NLP Tasks

CMSC 473/673 - NATURAL LANGUAGE PROCESSING

Slides modified from Dr. Frank Ferraro & Dr. Jason Eisner

Learning Objectives

Define featurization & other ML terminology

Define some “classification” terminology

Distinguish between different text classification tasks

Formalize NLP Tasks at a high-level:

- What are the input/output for a particular task?
- What might the features be?
- What types of applications could the task be used for?

Similar to HW 1

Calculate elementary processes on a dataset

Logistics

HW 1 will be available after class – Due Feb 18

Homework 1: Being up to the Task

Learning Objectives

- Searching for basic information about NLP tasks.
- Exploring a dataset.
- Coming up with appropriate tasks for an application & providing your reasoning behind it.
- Determining appropriate inputs and outputs for tasks.
- Creating a system diagram.

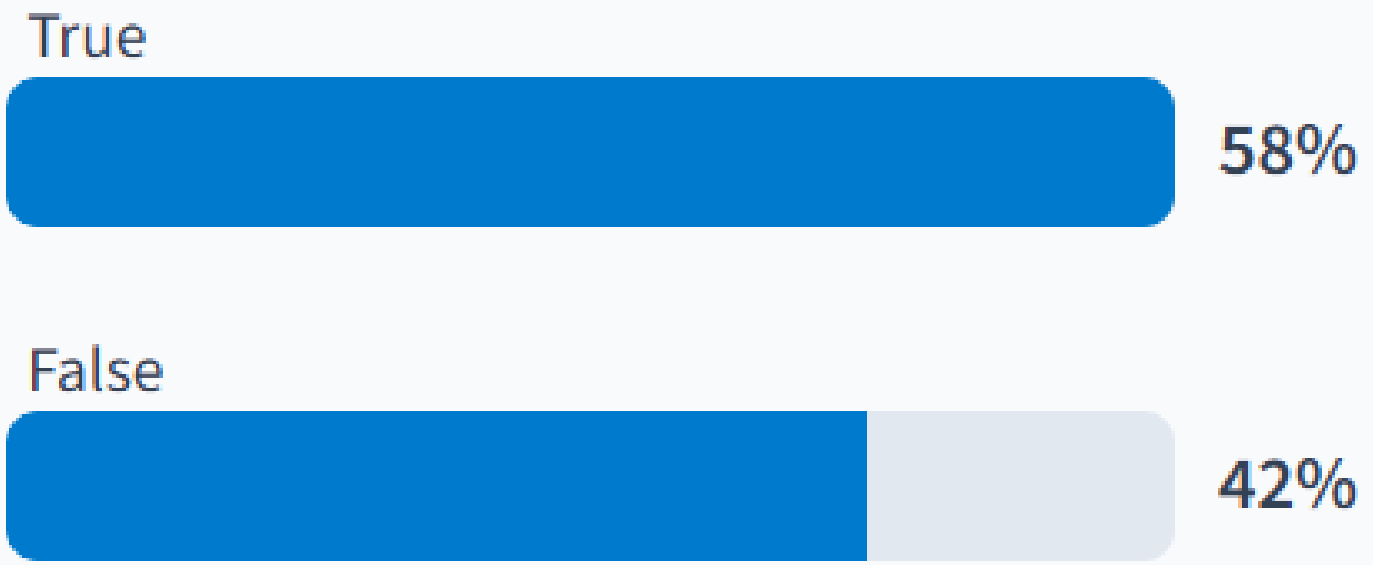
Description

You work for SuperDuperAI (SDAI), a start-up company that makes AI tools that their customers can use. You are their NLP specialist. One of SDAI's customers recently came to the company with a database of textbooks that they collected. They want SDAI to make them an app that can quiz people when they select a textbook.

The flow of the app will look like this:
a. The user types in a keyword that they're interested in, and the app finds relevant textbooks.
b. They select the textbook and chapter they want to use.
c. The app displays a question relevant to the chapter.
d. The user answers the question.
e. The app gives a numerical score for how well the user answered the question.

Being the NLP specialist on the team, **you are in charge of figuring out what is needed to create parts a, c, and e.**

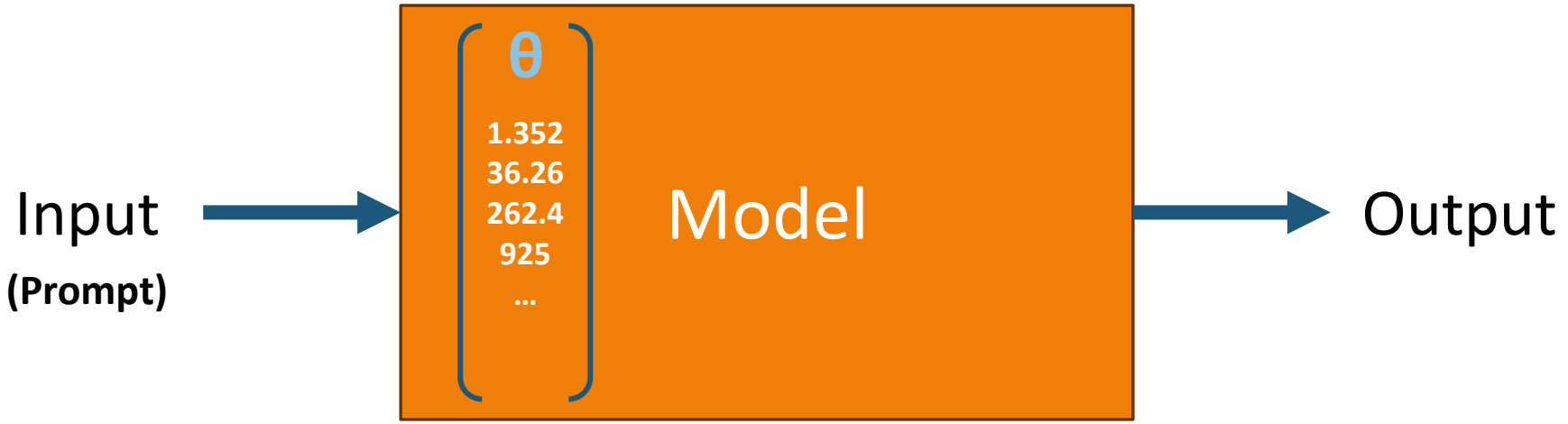
True or False: The following sentence has the same number of types as tokens (i.e., # types = # tokens)
The dog caught the frisbee .



