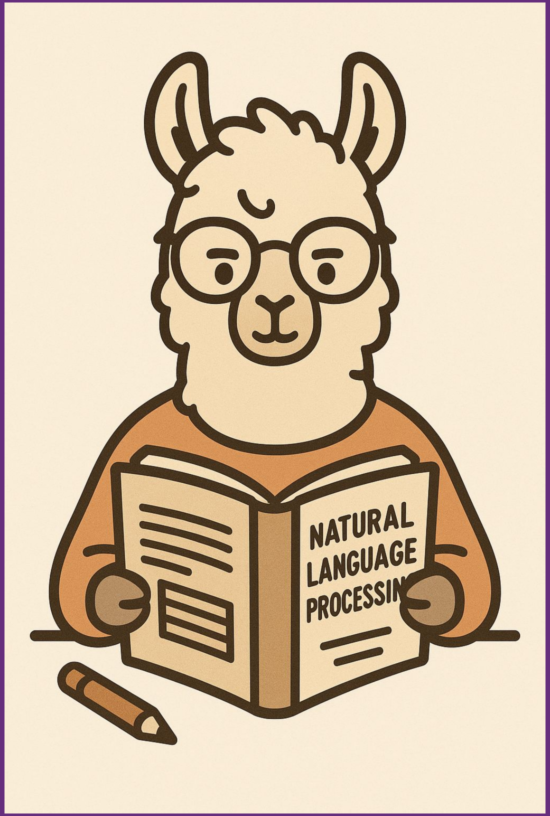


Image generated using Sora

Mila

11-06-2025

Introduction to Natural Language Processing

Part 1

Prakhar Ganesh

Before we start …

How's everyone doing?

Any questions from previous sessions?

Goals today...

- What is '*Natural Language Processing (NLP)*'?
    - Introduction to an interdisciplinary field
- Why do we need NLP?
    - Applications and Challenges
- Different ways of modeling language
    - Bag of Words, Causal Language Modeling, etc.
- Embeddings

What is Natural Language Processing?

What is Natural Language Processing?

It enables computers to understand, interpret and respond to human language.

What is Natural Language Processing?

*It enables **computers** to understand, interpret and respond to human language.*

- Computer Science, Artificial intelligence, Machine learning

What is Natural Language Processing?

*It enables computers to understand, interpret and respond to **human language**.*

- Linguistics, Social Science

Why ‘natural language’? What other kind of language is there?

Why ‘natural language’? What other kind of language is there?

Natural Language

Source: https://www.thoughtco.com/ambiguity-language-1692388

Why ‘natural language’? What other kind of language is there?

Natural Language

Source: https://www.thoughtco.com/ambiguity-language-1692388

Computer Language

```
class Coder(BaseHuman):

  def __init__(self):
    coffee.strength++
    env.update()
    env.theme = DARK

  def day(self):
    self.eat(1*hrs)
    self.code(12*hrs)
    self.eat(1*hrs)
    self.debug(4*hrs)
    time.sleep(6*hrs)
```

Why ‘natural language’? What other kind of language is there?

Natural Language

- Used for everyday communication between people

Computer Language

- Used for instructing computers to perform specific tasks

Why ‘natural language’? What other kind of language is there?

Natural Language

- Used for everyday communication between people
- Developed naturally

Computer Language

- Used for instructing computers to perform specific tasks
- Systematically designed

Why ‘natural language’? What other kind of language is there?

Natural Language

- Used for everyday communication between people
- Developed naturally
- Complex and ambiguous

Computer Language

- Used for instructing computers to perform specific tasks
- Systematically designed
- Precise and unambiguous

Why ‘natural language’? What other kind of language is there?

Natural Language

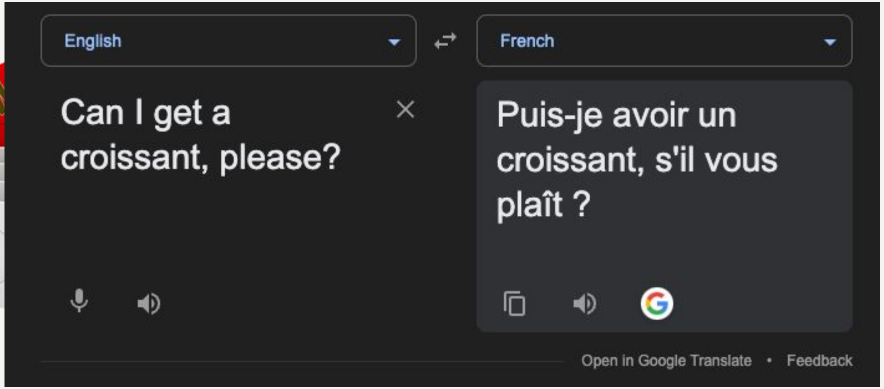
- Used for everyday communication between people
- Developed naturally
- Complex and ambiguous
- Highly nuanced and flexible

