CMSC 473/673 Introduction to Natural Language Processing

Instructor: Lara J. Martin (she/they)

TA: Omkar Kulkarni (he)

Learning Objectives

By the end of the course, you will be able to…

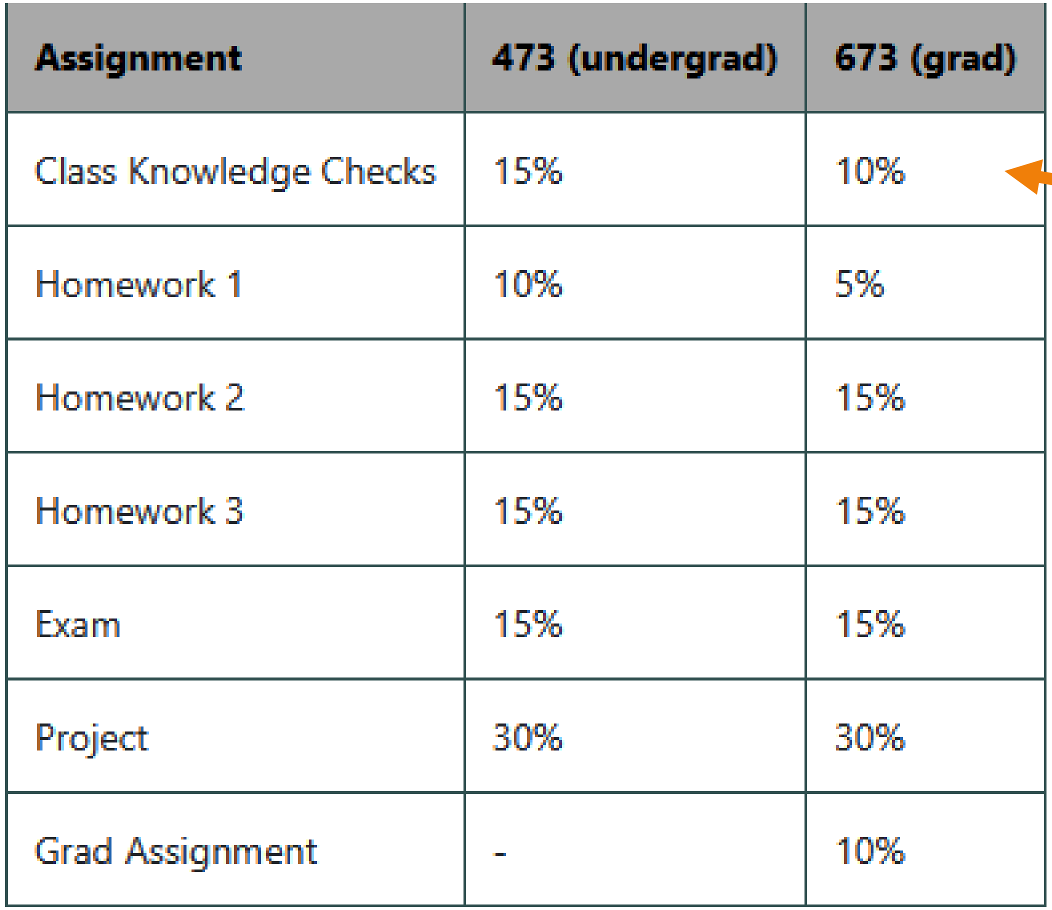
1. Recall common tasks in NLP and formulate problems for them. (HW1)
2. Diagnose and setup appropriate evaluation metrics for a given problem, including determining what an appropriate baseline might be. (HW2)
3. Compare and contrast language models and other NLP methods. (HW2, Exam)

Knowledge Checks

4. Implement AI systems that use popular NLP toolkits and libraries. (Grad Assignment, Project)
5. Construct a literature review from state-of-the-art research. (Grad Assignment, Project)
6. Plan and create an NLP system for a particular task. (HW3, Project)
7. Identify ethical issues in NLP systems and consider how they might be mitigated. (HW3)

Grades

Late days!
Everyone has 5 free late days that they can use for homework assignments or project milestones (except the last milestone)
Max use of 3 per assignment

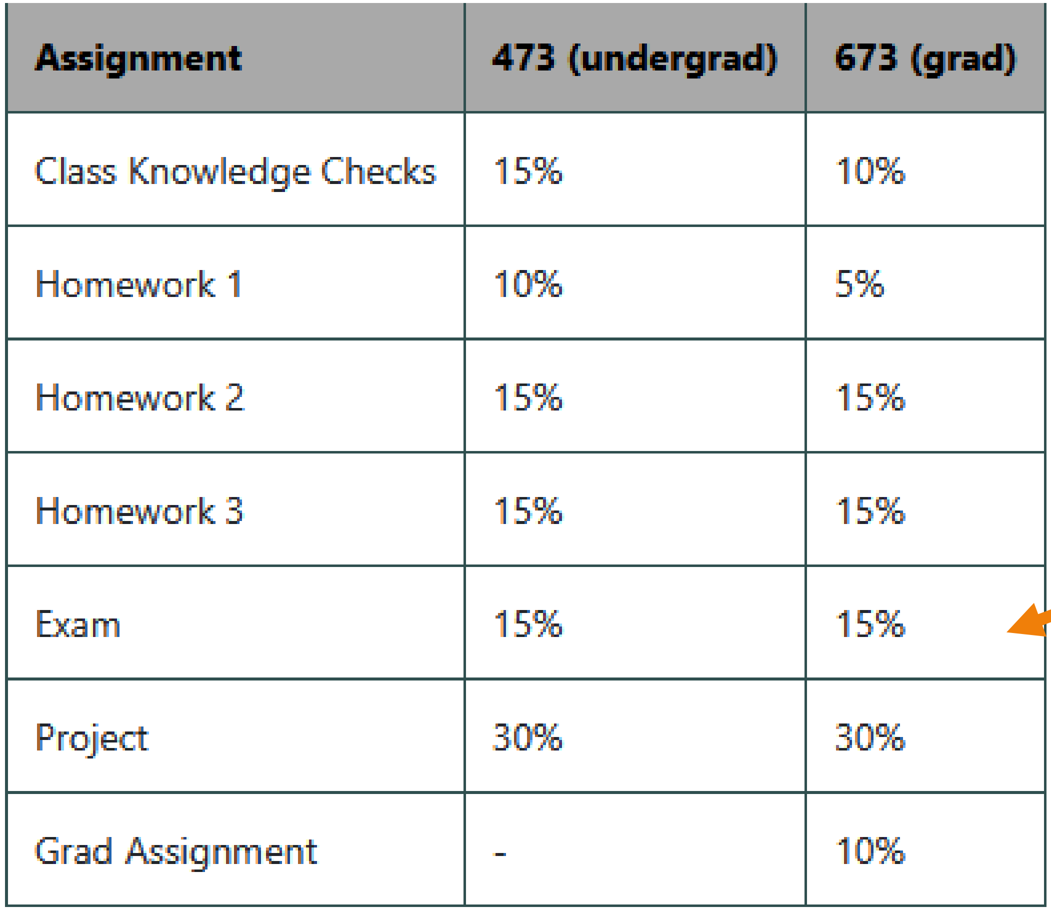


Assignment	473 (undergrad)	673 (grad)
Class Knowledge Checks	15%	10%
Homework 1	10%	5%
Homework 2	15%	15%
Homework 3	15%	15%
Exam	15%	15%
Project	30%	30%
Grad Assignment	-	10%

- In-class checks so that I can see how you're doing with the material
- Not graded for accuracy
- Can be made up by the end of the semester

- 3 homework assignments
- NLP tasks, evaluation & neural networks, prompt engineering & NLP ethics
- First homework is worth less than the other two
- Can be worked on alone or in pairs

Grades



Assignment	473 (undergrad)	673 (grad)
Class Knowledge Checks	15%	10%
Homework 1	10%	5%
Homework 2	15%	15%
Homework 3	15%	15%
Exam	15%	15%
Project	30%	30%
Grad Assignment	-	10%

- New for this semester
- I want to test your knowledge of NLP concepts

- Group project (around 3-5 people)
- You will come up with your own topic with my help

- Implementation or literature review

Academic Integrity

- If you feel the need to cheat on the assignment to do well on it, please talk to me or Omkar first. We can work it out ahead of time, but once you cheat it’s hard to do anything.

If you cheat or plagiarize, you…

- aren’t learning anything
- wasting money paying for tuition
- can get an F on the assignment or even for the entire class

More details on course website

If you want to use ChatGPT

- Make sure you're saying that you used it
- Provide your prompt and the original generation (along with how you edited it)
- Make sure that you're not avoiding the learning objectives by using it

- If you do not say you're using it and I notice, that is an academic integrity violation
- It's okay to use grammar tools (e.g., spell check or Grammarly) or small-scale prediction (e.g., next word prediction, tab completion), provided that they don't change the **substance** of your work