Helpful ML Terminology

Model: the (computable) way to go from **features** (input) to labels/scores (output)

Weights/parameters (θ): vectors of numbers that control how the model produces labels/scores from inputs. These are learned through **training**.

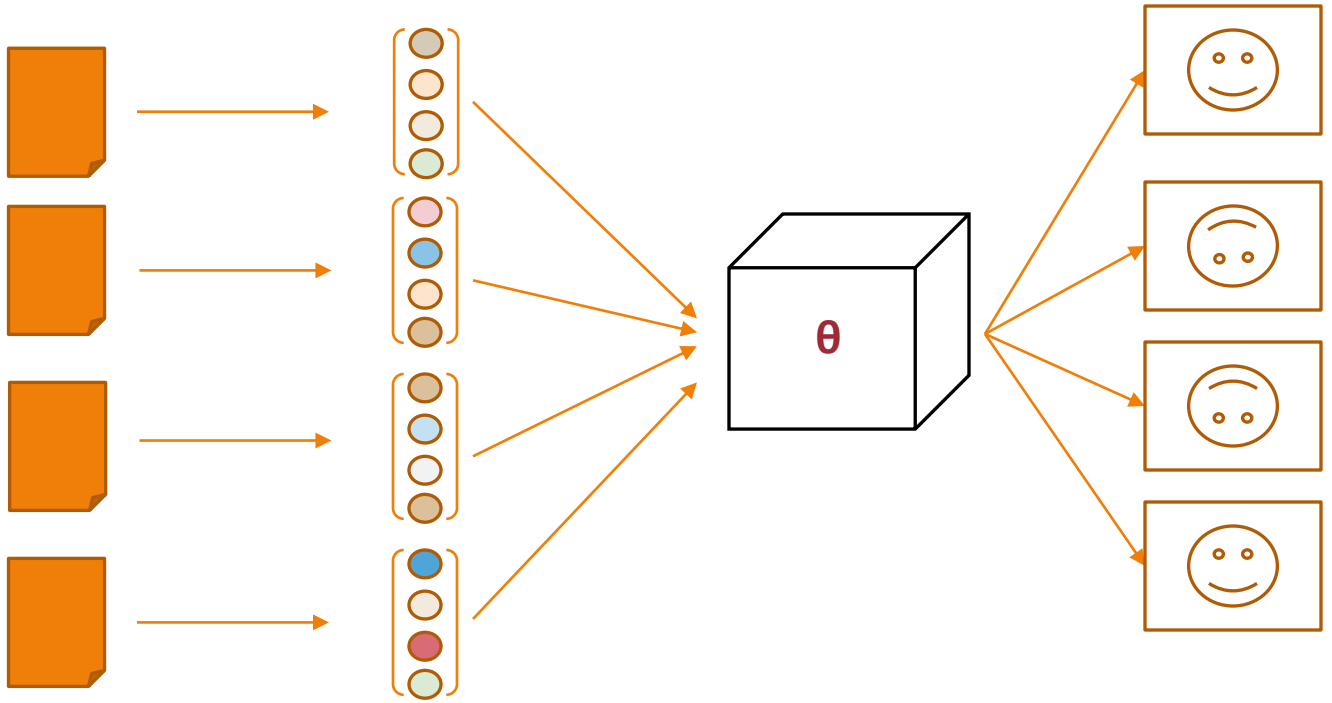


ML/NLP Framework

instances

features:
K-dimensional vector representations (one per instance)

ML model:
- take in featurized input
- output scores/labels
- contains weights θ



Helpful ML Terminology

Model: the (computable) way to go from **features** (input) to labels/scores (output)

Weights/parameters: vectors of numbers that control how the model produces labels/scores from inputs. These are learned through **training**.

Objective function: an algorithm/calculation, whose variables are the **weights** of the **model**, that we numerically optimize in order to learn appropriate weights based on the labels/scores. The **model's** weights are adjusted.

Evaluation function: an algorithm/calculation that scores how "correct" the **model's** predictions are. The **model's** weights are not adjusted.

Note: The evaluation and objective functions are often different!

(More) Helpful ML Terminology

Training / Learning:

- the process of adjusting the model's weights to learn to make good predictions.

Inference / Prediction / Decoding / Classification:

- the process of using a model's existing weights to make (hopefully!) good predictions

ML/NLP Framework for Learning

instances

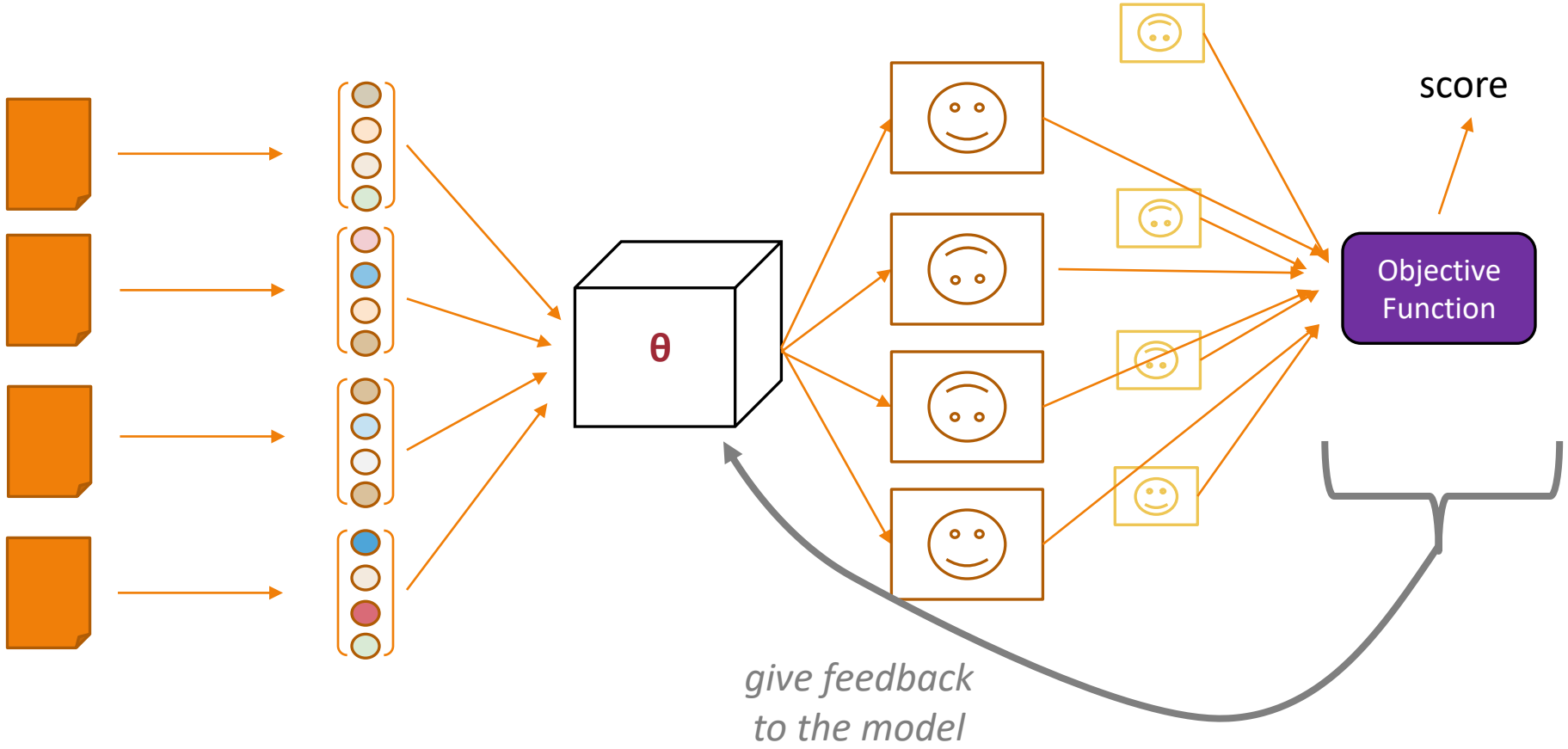
features:
K-dimensional vector representations (one per instance)