Computer Language

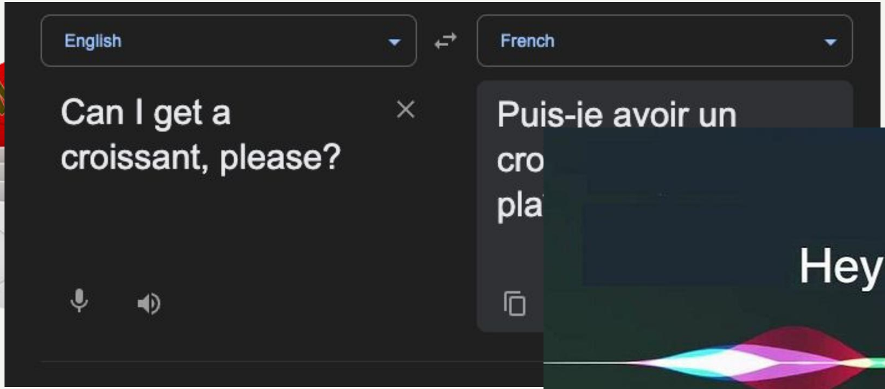
- Used for instructing computers to perform specific tasks
- Systematically designed
- Precise and unambiguous
- Limited in functionality and expressiveness

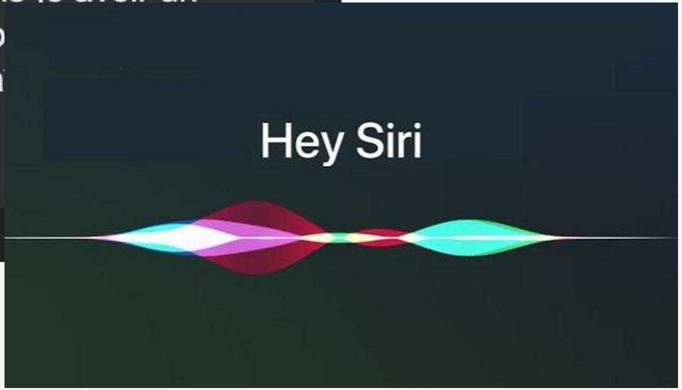
NLP is everywhere around us!

NLP is everywhere around us!

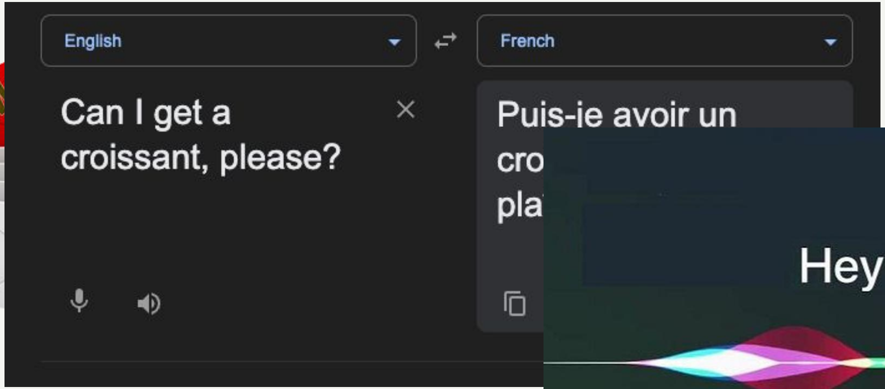


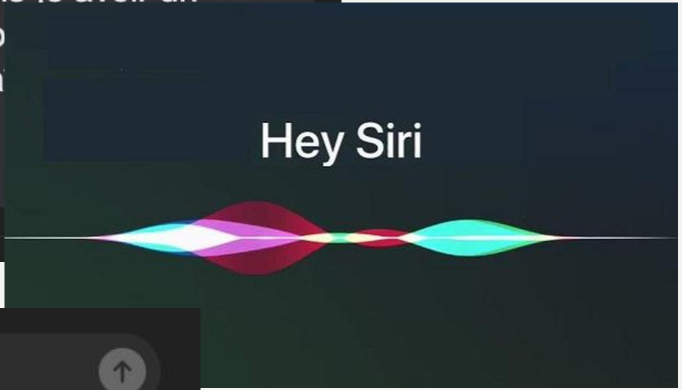
NLP is everywhere around us!

