

Rotman

NATURAL LANGUAGE PROCESSING

March 22, 2020 Prepared by Niti / FinHUB



Rotman School of Management
UNIVERSITY OF TORONTO

Agenda

1. Intro to Natural Language Processing (NLP)
2. Basics of Data Preprocessing in NLP
 - Tokenization
 - Normalization
3. Vectorization
 - Frequency Vectors
 - TFIDF Vectors
4. Developing NLP Pipelines

1. Natural Language Processing

1.1 What is Natural Language Processing?

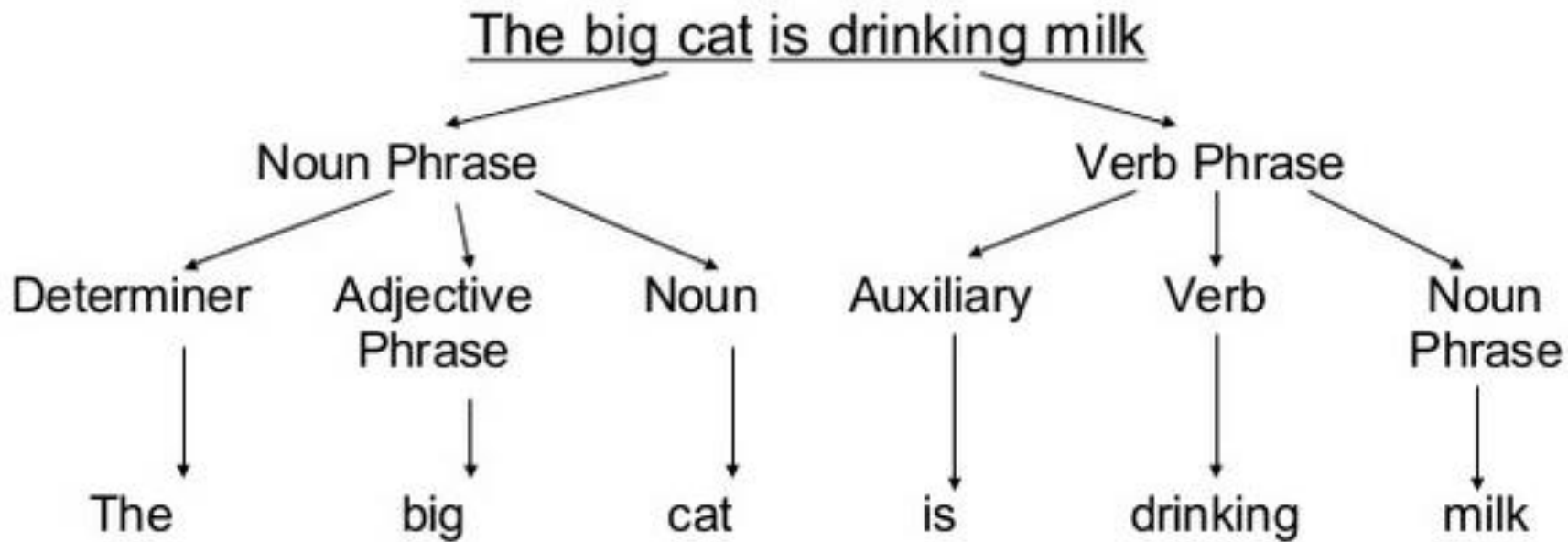
"Natural Language Processing (NLP) is a subfield of linguistics, computer science, information engineering, and artificial intelligence concerned with the interactions between computers and human (natural) languages".

- Wikipedia

1.2 What is Natural Language?

- Human languages
- Consists of :
 - **Vocabulary**, set of words
 - **Text** made of sequence of words from vocabulary
 - **Language** is constructed of a set of all possible texts

1.3 Syntactic Analysis of Natural Language



1.4 Why NLP is useful?

- Applications of NLP include
 - spam filtering
 - search engines,
 - checking spelling and grammar,
 - social website feeds,
 - speech recognition,
 - language translation, etc.
- Google Translate, for instance, is an example of NLP model

1.5 NLP Libraries in Python

- Natural language toolkit (NLTK)
- Scikit-Learn
- Gensim
- SpaCy
- TextBlob
- CoreNLP

