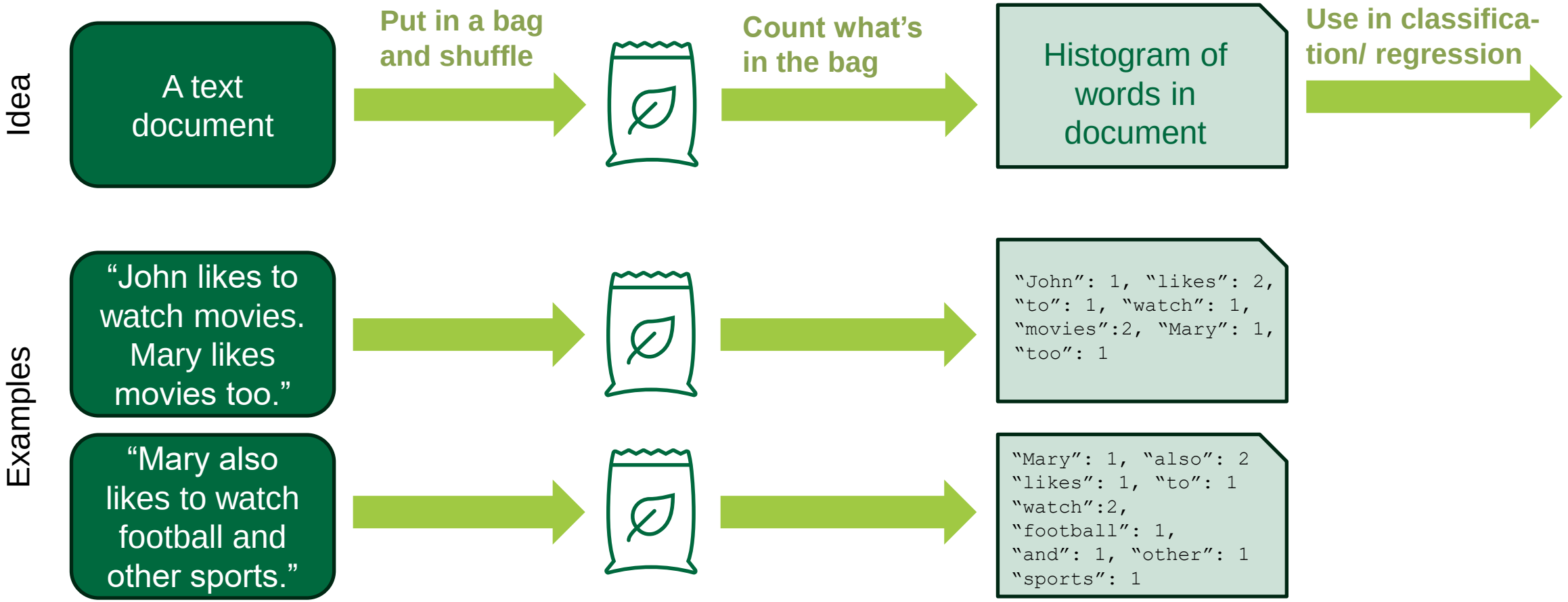
Image source: https://workspace.google.com/marketplace/app/word_cloud_for_docs/58662699010

Statistical models

Statistical approach

- Made possible by collection and/or digitalization of large ***corpora*** (i.e., text datasets)
- ***Tokenization***:
  - Represent a “unit of text” by some abstract token (e.g., integer)
  - What is the right unit of text? (character, subword, word, word sequence, sentence, etc.)
- ***Text embedding***:
  - Map tokens onto a new feature space in which distance is related to semantic similarity/dissimilarity
- Examples:
  - Text embedding: n-gram models including bag-of-words
  - Language models: latent semantic analysis

Bag-of-words



Example adapted from Wikipedia

Bag-of-words

- Every unique word w in the corpus defines a feature X_w
- For each observation (e.g., a document) the value of X_w is the frequency of w
- Encoded observation does not include positional data
- Statistical models are independent of word order