Learning Objectives

Develop a working vocabulary of terms in the field of NLP

Recognize NLP systems in your daily life

Define sub areas of linguistics

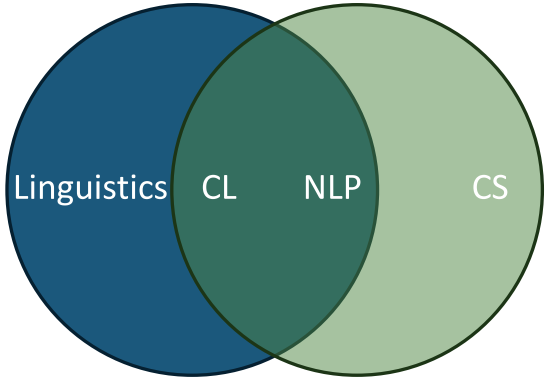
Distinguish between types and tokens

Define featurization & other ML terminology

Define some “classification” terminology

Distinguish between different text classification tasks

Computational Linguistics
=?
Natural Language Processing



The computational **study** of language

Computational Linguistics

≈

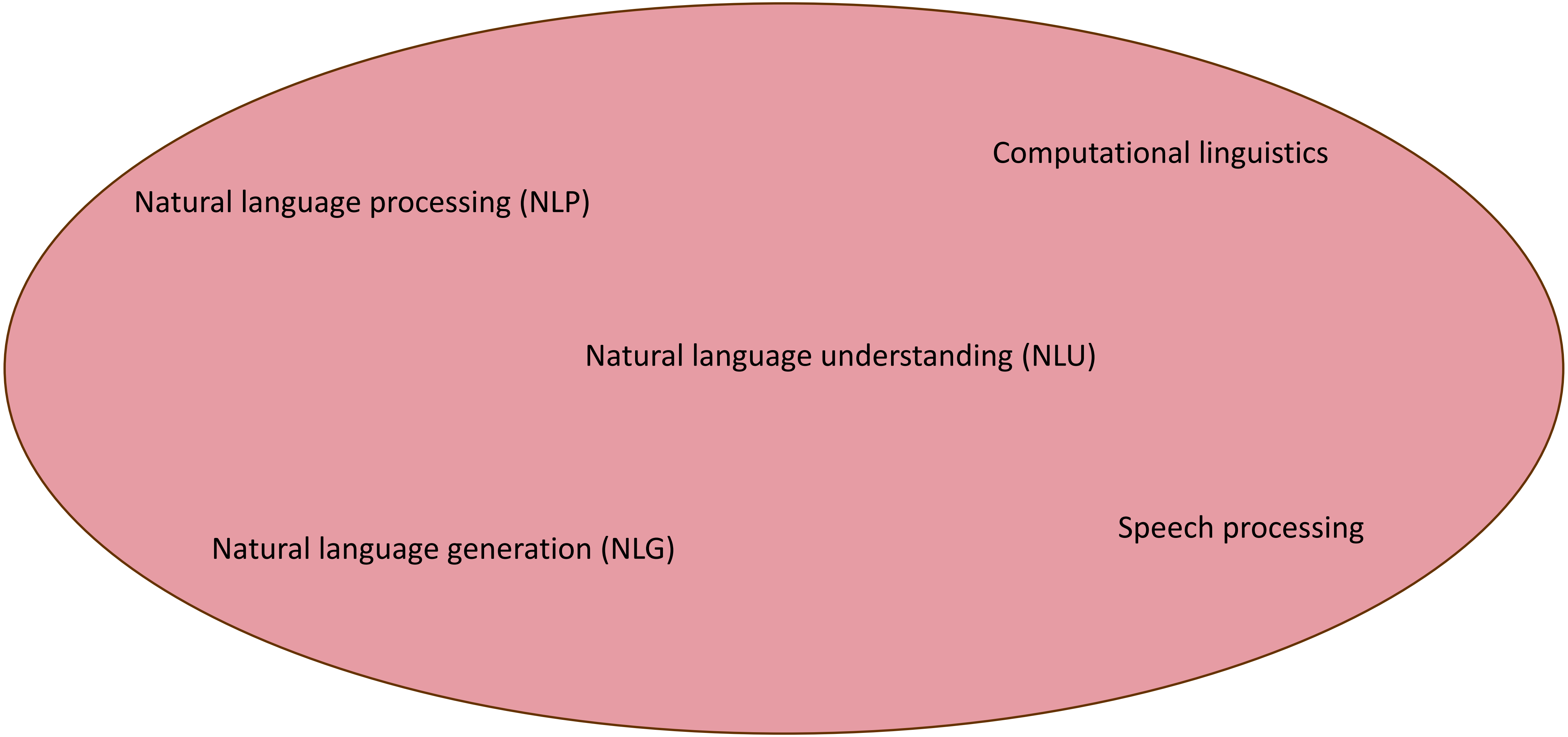
Natural Language Processing

The computational **use** of language

Association for Computational Linguistics

😅

Language technologies



Computational linguistics

Natural language processing (NLP)

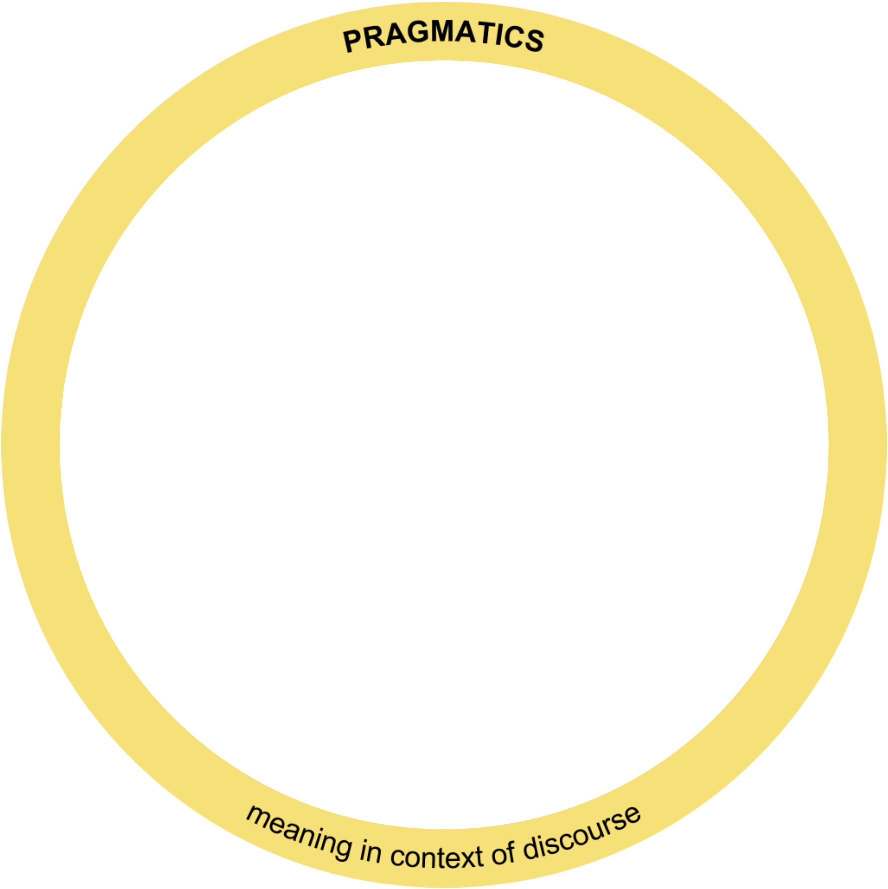
Natural language understanding (NLU)

Speech processing

Natural language generation (NLG)

Linguistics

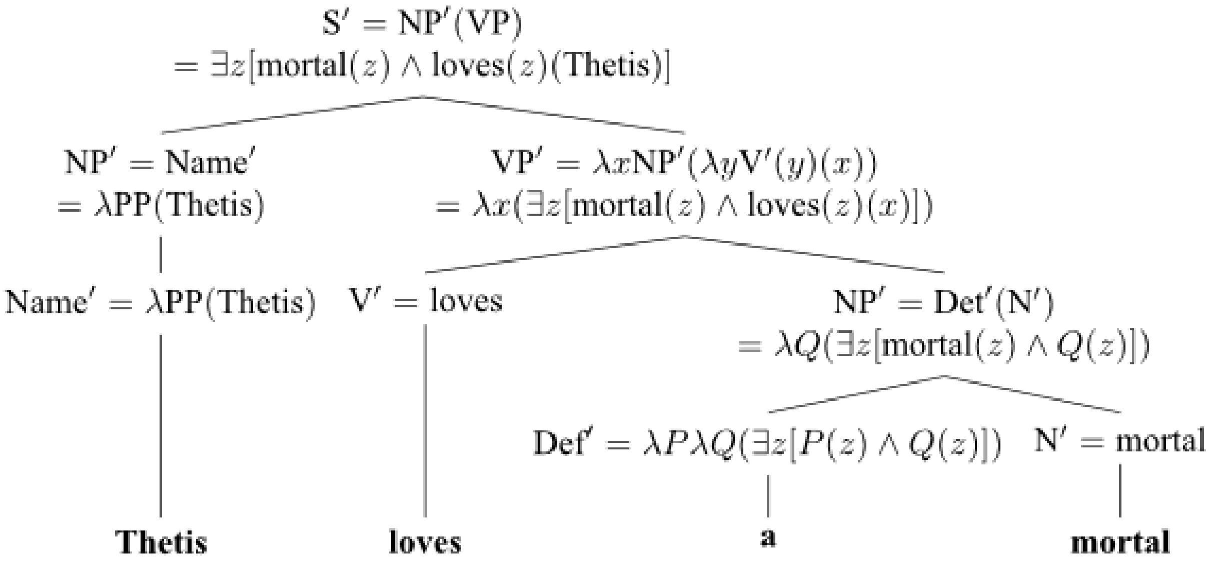
The study of language



https://en.wikipedia.org/wiki/Morphology_(linguistics)#/media/File:Major_levels_of_linguistic_structure.svg

Semantics

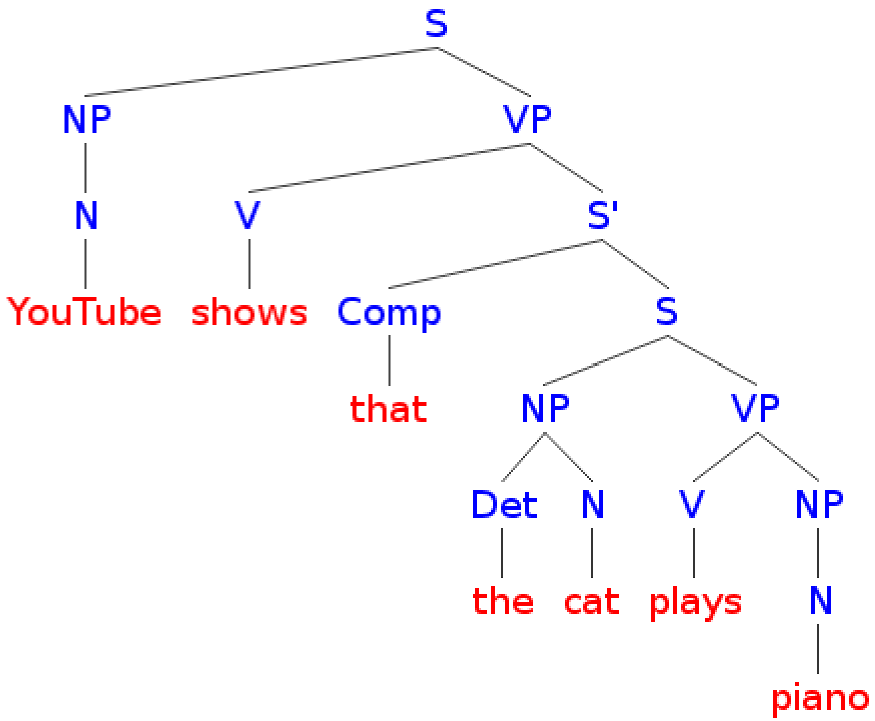
Meaning


$\mathrm{S}' = \mathrm{NP}'(\mathrm{VP})$
$= \exists z[\mathrm{mortal}(z) \wedge \mathrm{loves}(z)(\mathrm{Thetis})]$
$\mathrm{NP}' = \mathrm{Name}'$
$= \lambda \mathrm{PP}(\mathrm{Thetis})$
$\mathrm{VP}' = \lambda x \mathrm{NP}'(\lambda y \mathrm{V}'(y)(x))$
$= \lambda x(\exists z[\mathrm{mortal}(z) \wedge \mathrm{loves}(z)(x)])$
$\mathrm{Name}' = \lambda \mathrm{PP}(\mathrm{Thetis})$
$\mathrm{V}' = \mathrm{loves}$
$\mathrm{NP}' = \mathrm{Det}'(\mathrm{N}')$
$= \lambda Q(\exists z[\mathrm{mortal}(z) \wedge Q(z)])$
$\mathrm{Def}' = \lambda P \lambda Q(\exists z[P(z) \wedge Q(z)])$
$\mathrm{N}' = \mathrm{mortal}$
Thetis
loves
a
mortal

https://plato.stanford.edu/entries/computational-linguistics/

Syntax

Grammar



https://allthingslinguistic.com/post/100617668093/how-to-draw-syntax-trees-part-3-type-1-a