1.6 What is NLTK?

- leading platform in Python NLP library
- provides easy-to-use interfaces to over 50 corpora and lexical resources such as WordNet
- provides a suite of text processing libraries for tokenization, stemming, tagging, parsing, semantic reasoning and an active discussion forum

2. Basic Data Preprocessing for NLP

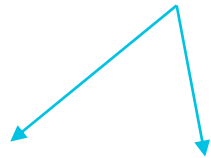
2.1 Tokenization

- Splitting text into sections
- Tokenization is the process of breaking a stream of text up into words, phrases, symbols and other **meaningful elements** called tokens

2.1 Tokenization – an example

- Using NLTK's "word_tokenize" function -

I haven't watched the show at the theatre.

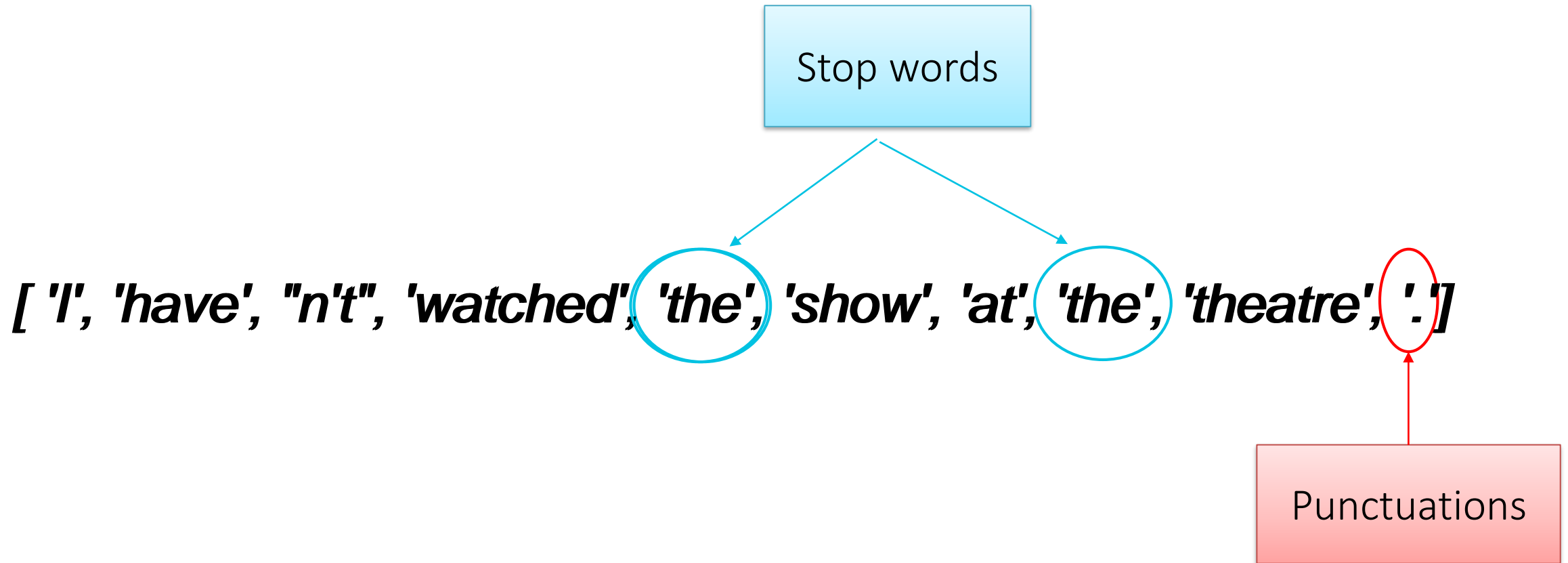


['I', 'have', "n't", 'watched', 'the', 'show', 'at', 'the', 'theatre', '.']