ML model:
- take in featurized input
- output scores/labels
- contains weights θ

output

"Gold" (correct) labels

Objective Function/ Learning



ML/NLP Framework for Prediction

instances

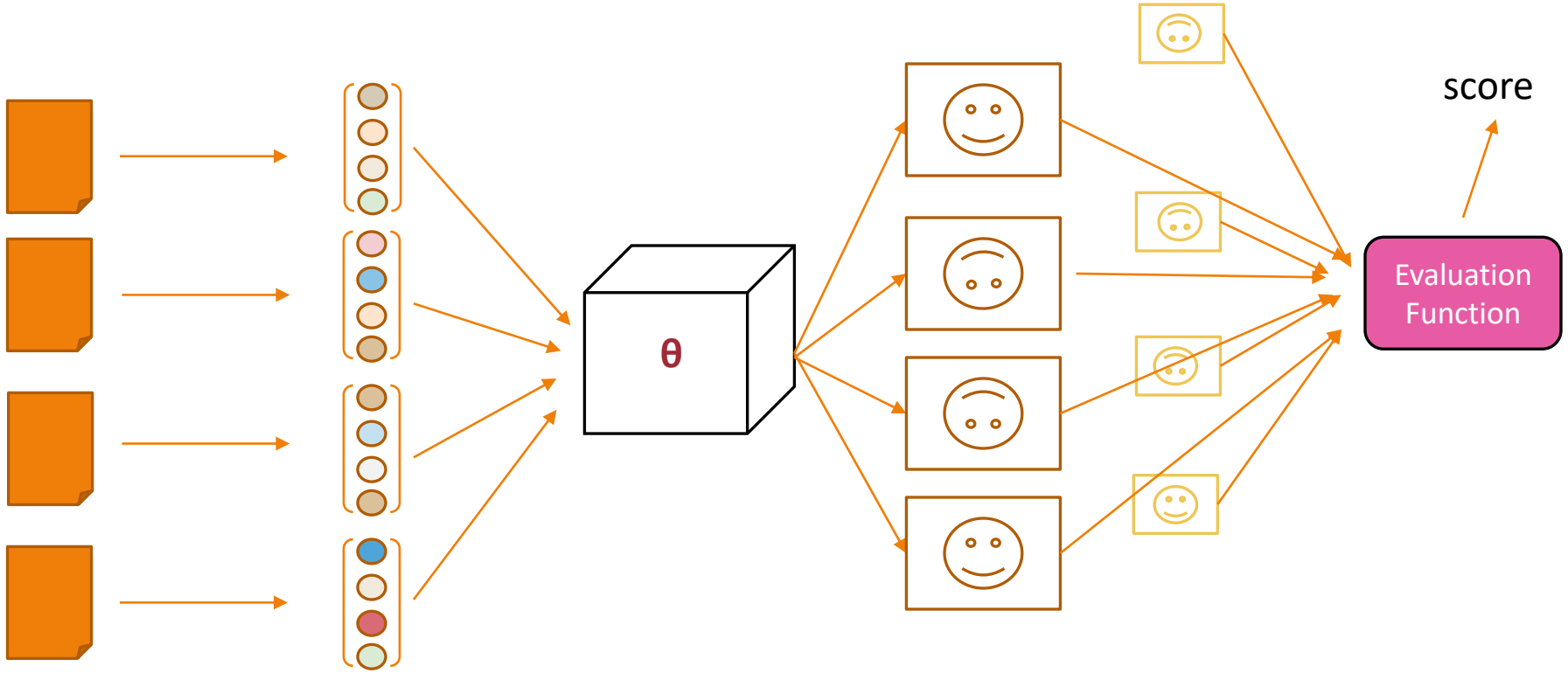
features:
K-dimensional vector representations (one per instance)