Phonology

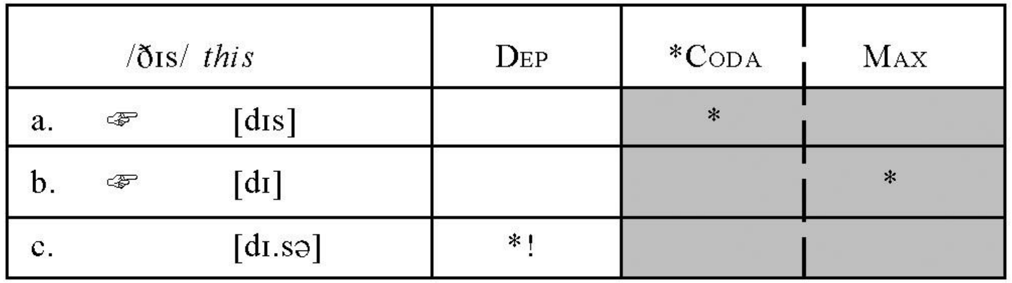
Processing of sounds

https://upload.wikimedia.org/wikipedia/commons/a/a5/Tsunami_by_hokusai_19th_century.jpg

tsunami

↓

sunami

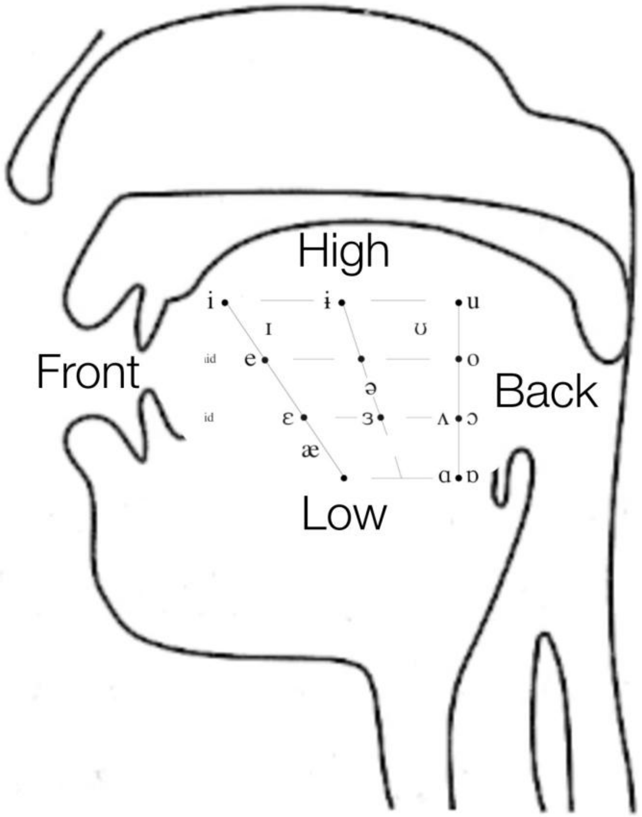


/ðɪs/ *this*			Dep	*Coda	Max
a.	☞	[dɪs]		*	
b.	☞	[dɪ]			*
c.		[dɪ.sə]	*!		

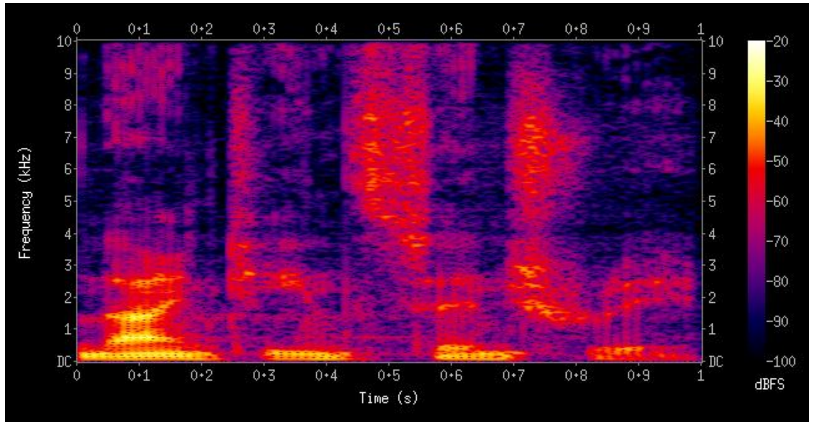
https://pubs.asha.org/doi/10.1044/0161-1461%282001/022%29

Phonetics

Physical production/understanding of sounds



https://wstyler.ucsd.edu/talks/l111_3_phonetics_review_handout.html



https://en.wikipedia.org/wiki/Spectrogram#/media/File:Spectrogram-19thC.png

Back to CL vs NLP

Computational linguistics: Using computers to solve linguistic questions

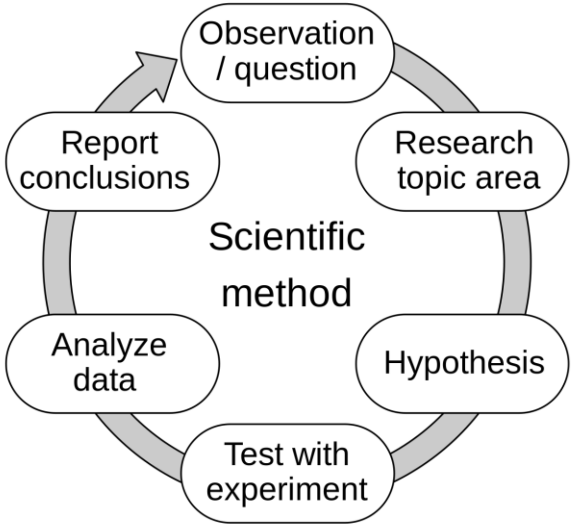
- E.g., How does language X order their sentences? SVO, SOV, VOS…?

And this can inform NLP work

- E.g., How can we create a system that generates text in language X?

Or not…

- E.g., Let’s feed a model a bunch of text so that it can generate text in language X.



How do we solve any of these problems?

Data!

https://upload.wikimedia.org/wikipedia/commons/thumb/8/82/The_Scientific_Method.svg/1200px-The_Scientific_Method.svg.png

Where does the data come from?

Corpus (plural: corpora)

- Literally a “body” of text

Languages with few corpora are called “low-resource languages”

- This might not mean the language is endangered!

We can collect corpora in a few different ways:

- Curation: data tagged & organized by experts
- Internet: data “scraped” from open-access sources (Wikipedia, Reddit)
  - Or data collected with permission from closed sources (Facebook, texts) – more rare
- Elicitation: carefully getting participants to produce language (lab studies, crowdsourcing, field studies)
- Pre-existing corpora

! Facebook has gotten into trouble several times for using data or manipulating people’s feeds without their permission

Benchmarking

Collecting & publishing corpora is helpful for…

- Replication
- Improving performance

Your task

Benchmarking

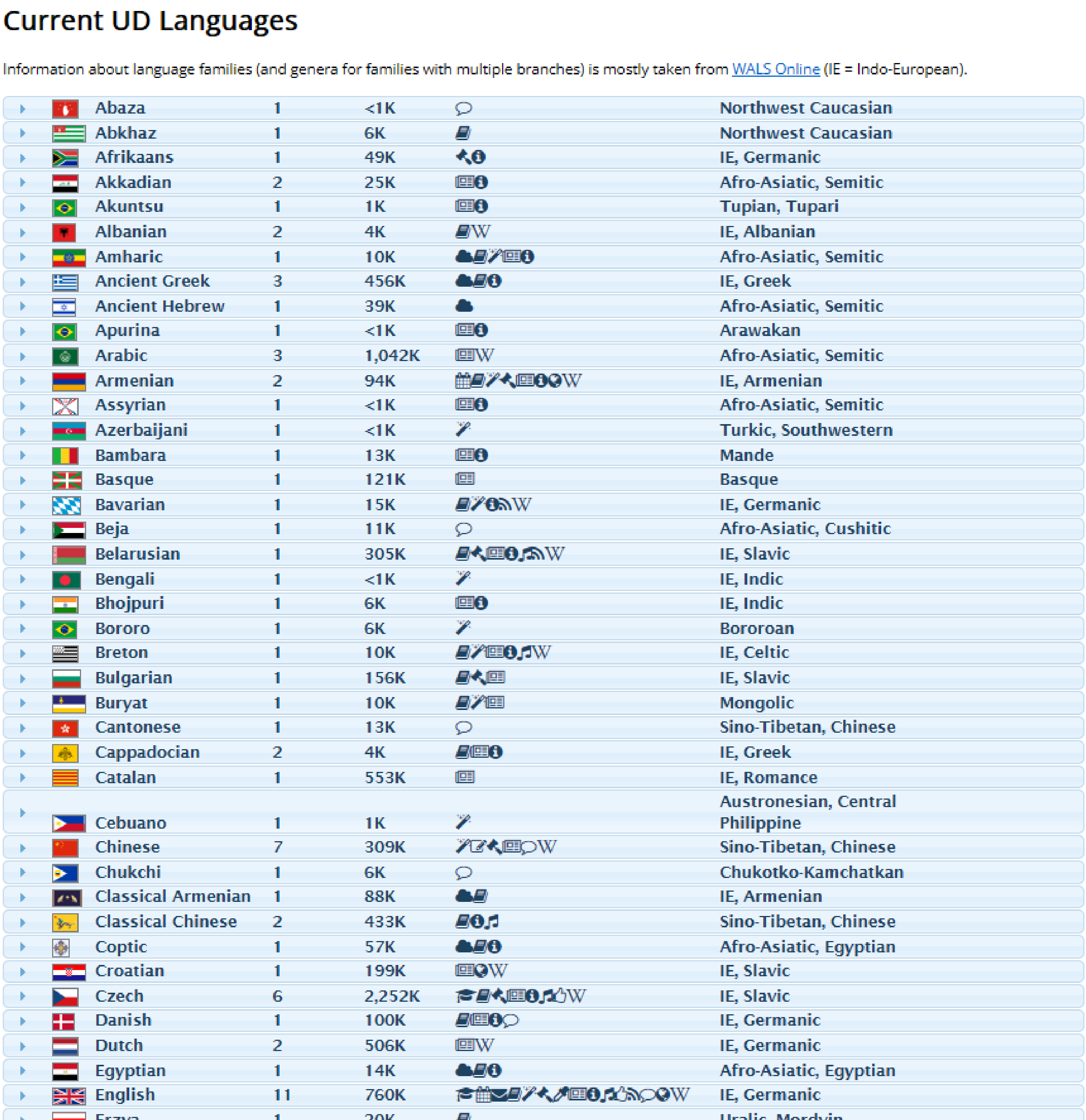
If you want people to work on your problem, make it easy for them to get started and to measure their progress. Provide:

- Test data, for evaluating the final systems
- Development data, for measuring whether a change to the system helps, and for tuning parameters
- An evaluation metric (formula for measuring how well a system does on the dev or test data)
- A program for computing the evaluation metric
- Labeled training data and other data resources
- A prize? – with clear rules on what data can be used

What does the data look like?

Curated data (and some collected data) are usually labeled, especially when made for a particular **task**

- E.g., Universal dependencies (https://universaldependencies.org/)



Current UD Languages

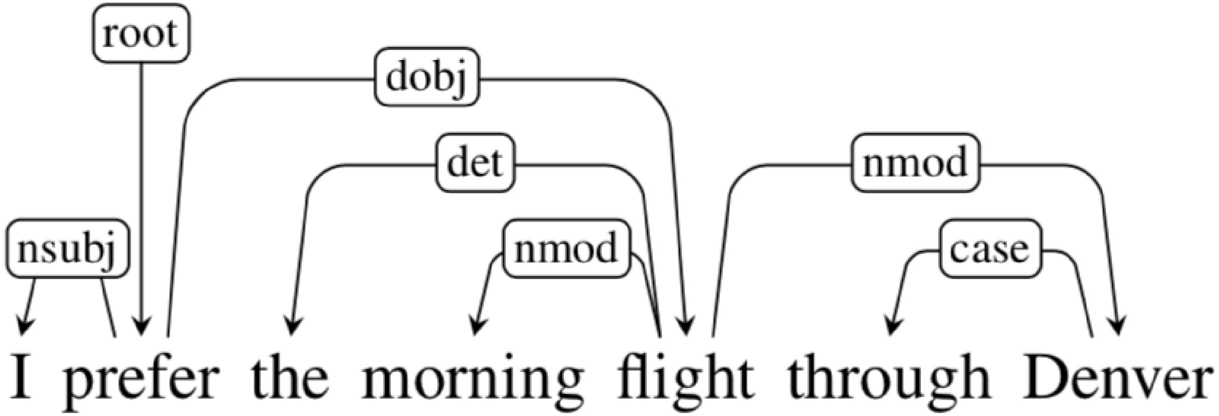
Information about language families (and genera for families with multiple branches) is mostly taken from WALS Online (IE = Indo-European).

Abaza	1	<1K	💬	Northwest Caucasian
Abkhaz	1	6K	📖	Northwest Caucasian
Afrikaans	1	49K	🔨ℹ	IE, Germanic
Akkadian	2	25K	📰ℹ	Afro-Asiatic, Semitic
Akuntsu	1	1K	📰ℹ	Tupian, Tupari
Albanian	2	4K	📖W	IE, Albanian
Amharic	1	10K	☁📖🪄📰ℹ	Afro-Asiatic, Semitic
Ancient Greek	3	456K	☁📖ℹ	IE, Greek
Ancient Hebrew	1	39K	☁	Afro-Asiatic, Semitic
Apurina	1	<1K	📰ℹ	Arawakan
Arabic	3	1,042K	📰W	Afro-Asiatic, Semitic
Armenian	2	94K	📅📖🪄🔨📰ℹ🌐W	IE, Armenian
Assyrian	1	<1K	📰ℹ	Afro-Asiatic, Semitic
Azerbaijani	1	<1K	🪄	Turkic, Southwestern
Bambara	1	13K	📰ℹ	Mande
Basque	1	121K	📰	Basque
Bavarian	1	15K	📖🪄ℹ📶W	IE, Germanic
Beja	1	11K	💬	Afro-Asiatic, Cushitic
Belarusian	1	305K	📖🔨📰ℹ♫📶W	IE, Slavic
Bengali	1	<1K	🪄	IE, Indic
Bhojpuri	1	6K	📰ℹ	IE, Indic
Bororo	1	6K	🪄	Bororoan
Breton	1	10K	📖🪄📰ℹ♫W	IE, Celtic
Bulgarian	1	156K	📖🔨📰	IE, Slavic
Buryat	1	10K	📖🪄📰	Mongolic
Cantonese	1	13K	💬	Sino-Tibetan, Chinese
Cappadocian	2	4K	📖📰ℹ	IE, Greek
Catalan	1	553K	📰	IE, Romance
Cebuano	1	1K	🪄	Austronesian, Central Philippine
Chinese	7	309K	🪄📝🔨📰💬W	Sino-Tibetan, Chinese
Chukchi	1	6K	💬	Chukotko-Kamchatkan
Classical Armenian	1	88K	☁📖	IE, Armenian
Classical Chinese	2	433K	📖ℹ♫	Sino-Tibetan, Chinese
Coptic	1	57K	☁📖ℹ	Afro-Asiatic, Egyptian
Croatian	1	199K	📰🌐W	IE, Slavic
Czech	6	2,252K	🎓📖🔨📰ℹ♫👍W	IE, Slavic
Danish	1	100K	📖📰ℹ💬	IE, Germanic
Dutch	2	506K	📰W	IE, Germanic
Egyptian	1	14K	☁📖ℹ	Afro-Asiatic, Egyptian
English	11	760K	🎓📅✉📖🪄🔨💉📰ℹ♫👍📶💬🌐W	IE, Germanic
Erzya	1	20K	📖	Uralic, Mordvin

What does the data look like?

Curated data (and some collected data) are usually labeled, especially when made for a particular **task**

- E.g., Universal dependencies (https://universaldependencies.org/)



https://medium.com/data-science-in-your-pocket/dependency-parsing-associated-algorithms-in-nlp-96d65dd95d3e

Modalities

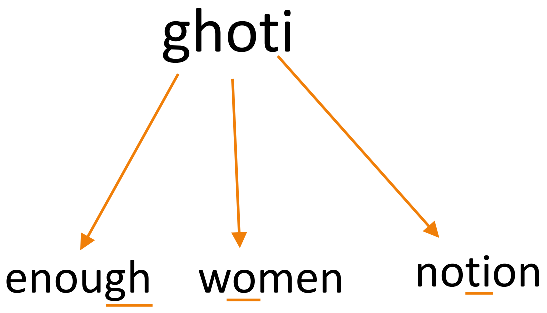
Text

TTS isn’t straight forward. Unless you have information on how text is pronounced, an orthography (a writing system) by itself can be misleading.

Audio (speech)

Video (closed captioning, sign languages)

Pictures (handwriting recognition, image captioning)


ghoti

enough

women

notion

Any of these can be labeled

What’s in a word?

https://www.freepngimg.com/download/bat/9-2-bat-png-hd.png

bat

What’s in a word?

bats

https://www.freepngimg.com/download/bat/9-2-bat-png-hd.png

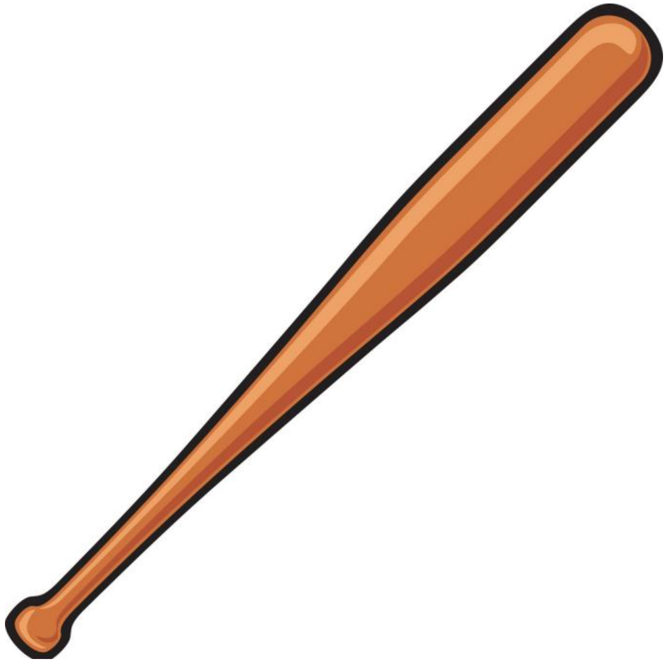
What’s in a word?