2.2 Normalization

- Process of transforming text into a single canonical form
- Tokenization + more
 - Convert all letters to lower or upper case
 - Removing punctuations
 - Removing white spaces
 - Removing stops words
 - Part of speech (POS) tagging
- Process of normalization is different for different corpus

2.2 Normalization



There is no universal list of stop words

2.3 Normalization – Stemming

- Process of reducing a word to its stem, base or root form
 - Stemmer, stemming, stemmed → stem
 - Girls, girl → girl
- Goal is to remove word affixes, which generally indicate plurality in Latin languages
- Stemming is useful because it is a fast feature reduction method

2.3 Normalization – Lemmatization

- Process of reducing a word to its lemma
 - gardening → to garden
 - Gardener, garden → gardener, garden
- It can handle irregular cases as well as handle tokens with different parts of speech
- Lemmatization takes time but is generally more effective in its representation

2.5 Normalization – POS Tagging

- Assigning syntactic tag to each word in a sentence

Stanford Parser

Please enter a sentence to be parsed:

Surgical resection specimens of 85 invasive ductal carcinomas of 85 women who had undergone 3D ultrasound were included.

Language:

[Sample Sentence](#)

Your query

Surgical resection specimens of 85 invasive ductal carcinomas of 85 women who had undergone 3D ultrasound were included.

Tagging

Surgical/NNP resection/NN specimens/NNS of/IN 85/CD invasive/JJ
ductal/JJ carcinomas/NNS of/IN 85/CD women/NNS who/WP had/VBD
undergone/VBN 3D/CD ultrasound/NN were/VBD included/VBN ./.

- NNP: Proper Noun, singular
- JJ: Adjective
- VBN: Verb, past participle

3. Vectorization

3.1 Vectorization

- To apply machine learning to NLP, we must convert the natural texts into numeric data i.e vectorization
- Features must represent attributes and properties of documents, such as its content as well as meta data - document length, author, source, etc.
- Vectorization creates a **high-dimensional semantic space** where documents that have similar meaning are closer together and those that are different are farther apart