ML model:
- take in featurized input
- output scores/labels
- contains weights θ

output

"Gold" (correct) labels

Evaluation Function



ML/NLP Framework for Learning & Prediction

instances

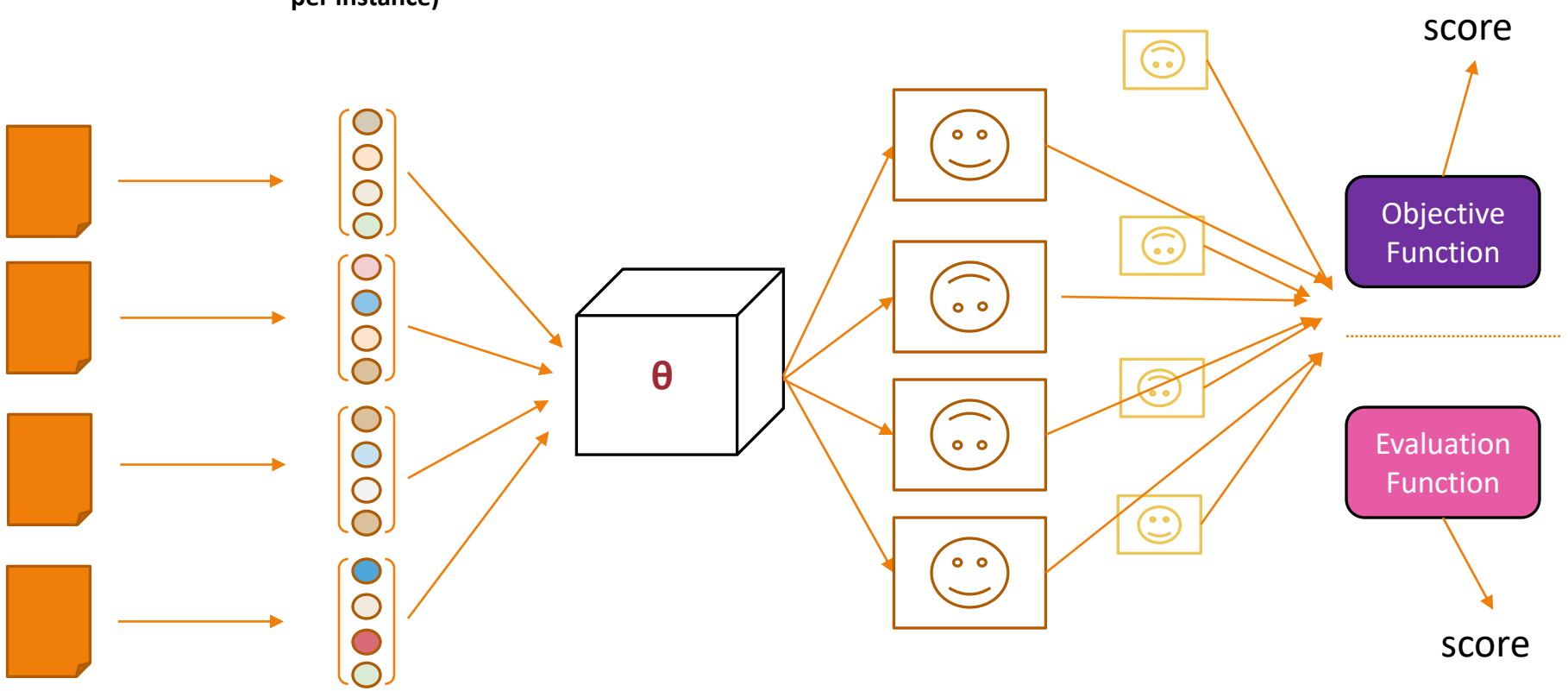
features:
K-dimensional vector representations (one per instance)

ML model:
- take in featurized input
- output scores/labels
- contains weights θ

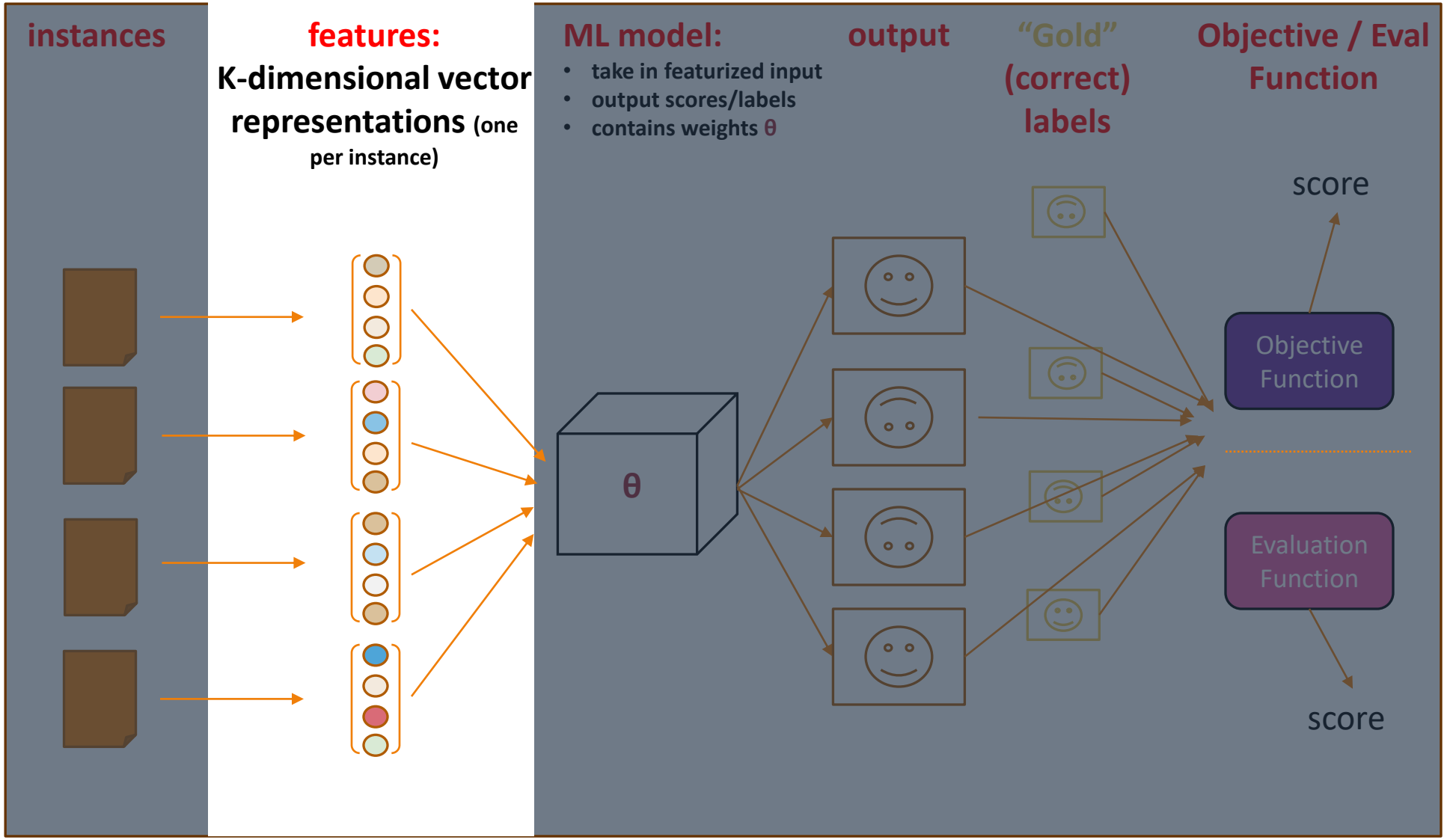
output

"Gold" (correct) labels

Objective / Eval Function



First: Featurization / Encoding / Representation



ML Term: “Featurization”