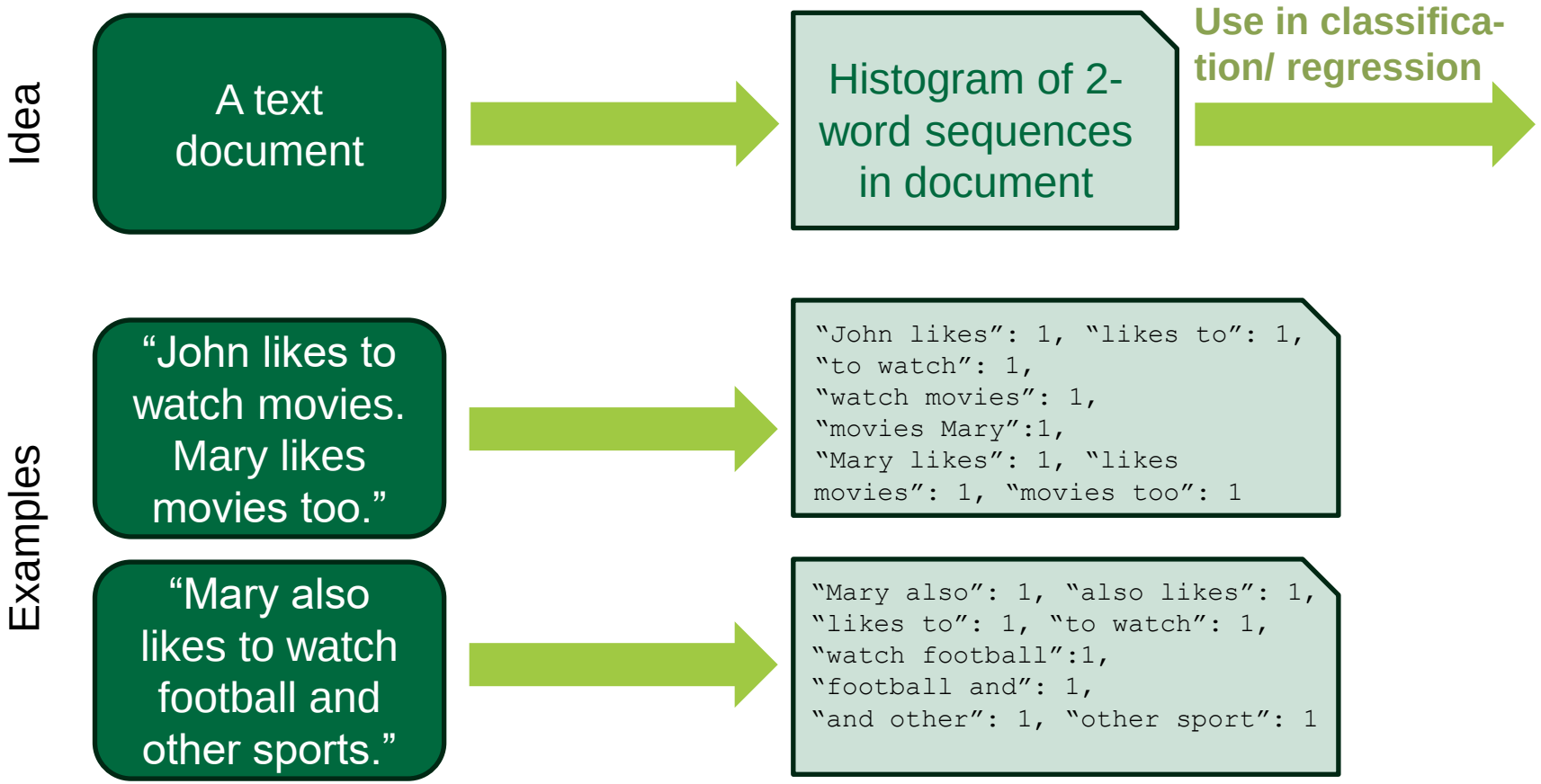
$$P(text) = P\left(X_{w_1}, X_{w_2}, X_{w_3}, \ldots\right)$$

often

$$P(text) = P(w_1)P(w_2)P(w_3)\ldots P(w_m)$$

- Suitable for text classification, sentiment analysis, etc.