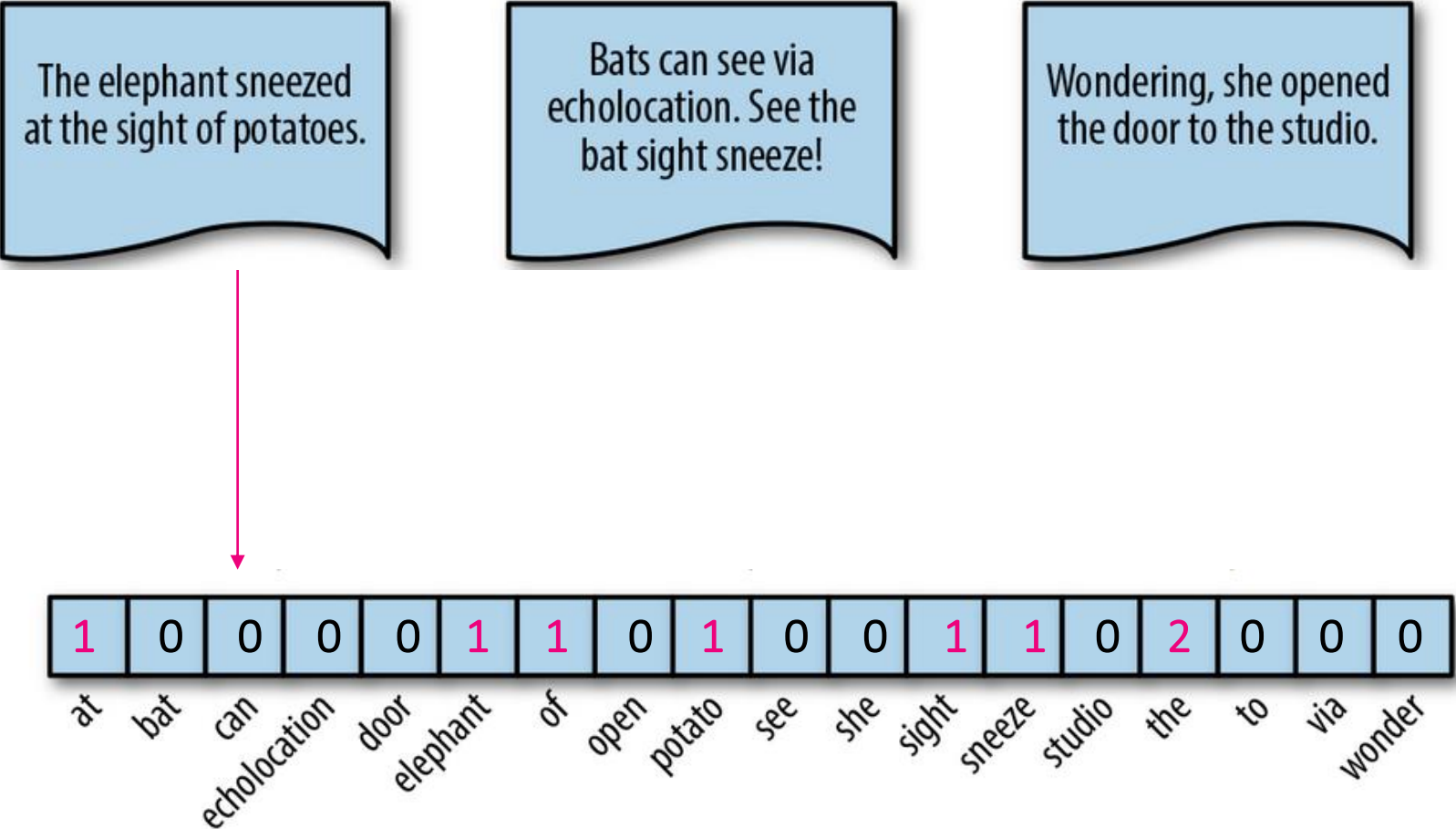
3.2 Methods of Vectorization

- Frequency vector
- One-Hot Encoding
- TFIDF
- Distributed Representation
 - Word2vec
 - Doc2vec

3.3 Vectorization – Frequency Vectors

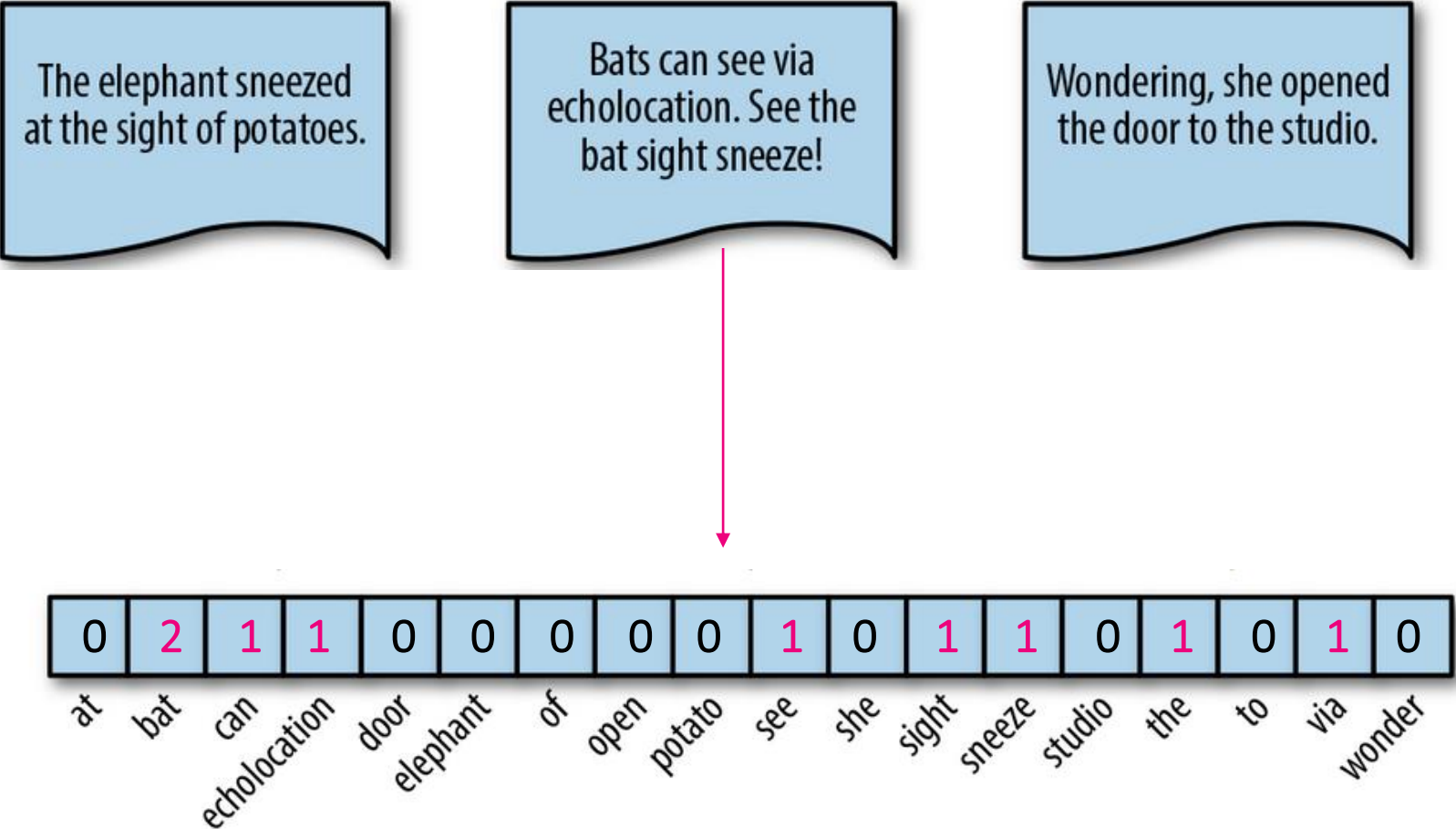
- The simplest vectorization method is the **bag of words (BOW)** model that encodes meaning and similarity based on vocabulary
- Every document from the corpus is represented as a vector whose length is equal to the vocabulary of the corpus
- The simplest vector encoding model is to simply fill in the vector with the **frequency of each word** as it appears in the document