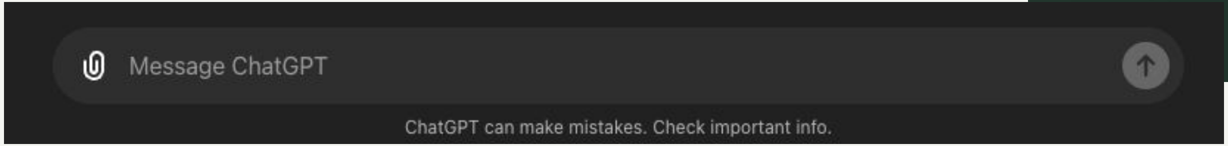




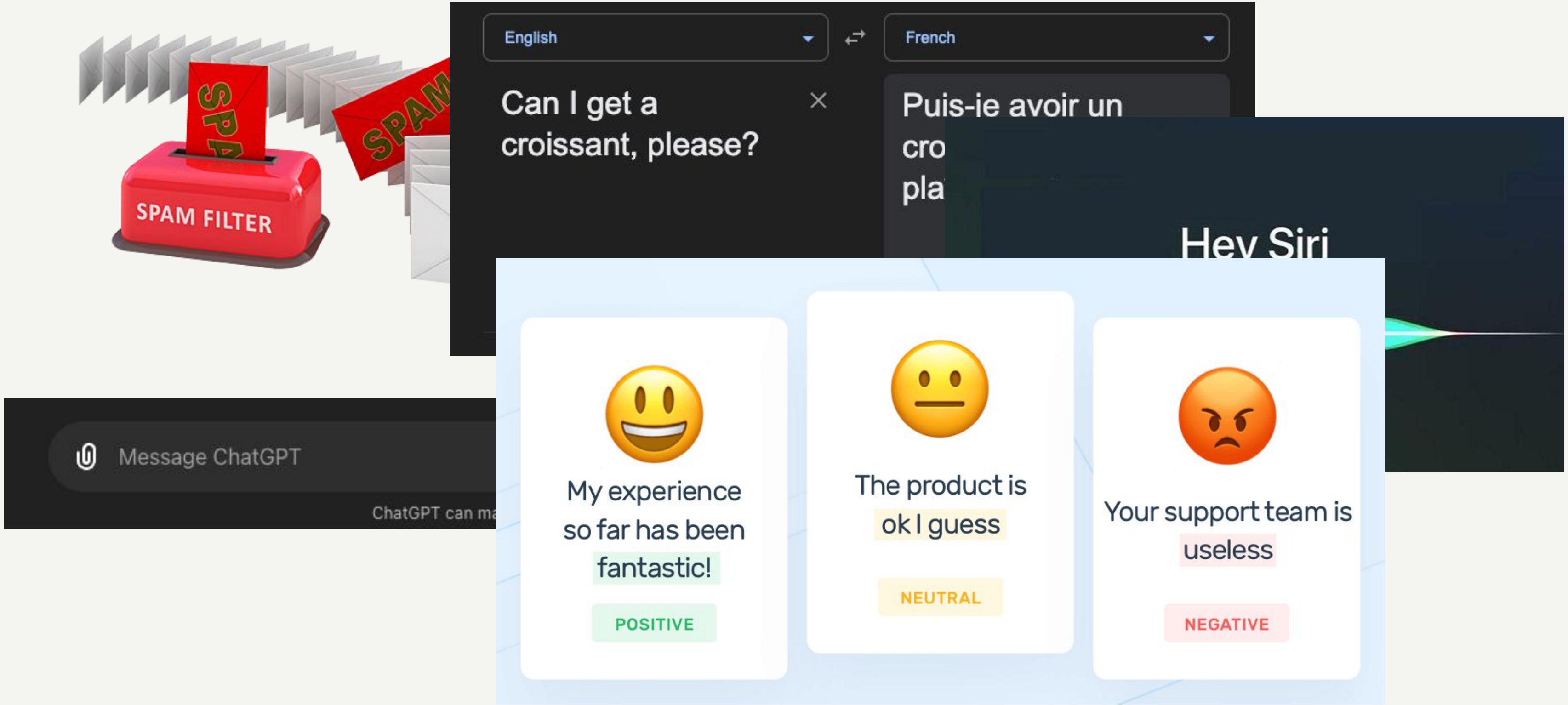
NLP is everywhere around us!