Bigram model



Example adapted from Wikipedia

Bigram model

- Every unique sequence of 2 words (w_1, w_2) in the corpus defines a feature X_{w_1,w_2}
- For each observation the value of X_{w_1,w_2} is the frequency of (w_1, w_2)
- Statistical models are independent of word order

$$P(text) = P\left(X_{w_1,w_2}, X_{w_2,w_3}, X_{w_3,w_4}, \ldots, X_{w_{m-1},w_m}\right)$$

often

$$P(text) = P(w_2|w_1)P(w_3|w_2)P(w_4|w_3) \ldots P(w_m|w_{m-1})$$

N-gram models

- Every unique word of N words $\vec{w} = (w_1, w_2, \ldots, w_N)$ in the corpus defines a feature $X_{\vec{w}}$
- For each observation the value of $X_{\vec{w}}$ is the frequency of $\vec{w}$
- Statistical models are independent of word order

$$P(text) = P\left(X_{\vec{w}_1}, X_{\vec{w}_2}, X_{\vec{w}_3}, \ldots\right)$$

often

$$P(text) = P(w_N | w_1, w_2, \ldots, w_{N-1}) P(w_{N+1} | w_2, w_3, \ldots, w_N) \ldots$$

- Suitable for text completion, automated code writing, early GenAI, etc.

Language models

Latent semantic analysis

- Arrange all n observations in a training set into an occurrence matrix $(\vec{X}_1, \vec{X}_2, \ldots, \vec{X}_n)$ of words or word sequences
- Values are ***TD-IDF*** (text-frequence-inverse-document-frequency)
- Low-rank matrix approximation/ dimension reduction
- Use distance in reduced space for semantic analysis