3.3 Vectorization – Frequency Vectors



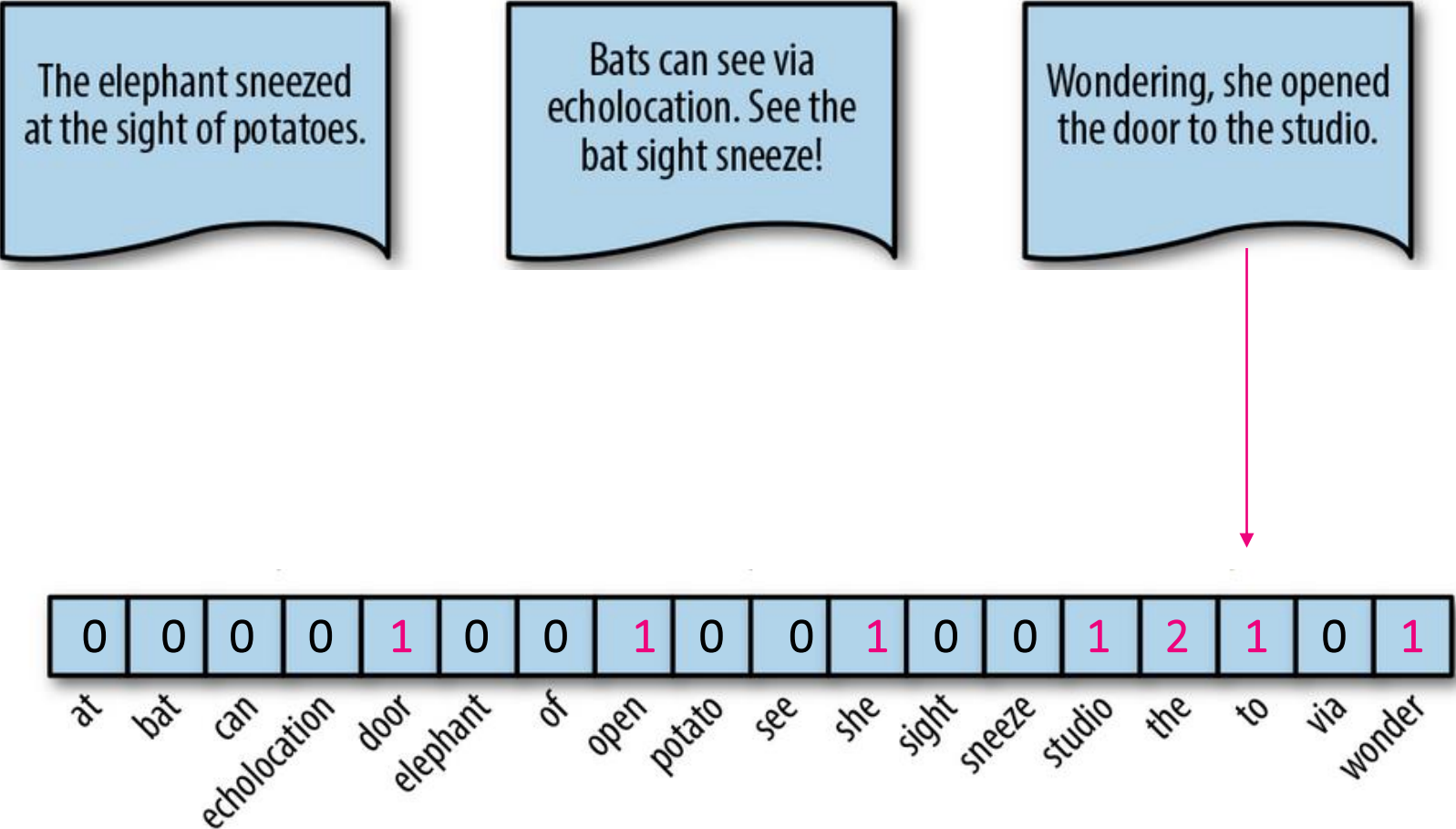
<https://www.oreilly.com/library/view/applied-text-analysis/9781491963036/ch04.html>