Fledermaus

flutter mouse

https://www.freepngimg.com/download/bat/9-2-bat-png-hd.png

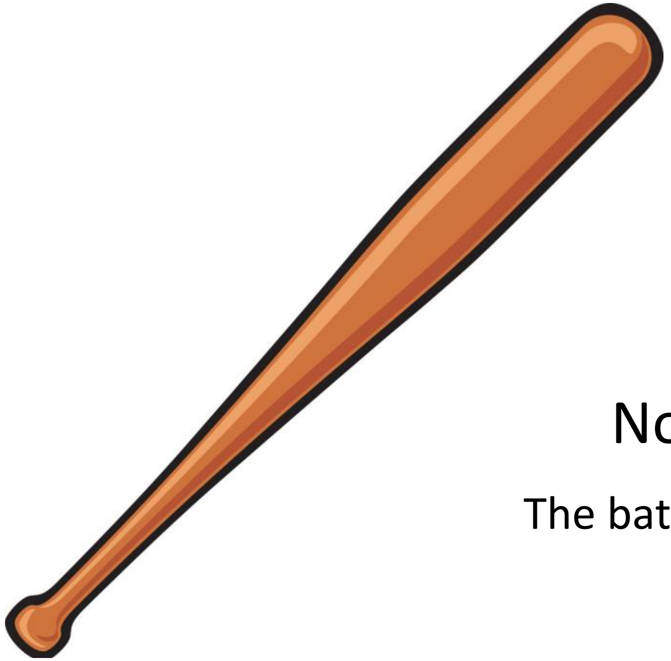
What’s in a word?



bat

https://www.vectorstock.com/royalty-free-vector/baseball-bat-vector-1448799

What’s in a word?



bat

Noun?

The bat was heavy.

Verb?

They bat 1000.

https://www.vectorstock.com/royalty-free-vector/baseball-bat-vector-1448799

What’s in a word?

):

What’s in a word?

my leg is hurting nasty):

What’s in a word?

add two cups (a pint): bring to a boil

Tokens vs Types

The film got a great opening and the film went on to become a hit .

Vocabulary: the words (items) you know

Type: an element of the vocabulary.

Token: an instance of that type in running text.

How many of types & tokens appear in the above sentence?

Tokens vs Types

Types

- The
- film
- got
- a
- great
- opening
- and
- the
- went
- on
- to
- become
- hit
- .

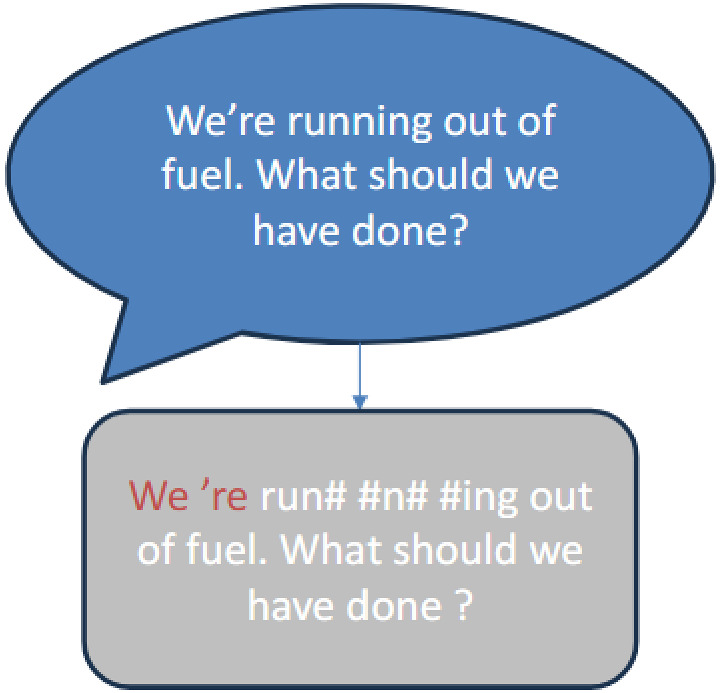
Tokens

- The
- film
- got
- a
- great
- opening
- and
- the
- ~~film~~
- went
- on
- to
- become
- ~~a~~
- hit
- .

For your {task} how do you define tokens?

Sometimes:

1. They’re defined for you by the *dataset creator*



We’re running out of fuel. What should we have done?

We ’re run# #n# #ing out of fuel. What should we have done ?

why?

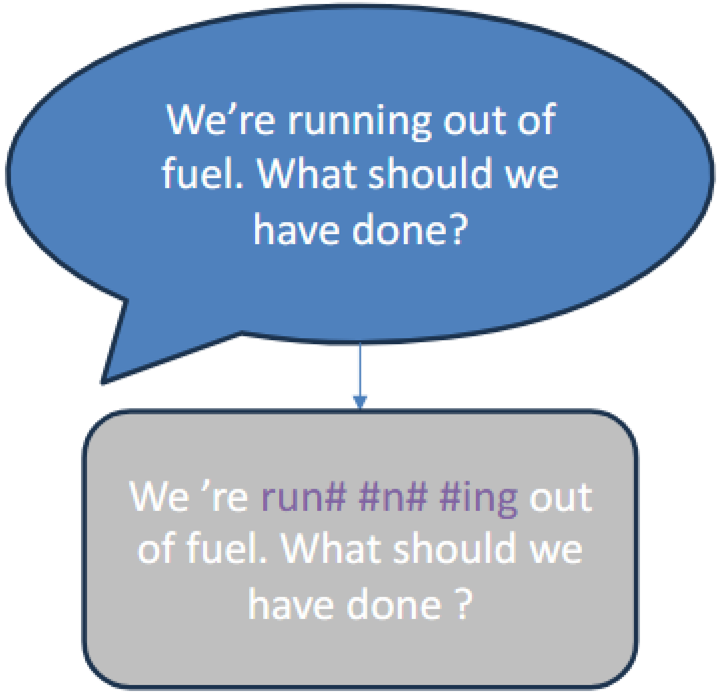
- *scaleably handling novel words*
- *linguistic reasons*
- *historical reasons / technical debt*

What usually happens when you input a word that your writing/texting program doesn’t recognize?

For your {task} how do you define tokens?

Sometimes:

1. They’re defined for you by the *dataset creator*
2. They’re defined by the *model*



We’re running out of fuel. What should we have done?

We ’re run# #n# #ing out of fuel. What should we have done ?

(why? scaleably handling novel words)

For your {task} how do you define tokens?

Sometimes:

1. They’re defined for you by the *dataset creator*
2. They’re defined by the *model*
3. It might be part of the *research problem itself*

pişirdiler
They cooked it.

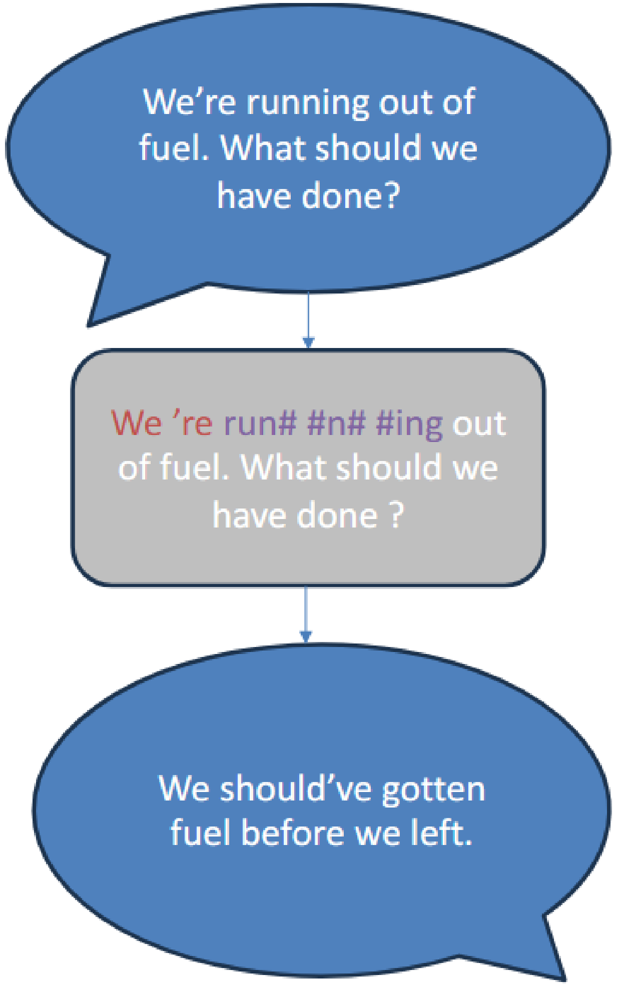
vs.

pişmişlermişlerdi
They had it cooked it.

For your {task} how do you define tokens?

Sometimes:

1. They’re defined for you by the *dataset creator*
2. They’re defined by the *model*
3. It might be part of the *research problem itself*
4. They’re defined by the *end user*
    1. You’ll need to handle points 1 and/or 2 on-the-backend…
    2. and then reversing the process to present output to the user



We’re running out of fuel. What should we have done?

We ’re run# #n# #ing out of fuel. What should we have done ?

We should’ve gotten fuel before we left.