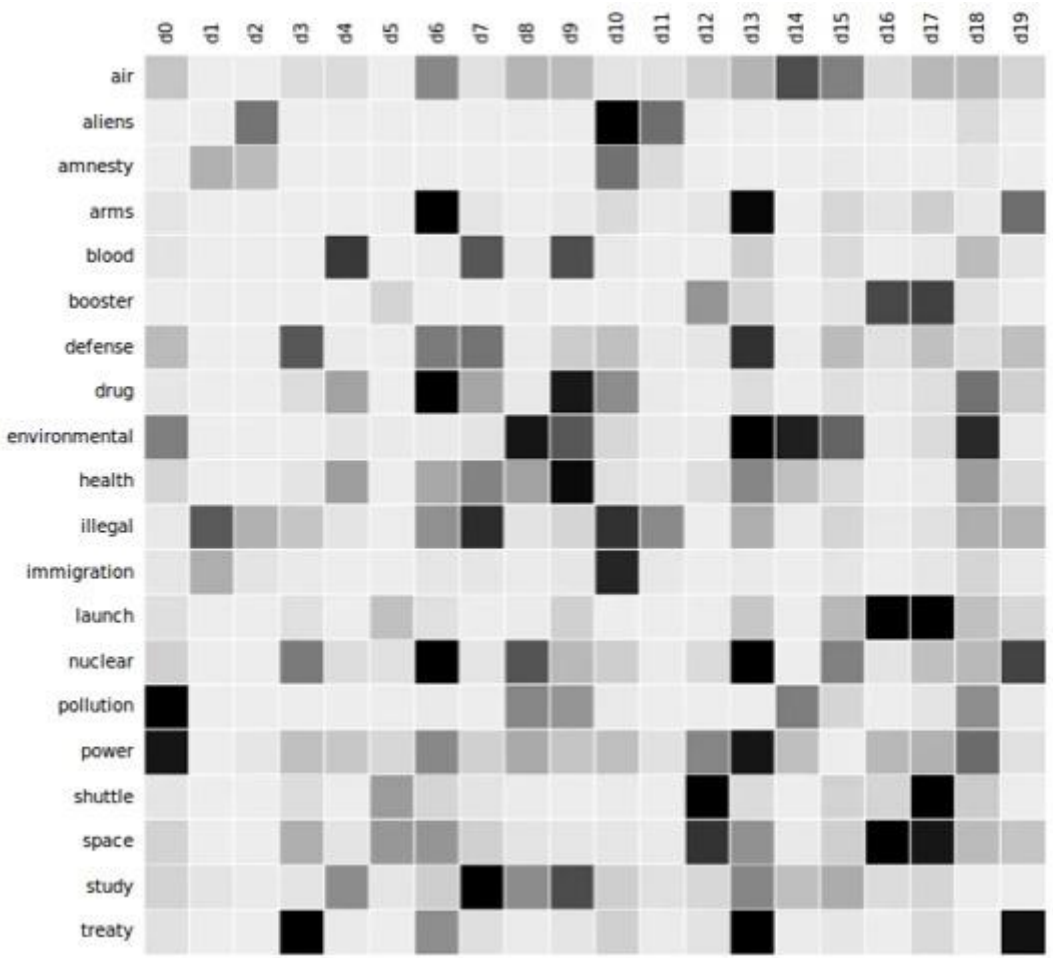


Image source: Wikipedia

Neural models

Deep learning (reviseted)

- A network with m hidden layers:

$$\vec{x}^{(k+1)} = \sigma\left(W^{(k)}\vec{x}^{(k)}\right) \text{ for } k = 1, \ldots m$$
$$\vec{x}^{(out)} = \text{softmax}\left(W^{(m)}\vec{x}^{(m)}\right)$$

- Each transformation from one layer to the next corresponds to a feature mapping
- Number of nodes in layer m corresponds to number of features in that layer