The procedure of extracting **features** for some input

Often viewed as a K-dimensional vector function f of the input language x

$$f(x) = (f_1(x), \ldots, f_K(x))$$

Each of these is a feature
(/feature function)

ML Term: “Featurization”

The procedure of extracting **features** for some input

Often viewed as a K-dimensional vector function f of the input language x

$$f(x) = (f_1(x), \dots, f_K(x))$$

In supervised settings, it can equivalently be viewed as a K-dimensional vector function f of the input language x and a potential label y

- $f(x, y) = (f_1(x, y), \dots, f_K(x, y))$

Features can be thought of as “soft” rules

- E.g., positive sentiments tweets may be *more likely* to have the word “happy”

Defining Appropriate Features

Feature functions help extract useful features (characteristics) of the data

They turn data into numbers

Features that are not 0 are said to have fired

Defining Appropriate Features

Feature functions help extract useful features (characteristics) of the data

They turn data into numbers

Features that are not 0 are said to have fired

You can define classes of features by templating (we'll come back to this!)

Often binary-valued (0 or 1), but can be real-valued

Three Common Types of Featurization in NLP

1. Bag-of-words (or bag-of-characters, bag-of-relations)

2. Linguistically-inspired features

3. Dense features via embeddings

Three Common Types of Featurization in NLP

1. Bag-of-words (or bag-of-characters, bag-of-relations)

- easy to define / extract
- sometimes still very useful

2. Linguistically-inspired features

3. Dense features via embeddings

Three Common Types of Featurization in NLP

1. Bag-of-words (or bag-of-characters, bag-of-relations)
   - easy to define / extract
   - sometimes still very useful
2. Linguistically-inspired features
   - harder to define
   - helpful for interpretation
   - depending on task: conceptually helpful
   - currently, not freq. used
3. Dense features via embeddings

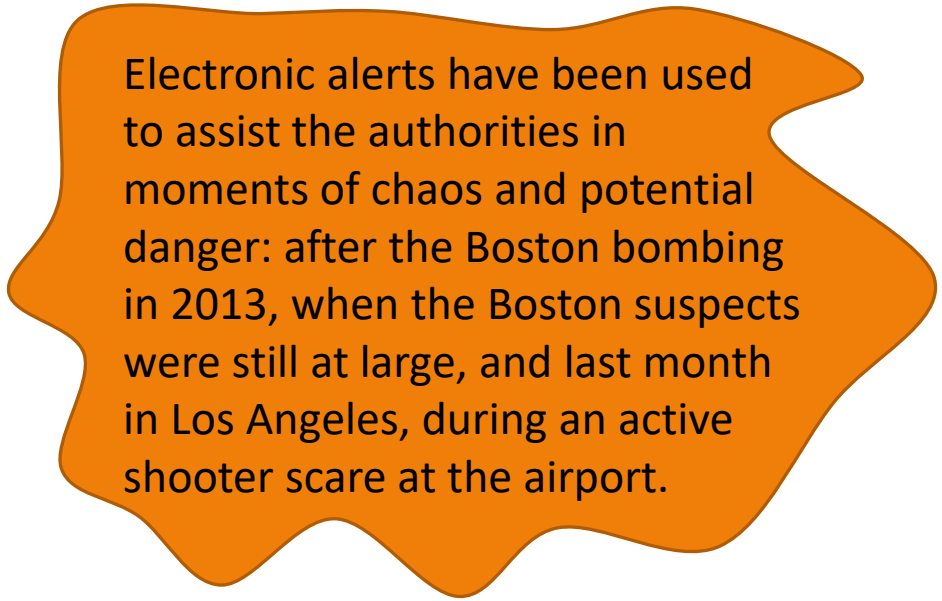
Three Common Types of Featurization in NLP

1. Bag-of-words (or bag-of-characters, bag-of-relations)
   - easy to define / extract
   - sometimes still very useful
2. Linguistically-inspired features
   - harder to define
   - helpful for interpretation
   - depending on task: conceptually helpful
   - currently, not freq. used
3. Dense features via embeddings
   - harder to define
   - harder to extract (unless there's a model to run)
   - currently: freq. used

Three Common Types of Featurization in NLP

1. Bag-of-words (or bag-of-characters, bag-of-relations)
   - Identify ***unique*** sufficient atomic sub-parts (e.g., words in a document)
   - Define simple features over these, e.g.,
     - Binary (0 or 1) ➔ indicating presence
     - Natural numbers ➔ indicating number of times in a context
     - Real-valued ➔ various other score (we'll see examples throughout the semester)
2. Linguistically-inspired features
3. Dense features via embeddings

Example: Document Classification via Bag-of-Words Features



TECH

NOT TECH

Let's make a core assumption: the **label** can be predicted from **counts of individual word types**

Example: Document Classification via Bag-of-Words Features

Electronic alerts have been used to assist the authorities in moments of chaos and potential danger: after the Boston bombing in 2013, when the Boston suspects were still at large, and last month in Los Angeles, during an active shooter scare at the airport.

TECH

NOT TECH

Q: What types of words would be features to predict "Tech" and "not Tech"?

Let's make a core assumption: the **label** can be predicted from **counts of individual word types**

Example: Document Classification via Bag-of-Words Features

Electronic alerts have been used to assist the authorities in moments of chaos and potential danger: after the Boston bombing in 2013, when the Boston suspects were still at large, and last month in Los Angeles, during an active shooter scare at the airport.

feature extraction

TECH

NOT TECH

With V word types, define V feature functions $f_i(x)$ as

$f_i(x)$ =# of times word type *i* appears in document x

Core assumption: the label can be predicted from counts of individual word types

Example: Document Classification via Bag-of-Words Features

Electronic alerts have been used to assist the authorities in moments of chaos and potential danger: after the Boston bombing in 2013, when the Boston suspects were still at large, and last month in Los Angeles, during an active shooter scare at the airport.