Knowledge Check

When poll is active respond at

PollEv.com/laramartin527

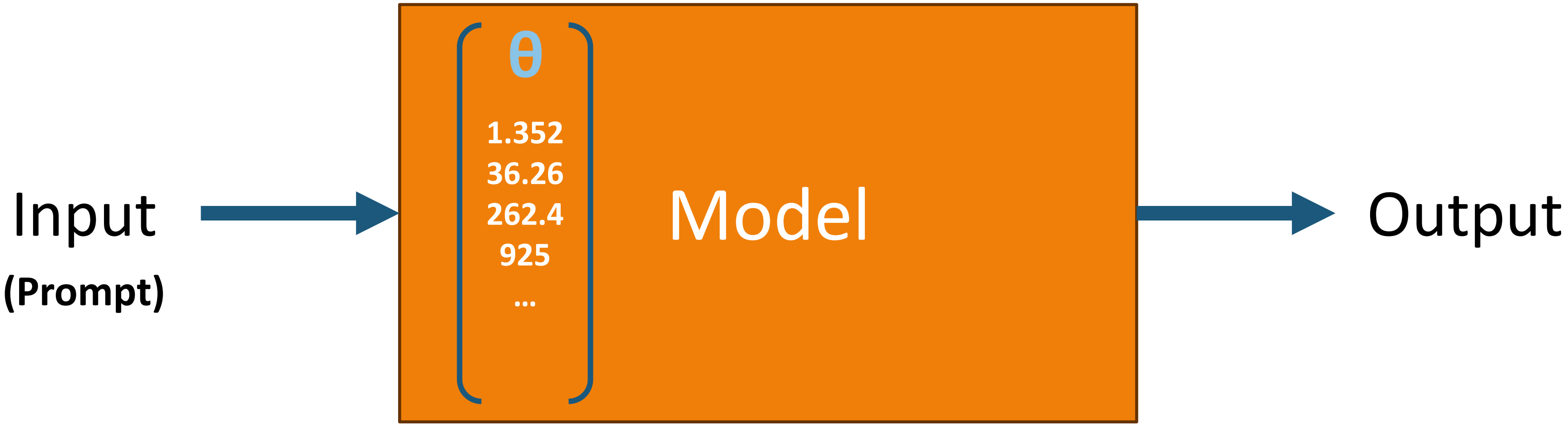
or

Send laramartin527 and your message to 22333

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

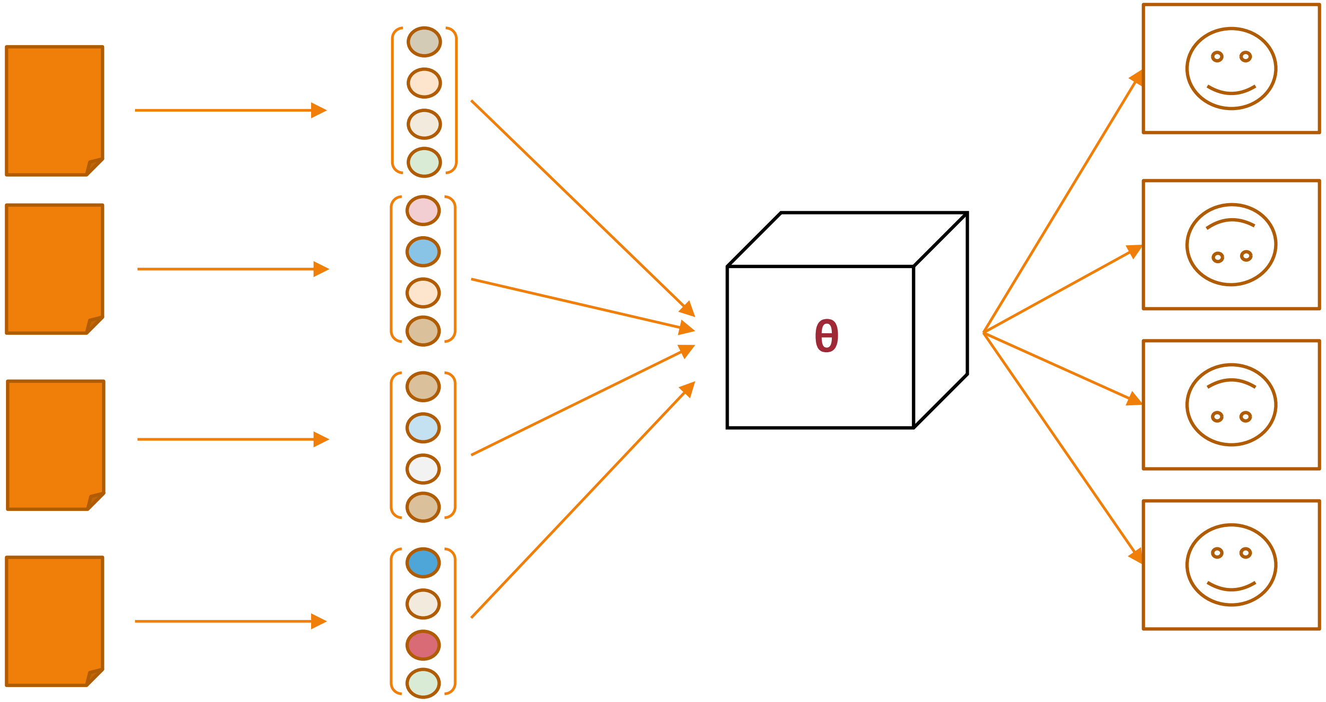
instances

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

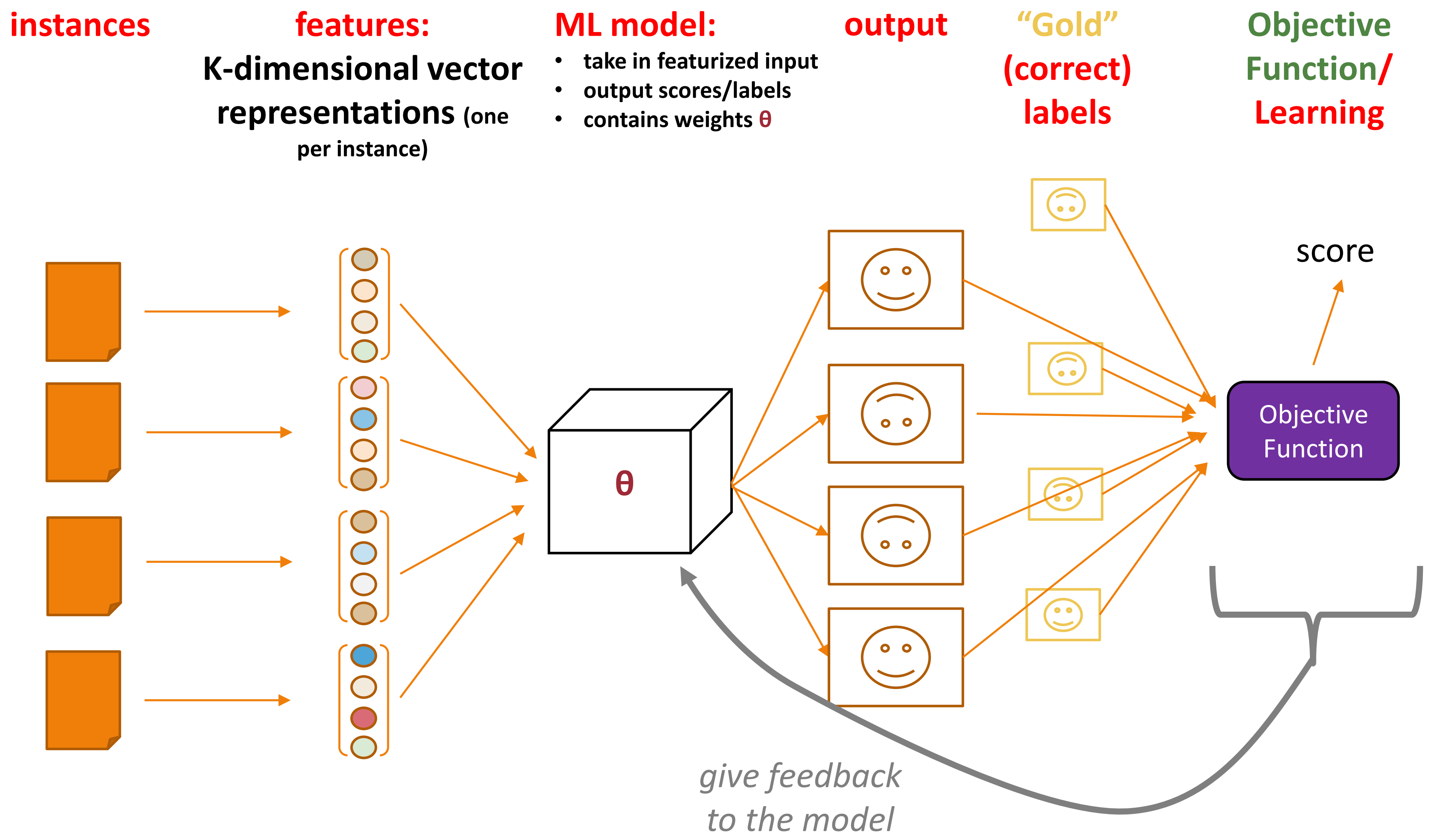
Training / Learning:

- 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

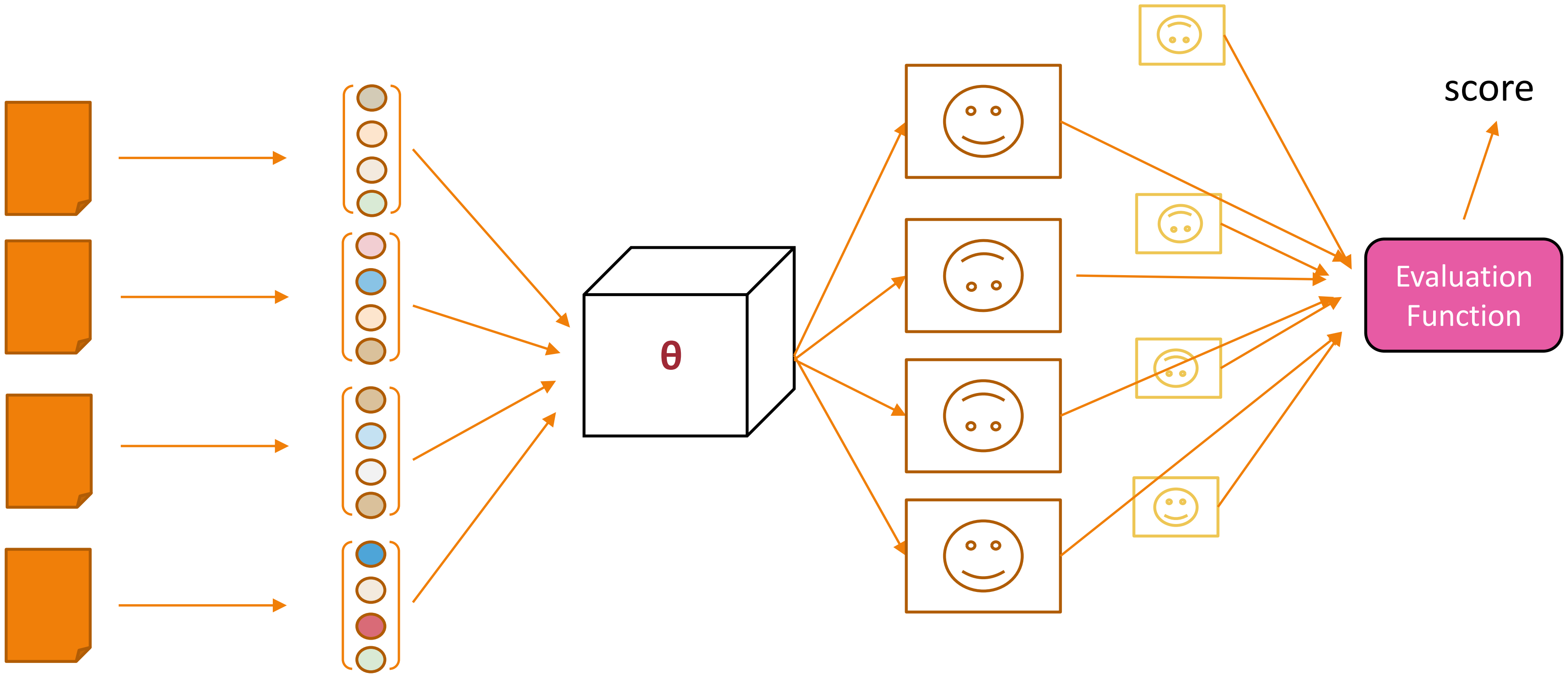
features:
K-dimensional vector representations (one per instance)