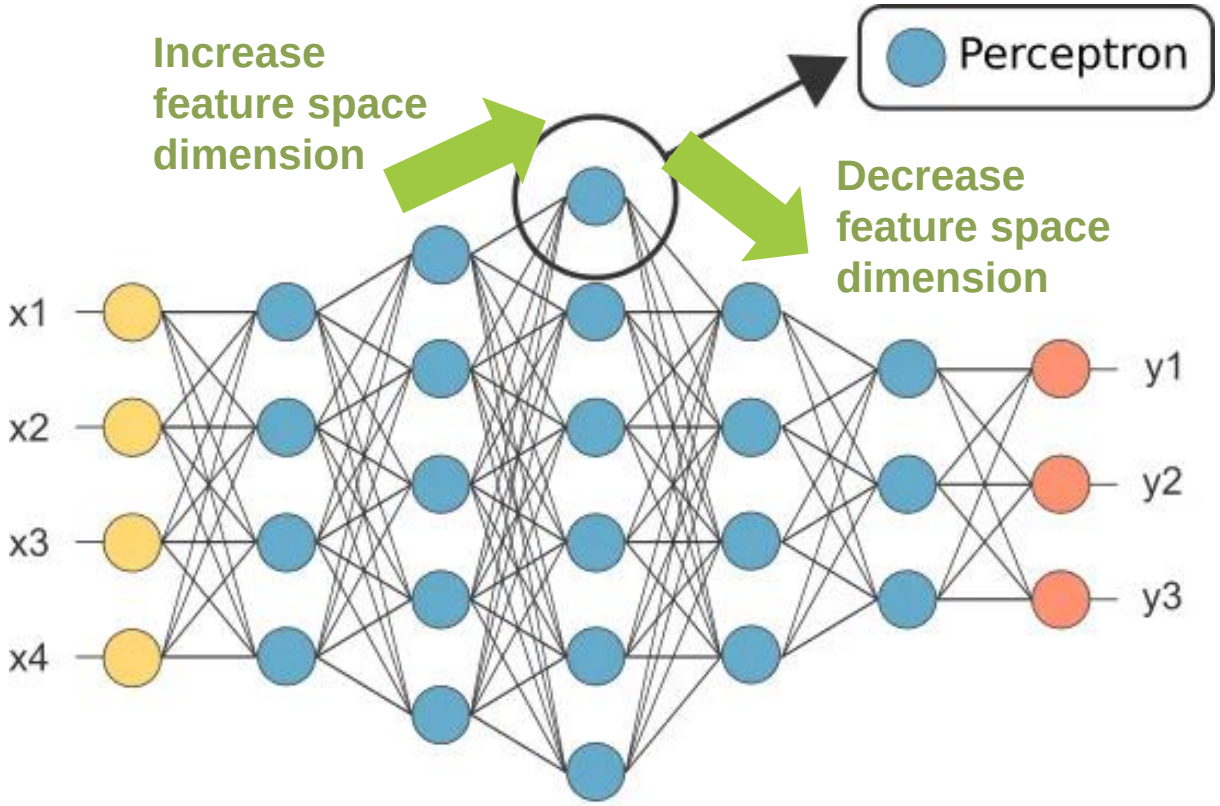
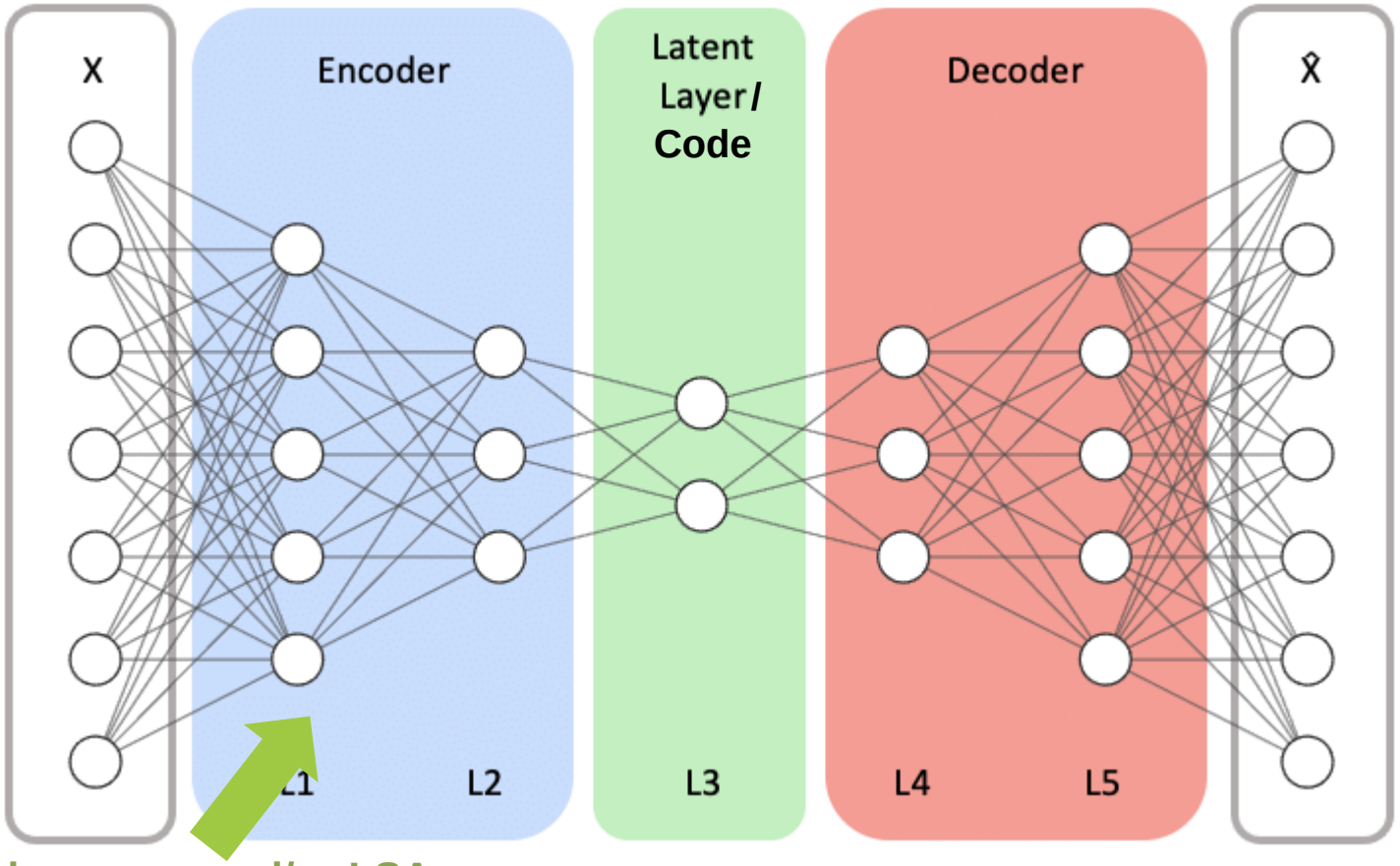