TECH

NOT TECH

feature extraction

$$f(x) = \left(f_i(x)\right)_i^V$$

With V word types, define V feature functions $f_i(x)$ as

$f_i(x)$ =# of times word type *i* appears in document x

Core assumption: the label can be predicted from counts of individual word types

Example: Document Classification via Bag-of-Words Features

Electronic alerts have been used to assist the authorities in moments of chaos and potential danger: after the Boston bombing in 2013, when the Boston suspects were still at large, and last month in Los Angeles, during an active shooter scare at the airport.

feature extraction

feature $f_i(x)$	value
alerts	1
assist	1
bombing	1
Boston	2
…	
sniffle	0
…	

TECH

NOT TECH

Core assumption: the label can be predicted from counts of individual word types

Example: Document Classification via Bag-of-Words Features

Electronic alerts have been used to assist the authorities in moments of chaos and potential danger: after the Boston bombing in 2013, when the Boston suspects were still at large, and last month in Los Angeles, during an active shooter scare at the airport.

TECH

NOT TECH

f(**x**): "bag of words"

feature $f_i(x)$	value
alerts	1
assist	1
bombing	1
Boston	2
…	
sniffle	0
…	

w: weights

feature	weight
alerts	.043
assist	-0.25
bombing	0.8
Boston	-0.00001
…	

Three Common Types of Featurization in NLP

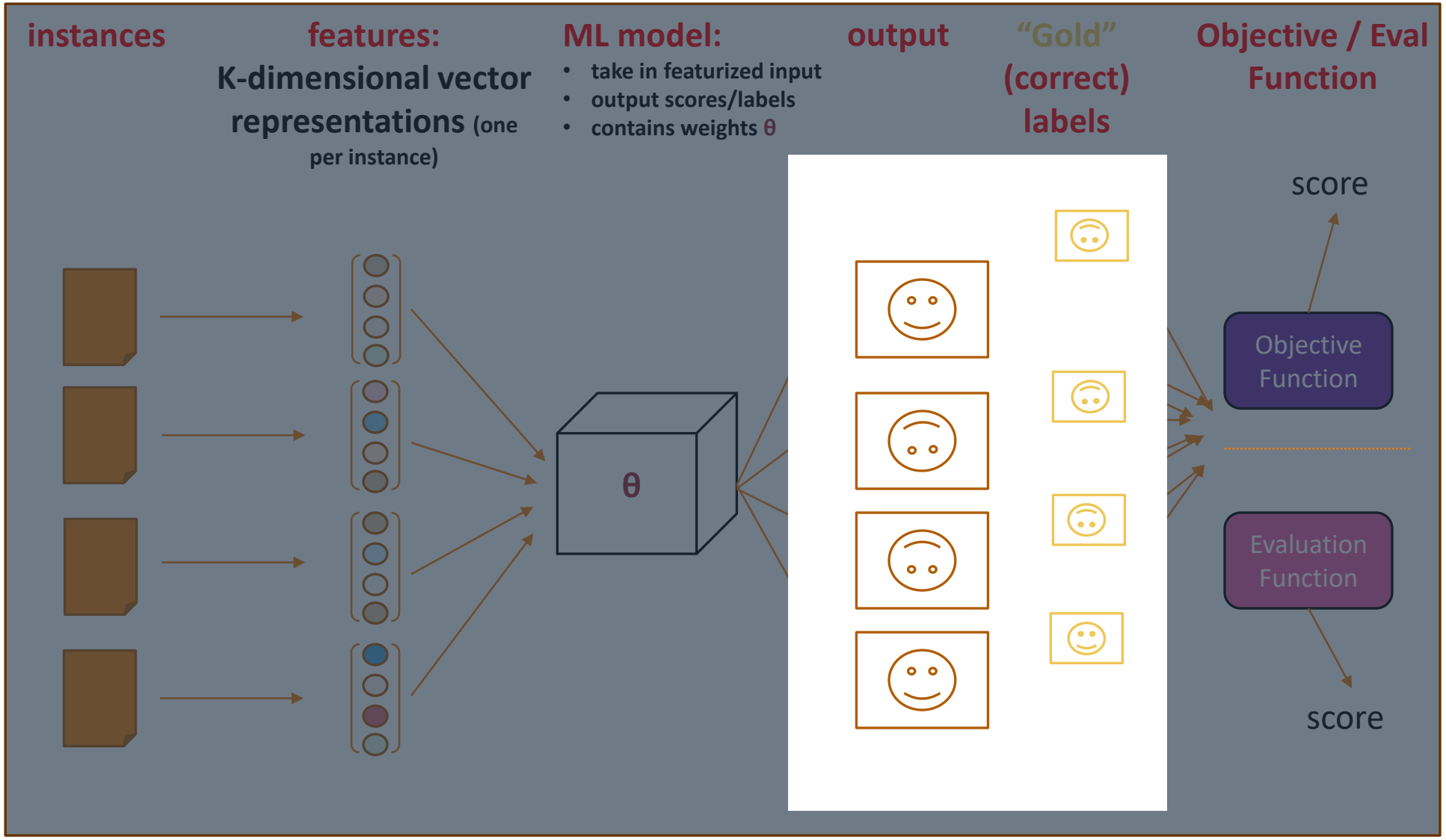
1. Bag-of-words (or bag-of-characters, bag-of-relations)
   - Identify ***unique*** sufficient atomic sub-parts (e.g., words in a document)
   - Define simple features over these, e.g.,
     - Binary (0 or 1) ➔ indicating presence
     - Natural numbers ➔ indicating number of times in a context
     - Real-valued ➔ various other score (we'll see examples throughout the semester)
2. Linguistically-inspired features
   - Define features from words, word spans, or linguistic-based annotations extracted from the document
3. Dense features via embeddings

Three Common Types of Featurization in NLP

1. Bag-of-words (or bag-of-characters, bag-of-relations)
   - Identify ***unique*** sufficient atomic sub-parts (e.g., words in a document)
   - Define simple features over these, e.g.,
     - Binary (0 or 1) ➔ indicating presence
     - Natural numbers ➔ indicating number of times in a context
     - Real-valued ➔ various other score (we'll see examples throughout the semester)
2. Linguistically-inspired features
   - Define features from words, word spans, or linguistic-based annotations extracted from the document
3. Dense features via embeddings
   - Compute/extract a real-valued vector, e.g., from word2vec, ELMO, BERT, …

Will be discussed in a future lecture

Second: Classification Terminology



Classification Types (Terminology)

Name	Number of Tasks (Domains) Labels are Associated with	# Label Types	Example
(Binary) Classification			
Multi-class Classification			
Multi-label Classification			
Multi-task Classification			

Classification Types (Terminology)