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)

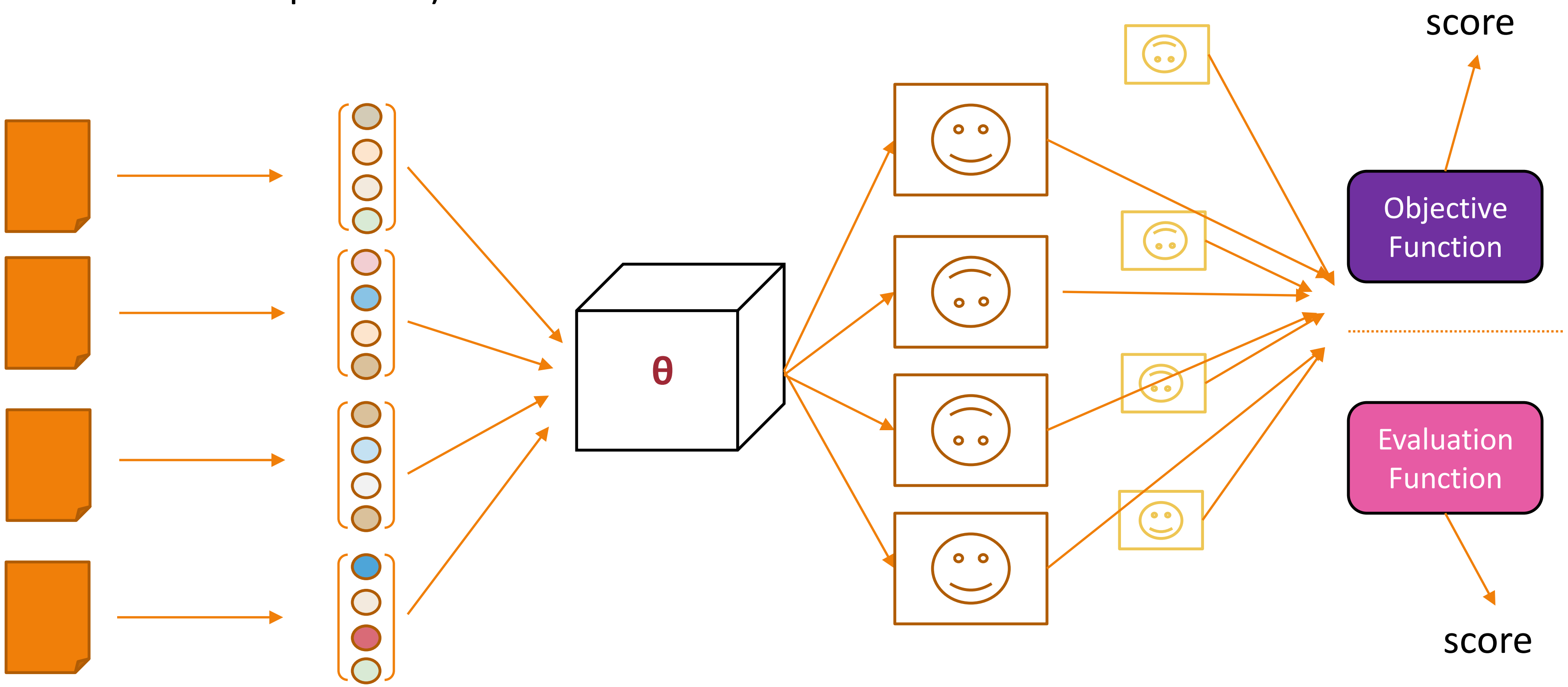
ML model:

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

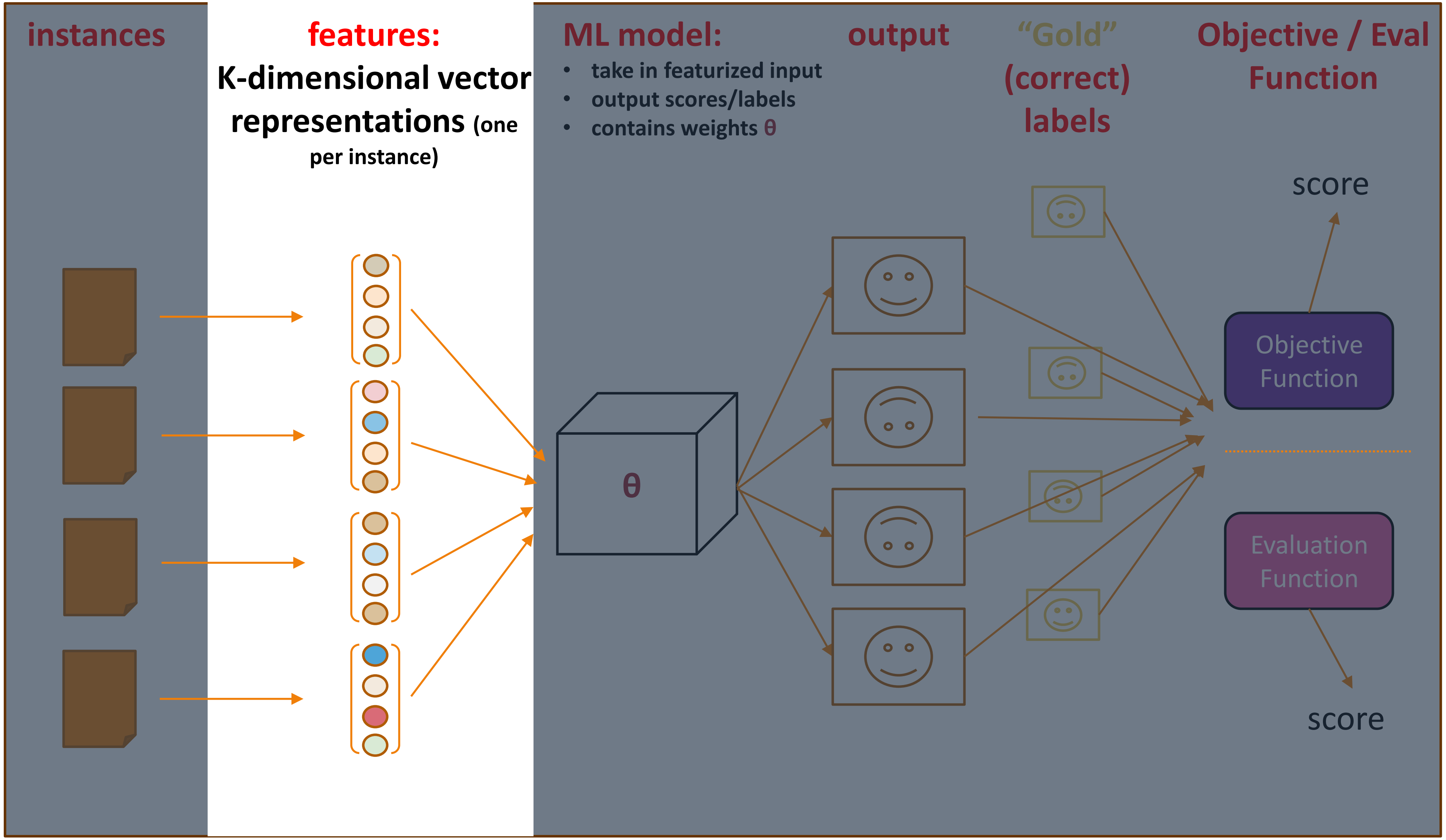
output

"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*

$$f(x) = (\underbrace{f_1(x)}, \dots, \underbrace{f_K(x)})$$

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

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

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.

TECH

NOT TECH

feature extraction

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

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

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

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

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

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

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

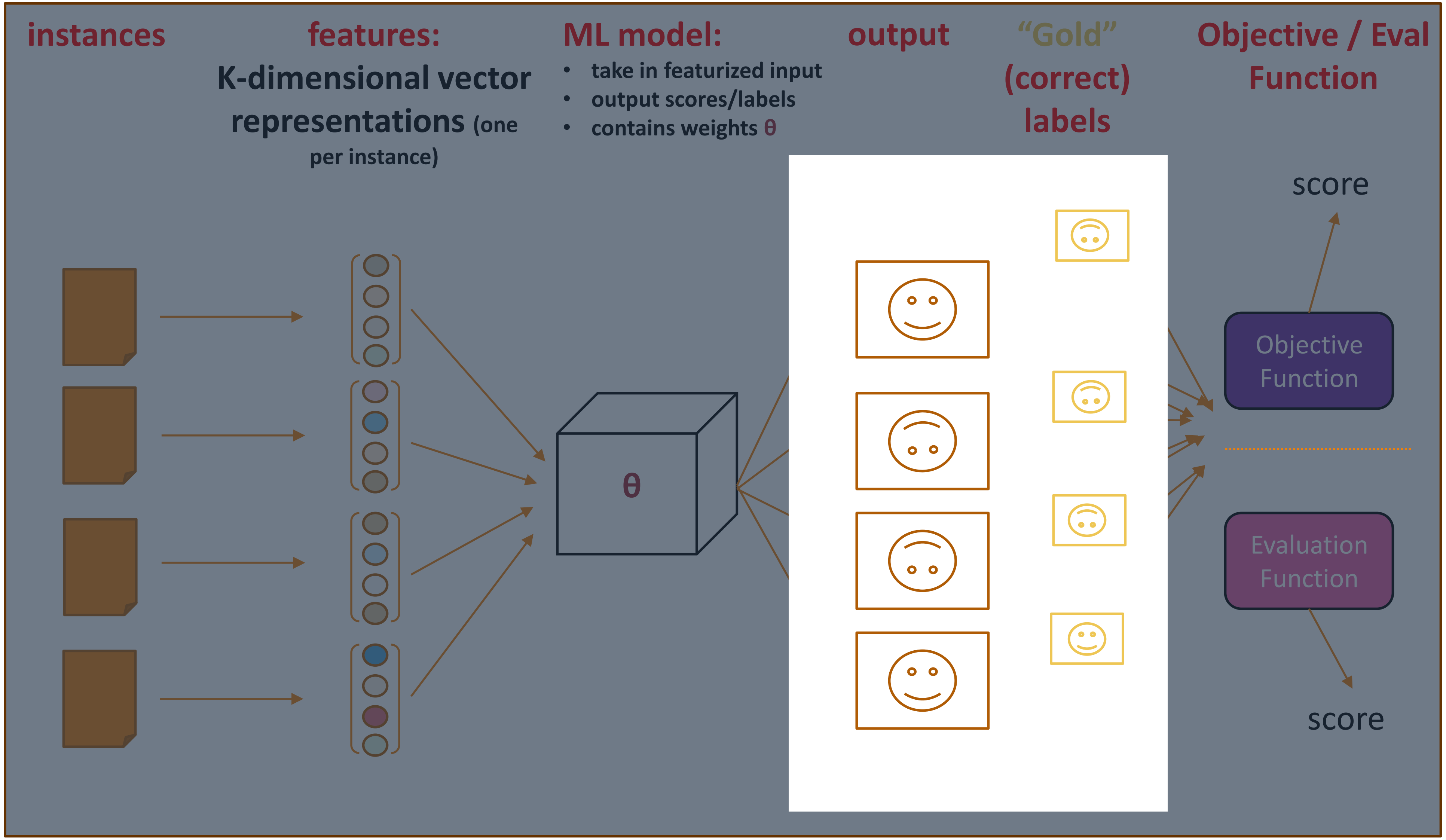
1. Bag-of-words (or bag-of-characters, bag-of-relations)
    - Identify ***unique*** sufficient atomic sub-parts (e.g., words in a document)
    - Define simple features over these, e.g.,
        - Binary (0 or 1) ➔ indicating presence
        - Natural numbers ➔ indicating number of times in a context
        - 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