Image source: https://medium.com/@b.terryjack/introduction-to-deep-learning-feed-forward-neural-networks-ffnns-a-k-a-c688d83a309d

Encoder-decoder architecture



Can include n-gram and/or LSA

Image source: Song, Youngrok, Sangwon Hyun, and Yun-Gyung Cheong. 2021. "Analysis of Autoencoders for Network Intrusion Detection" *Sensors* 21, no. 13: 4294. https://doi.org/10.3390/s21134294

Applications of encoder-decoder architectures

- Machine translation
  - Input: Phrase in language A
  - Code: Distilled meaning of phrase
  - Output: Phrase in language B
- Dimensionality reduction via autoencoders
  - Input/output: object, identical copy of object
  - Code: Lossless representation of object in reduced feature space
- Generative AI
  - Use random code and decoder to generate new objects

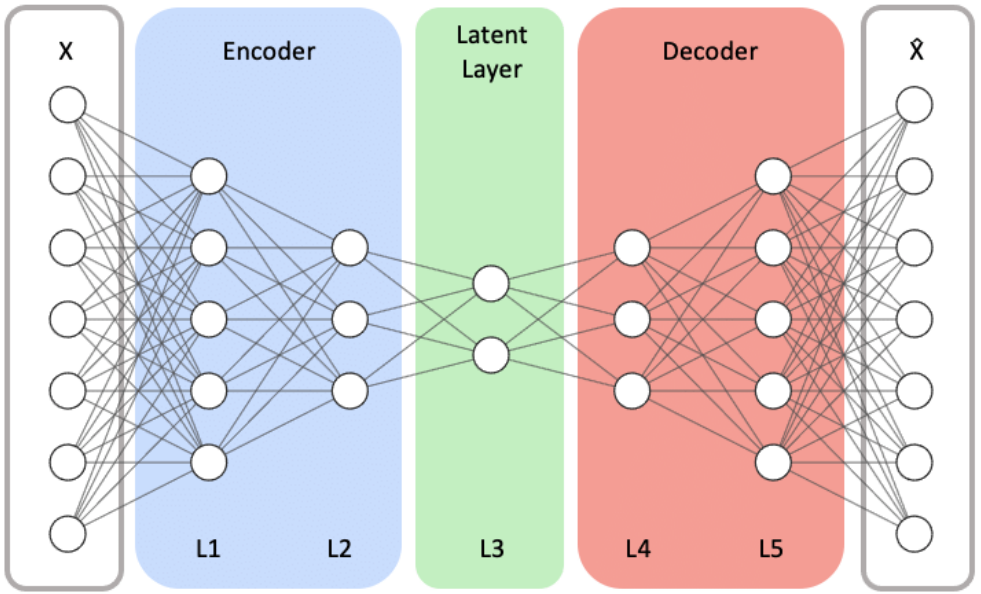


Image source: Song, Youngrok, Sangwon Hyun, and Yun-Gyung Cheong. 2021. "Analysis of Autoencoders for Network Intrusion Detection" *Sensors* 21, no. 13: 4294. https://doi.org/10.3390/s21134294

Words exist in context

- How can context be included in the code?
- RNN
  - Output depends on previous text units
- Bidirectional RNN
  - Output depends on previous and subsequent text units