Name	Number of Tasks (Domains) Labels are Associated with	# Label Types	Example
(Binary) Classification	1	2	Sentiment: Choose one of {positive or negative}
Multi-class Classification			
Multi-label Classification			
Multi-task Classification			

Classification Types (Terminology)

Name	Number of Tasks (Domains) Labels are Associated with	# Label Types	Example
(Binary) Classification	1	2	Sentiment: Choose one of {positive or negative}
Multi-class Classification	1	> 2	Part-of-speech: Choose one of {Noun, Verb, Det, Prep, …}
Multi-label Classification			
Multi-task Classification			

Classification Types (Terminology)

Name	Number of Tasks (Domains) Labels are Associated with	# Label Types	Example
(Binary) Classification	1	2	Sentiment: Choose one of {positive or negative}
Multi-class Classification	1	> 2	Part-of-speech: Choose one of {Noun, Verb, Det, Prep, …}
Multi-label Classification	1	> 2	Sentiment: Choose multiple of {positive, angry, sad, excited, …}
Multi-task Classification			

Classification Types (Terminology)

Name	Number of Tasks (Domains) Labels are Associated with	# Label Types	Example
(Binary) Classification	1	2	Sentiment: Choose one of {positive or negative}
Multi-class Classification	1	> 2	Part-of-speech: Choose one of {Noun, Verb, Det, Prep, …}
Multi-label Classification	1	> 2	Sentiment: Choose multiple of {positive, angry, sad, excited, …}
Multi-task Classification	> 1	Per task: 2 or > 2 (can apply to binary or multi-class)	Task 1: part-of-speech Task 2: named entity tagging … ---------- Task 1: document labeling Task 2: sentiment

Text Annotation Tasks ("Classification" Tasks)

1. Classify the entire document ("text categorization")
2. Classify word tokens individually
3. Classify word tokens in a sequence
4. Identify phrases ("chunking")
5. Syntactic annotation (parsing)
6. Semantic annotation
7. Text generation

Slide courtesy Jason Eisner, with mild edits

Text Annotation Tasks ("Classification" Tasks)

1. Classify the entire document ("text categorization")
2. Classify word tokens individually
3. Classify word tokens in a sequence
4. Identify phrases ("chunking")
5. Syntactic annotation (parsing)
6. Semantic annotation

Slide courtesy Jason Eisner, with mild edits

Text Classification

Assigning subject categories, topics, or genres

Spam detection

Authorship identification

Age/gender identification

Language Identification

Sentiment analysis

…

Text Classification

Assigning subject categories, topics, or genres

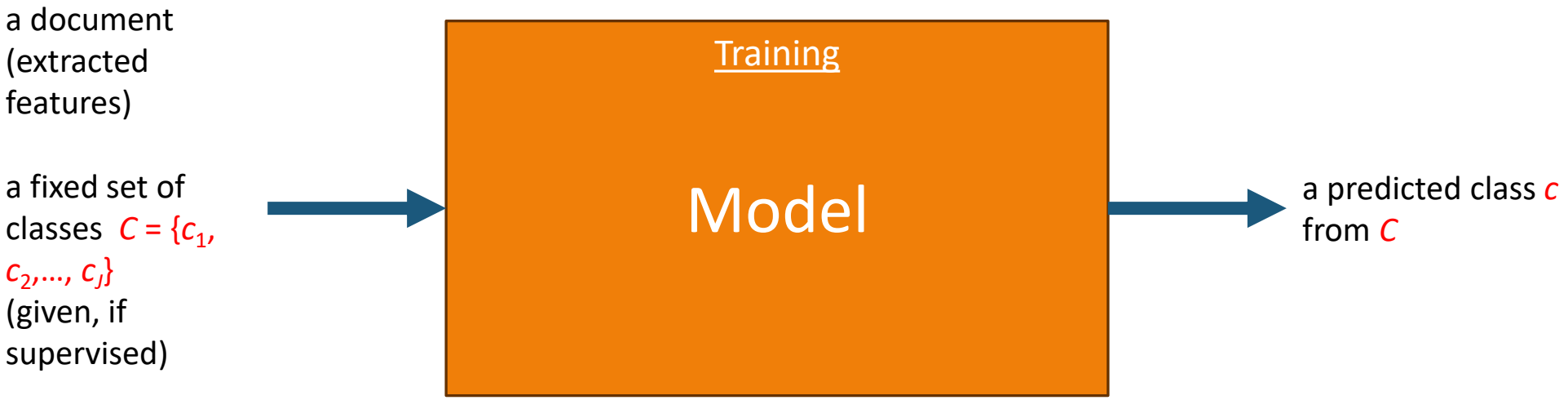
Spam detection

Authorship identification

Language Identification

Sentiment analysis

…



Text Classification

Assigning subject categories, topics, or genres

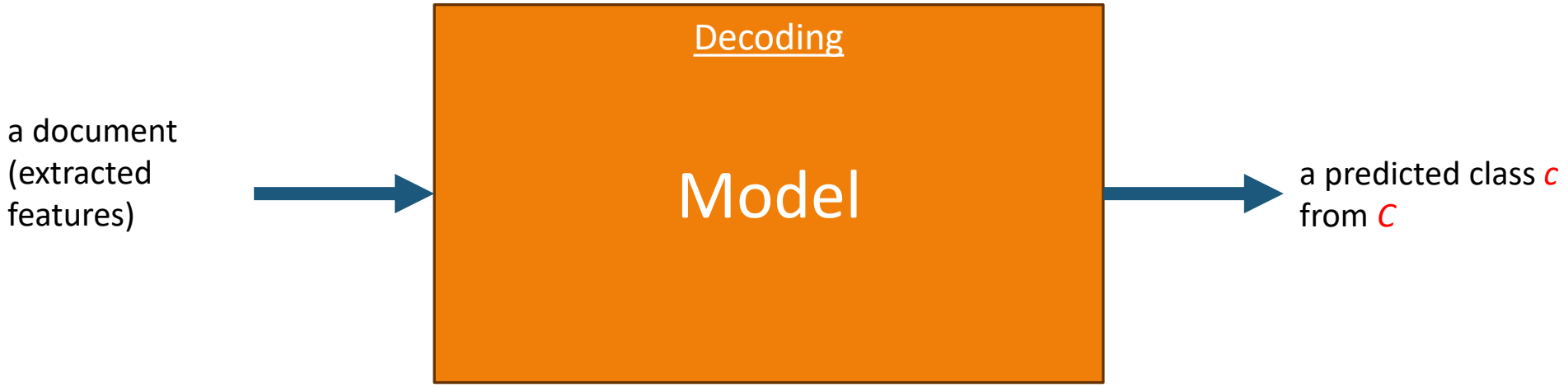
Spam detection

Authorship identification

Language Identification

Sentiment analysis

…



Text Classification: Hand-coded Rules?