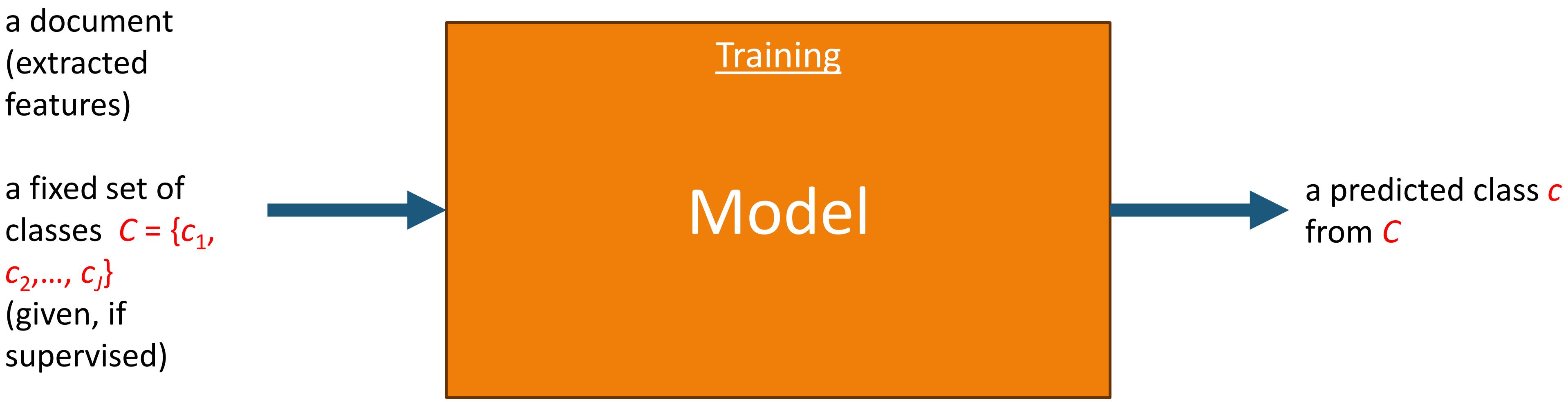
Spam detection

Authorship identification

Language Identification

Sentiment analysis

…



a document (extracted features)

a fixed set of classes $C = \{c_1, c_2, \ldots, c_J\}$ (given, if supervised)

a predicted class c from C

Text Classification

Assigning subject categories, topics, or genres

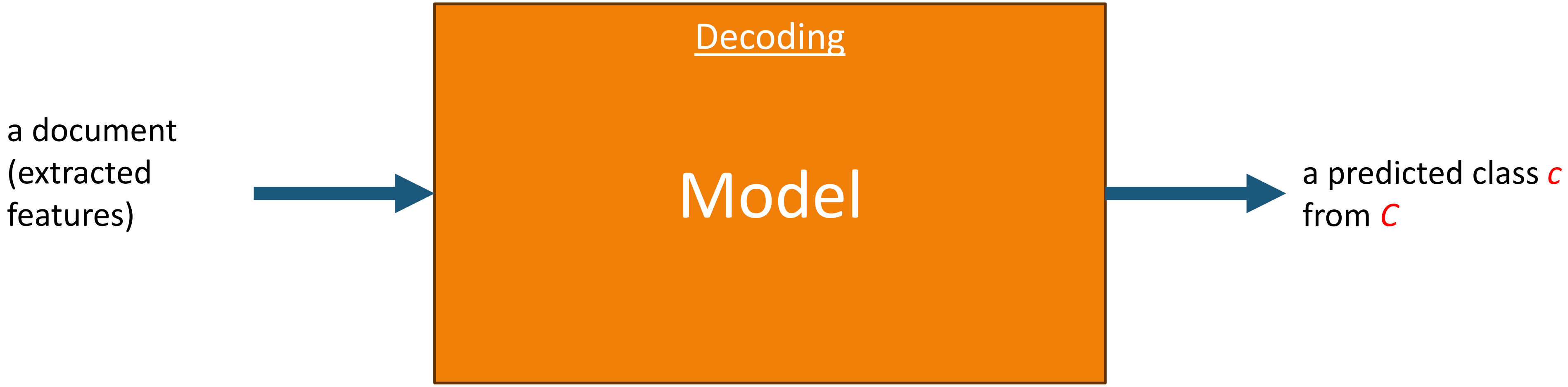
Spam detection

Authorship identification

Language Identification

Sentiment analysis

…



a document (extracted features)
Decoding
Model
a predicted class *c* from *C*

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

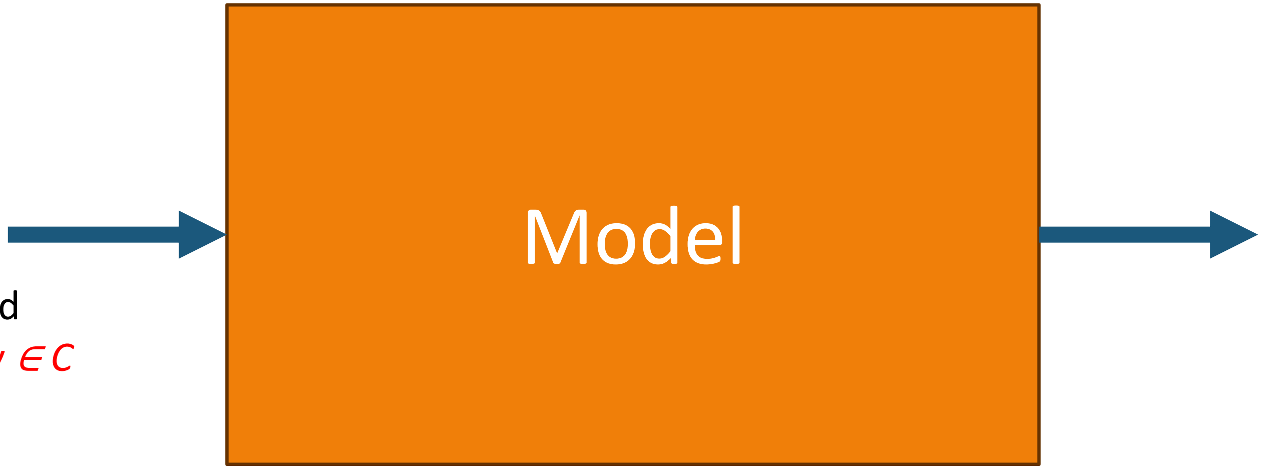
Sentiment analysis

…

a document d

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

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



Model

a learned classifier γ that maps documents to classes

Text Classification: Supervised Machine Learning

Assigning subject categories, topics, or genres

Spam detection

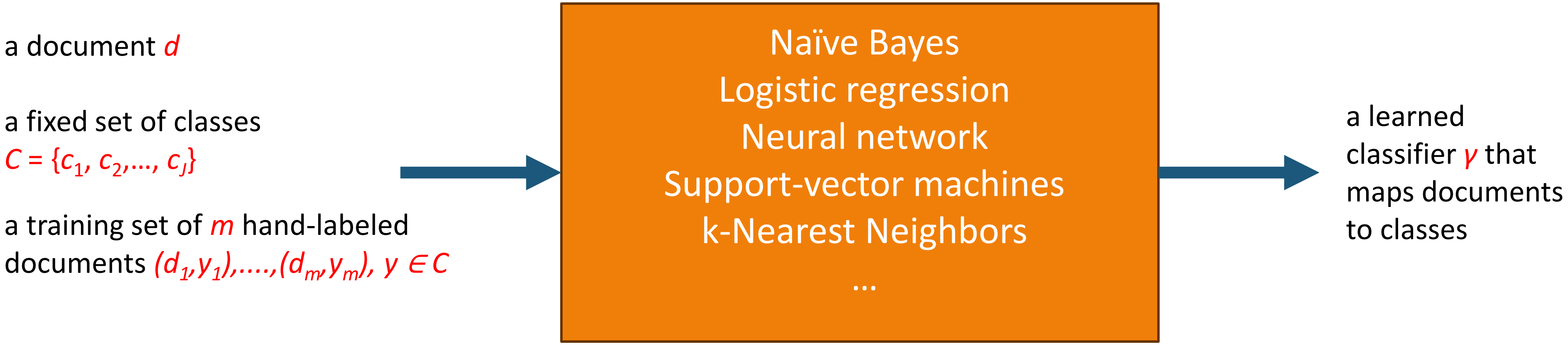
Authorship identification

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 $(d_1,y_1),\ldots.,(d_m,y_m),\ y \in C$

Naïve Bayes
Logistic regression
Neural network
Support-vector machines
k-Nearest Neighbors
…

a learned classifier *γ* that maps documents to classes