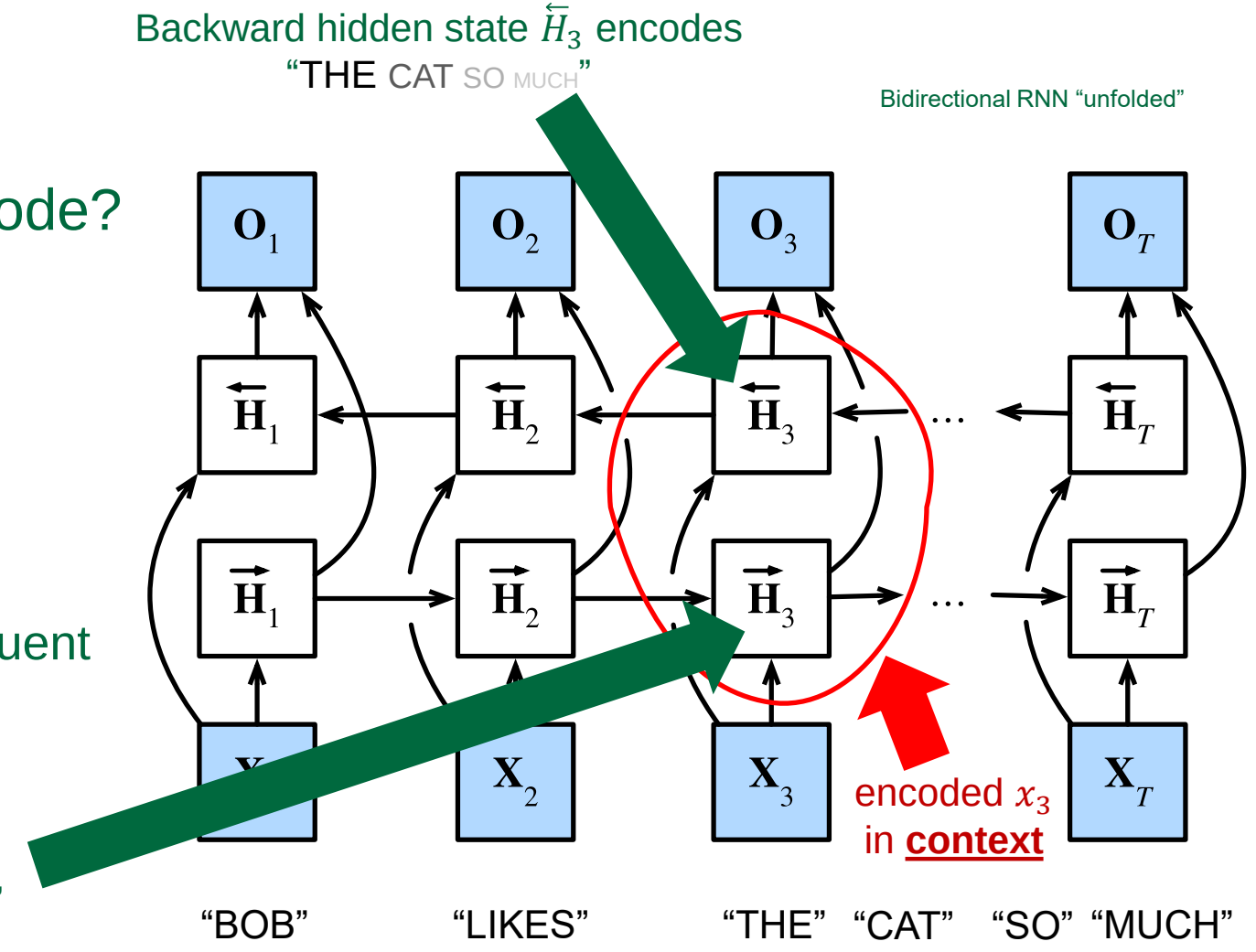


Image source: https://d2l.ai/chapter_recurrent-modern/bi-rnn.html