Assigning subject categories, topics, or genres

Spam detection

Authorship identification

Age/gender identification

Language Identification

Sentiment analysis

…

Rules based on combinations of words or other features

spam: black-list-address OR (“dollars” AND “have been selected”)

Accuracy can be high

If rules carefully refined by expert

Building and maintaining these rules is expensive

Can humans faithfully assign uncertainty?

Text Classification: Supervised Machine Learning

Assigning subject categories, topics, or genres

Spam detection

Authorship identification

Age/gender identification

Language Identification

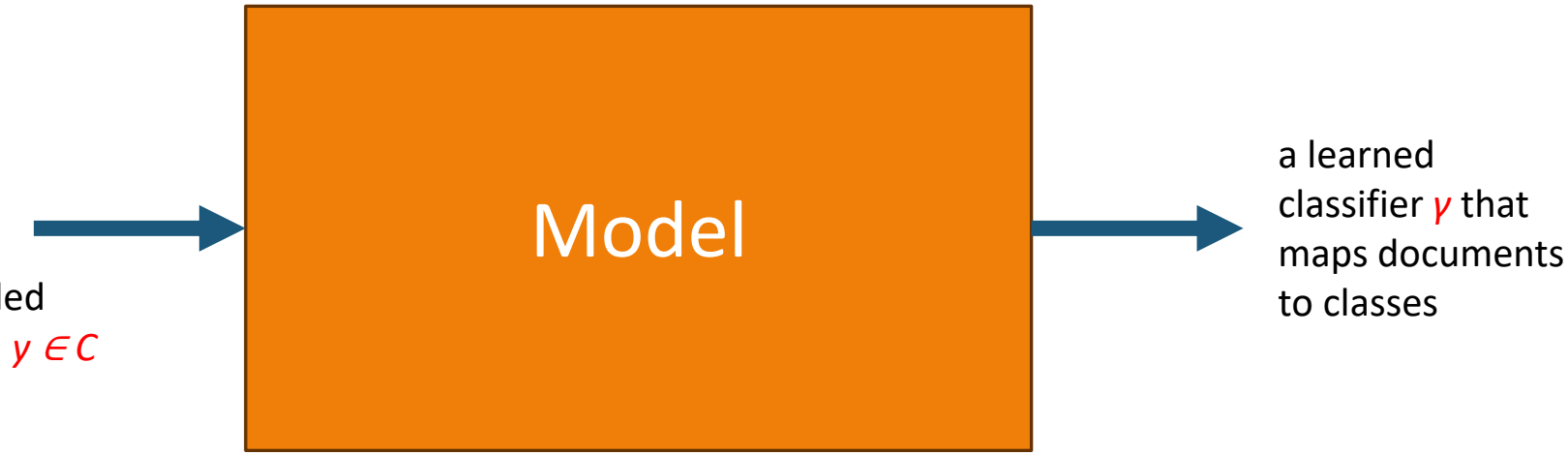
Sentiment analysis

…

a document *d*

a fixed set of classes
$C = \{c_1, c_2, \ldots, c_J\}$

a training set of *m* hand-labeled documents $(d_1,y_1),\ldots,(d_m,y_m)$, $y \in C$



a learned classifier *y* that maps documents to classes

Text Classification: Supervised Machine Learning

Assigning subject categories, topics, or genres

Spam detection

Authorship identification

Age/gender identification

Language Identification

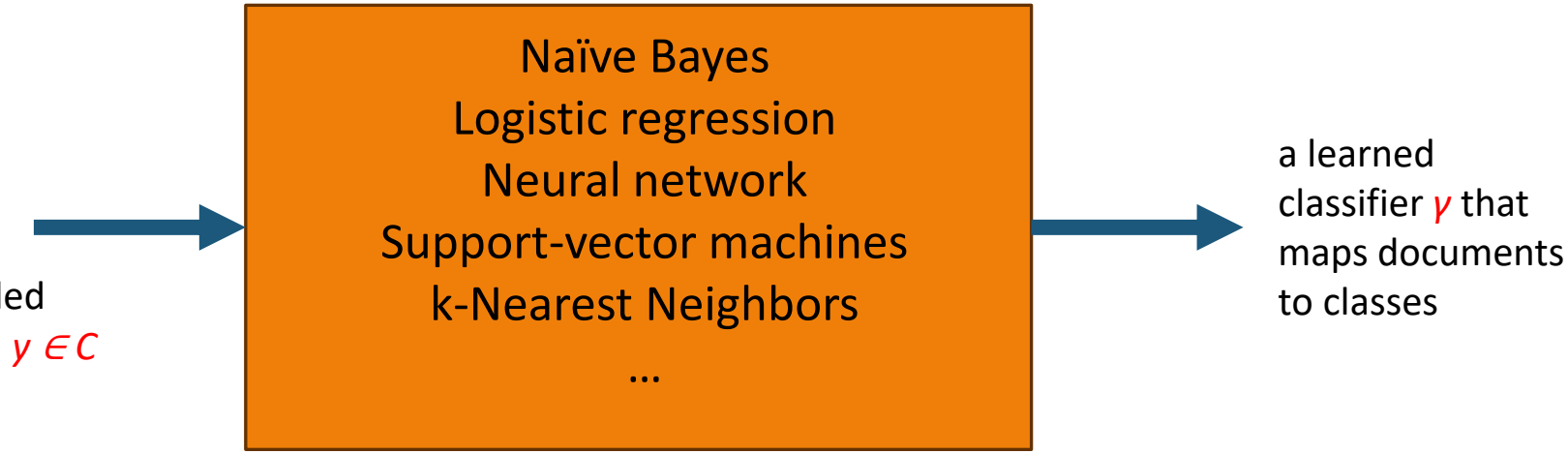
Sentiment analysis

…

a document *d*

a fixed set of classes
$C = \{c_1, c_2, \ldots, c_J\}$

a training set of *m* hand-labeled documents $(d_1,y_1),\ldots,(d_m,y_m)$, $y \in C$



a learned classifier *y* that maps documents to classes

Knowledge Check: Handling Types and Tokens

- 10 minutes to do it in class
- You can complete it after class
- Then submit it to Blackboard