3.3 Vectorization – Frequency Vectors



<https://www.oreilly.com/library/view/applied-text-analysis/9781491963036/ch04.html>

3.3 Vectorization – Frequency Vectors



<https://www.oreilly.com/library/view/applied-text-analysis/9781491963036/ch04.html>

3.3 Vectorization – Frequency Vectors

Drawbacks:

- can be extremely sparse when vocabularies get larger
- significant impact on speed of ML model
- disregard grammar and the relative position of words in documents
- frequently appearing tokens are considered significant than less frequent tokens
- context of the corpus is ignored

3.4 Vectorization – TFIDF Vectors

- Are meanings most likely encoded in more rare terms from a document?
- Term Frequency - Inverse Document Frequency
- Normalizes the frequency of tokens in a document with respect to the rest of the corpus
- Emphasizes terms that are very relevant to a specific document

3.4 Vectorization – TFIDF Vectors

Two steps to measure the relevance of a token to a document

$$\text{TF-IDF} = \text{TF}(t, d) \times \text{IDF}(t)$$

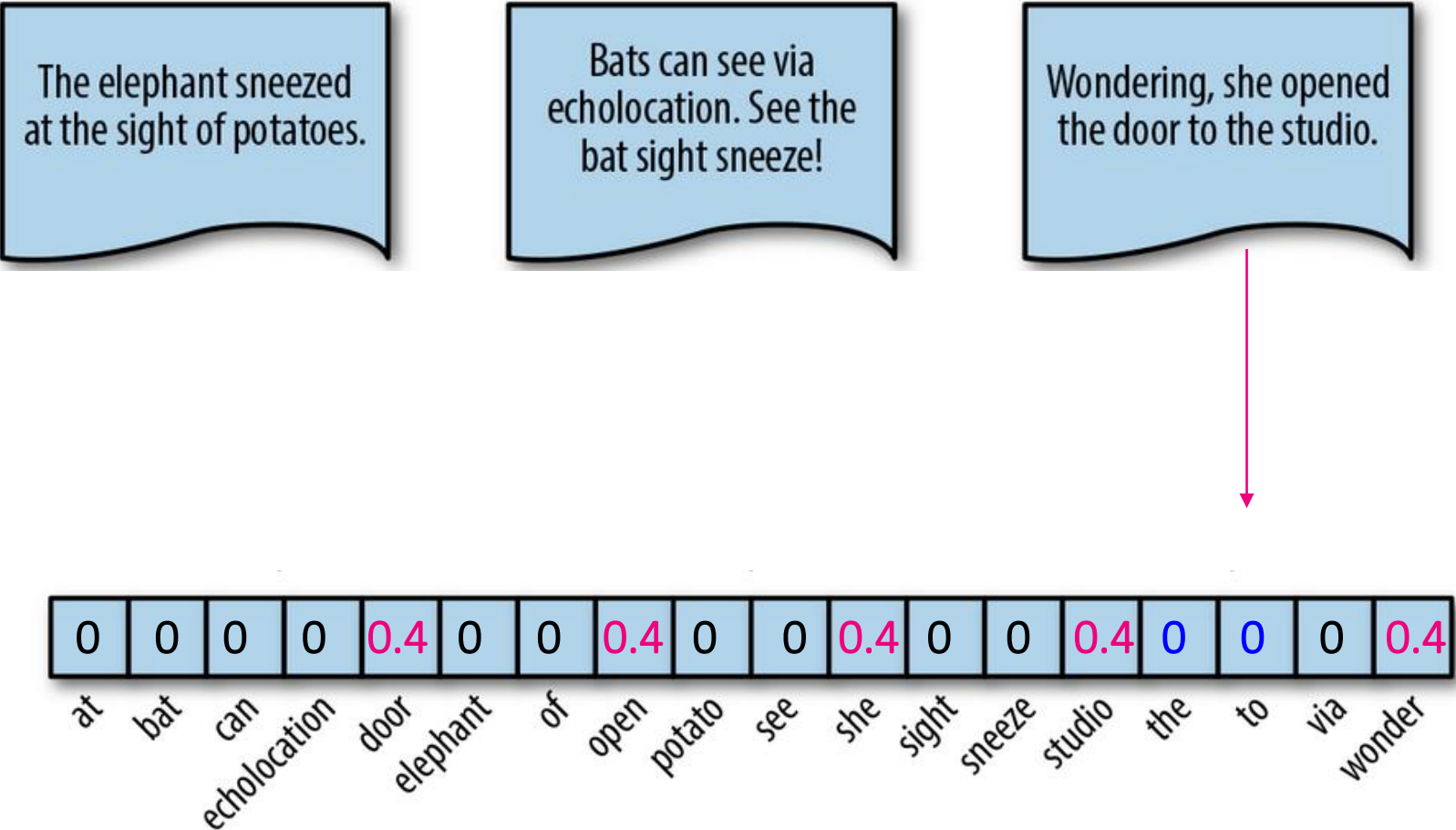
Term Frequency

Number of times term, t
appears in a document, d

Inverse Document
Frequency

$$\log \frac{1 + \text{no. of documents}}{1 + \text{df}(d, t)}$$

3.4 Vectorization – TFIDF Vectors

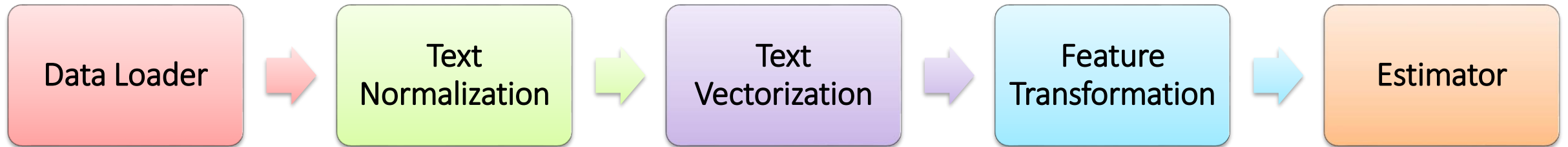


<https://www.oreilly.com/library/view/applied-text-analysis/9781491963036/ch04.html>

4. NLP Pipelines

4.1 NLP Pipelines

- Machine learning processes have series of transformers on raw data
- In each step the data is transformed to be ready for the next step until is it passed to the final estimator/classifier



4.1 NLP Pipelines

- It can become tedious to track the transformed data from one step to the next
- Pipeline objects enable us to integrate a series of transformers that combine normalization, vectorization and feature analysis into a single, well-defined mechanism

Questions?

Thank you