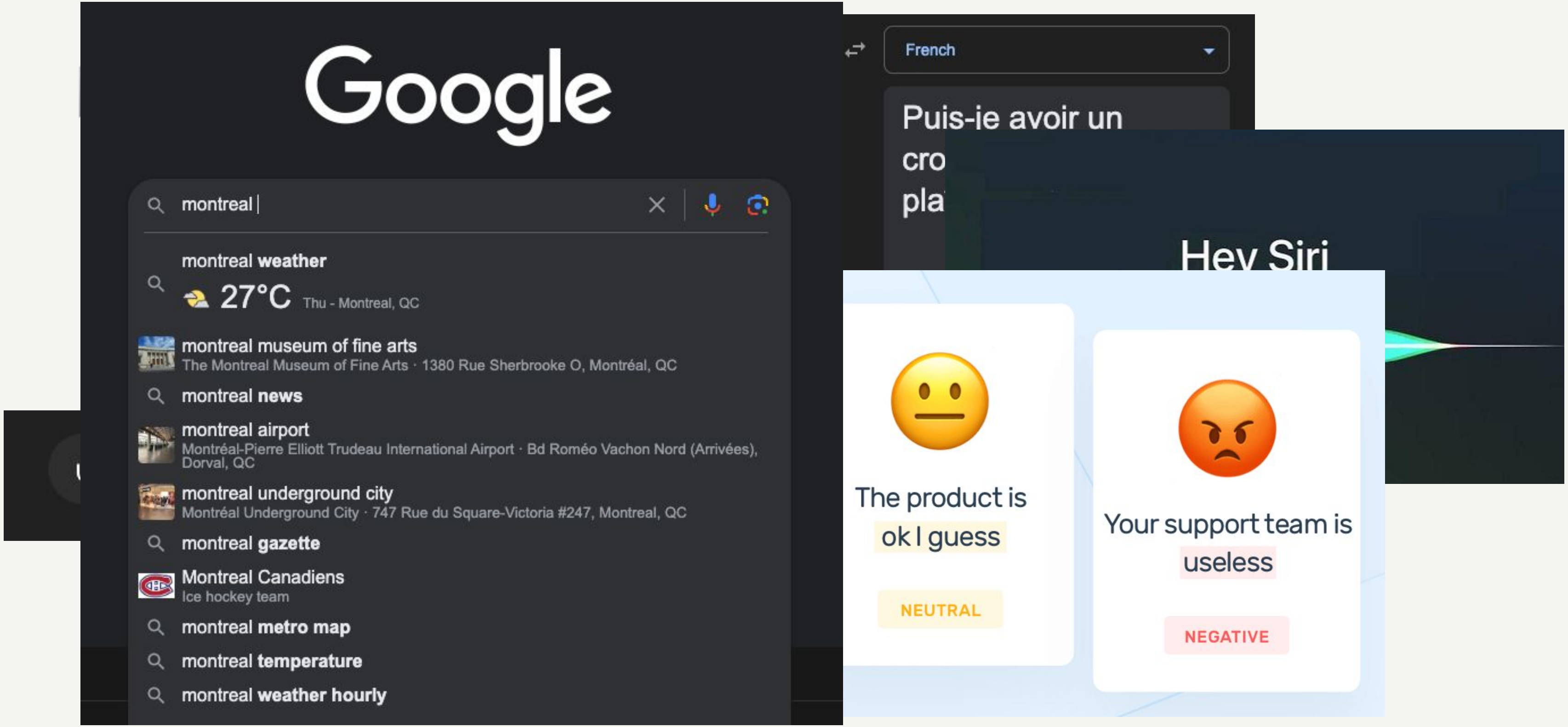




NLP is everywhere around us!

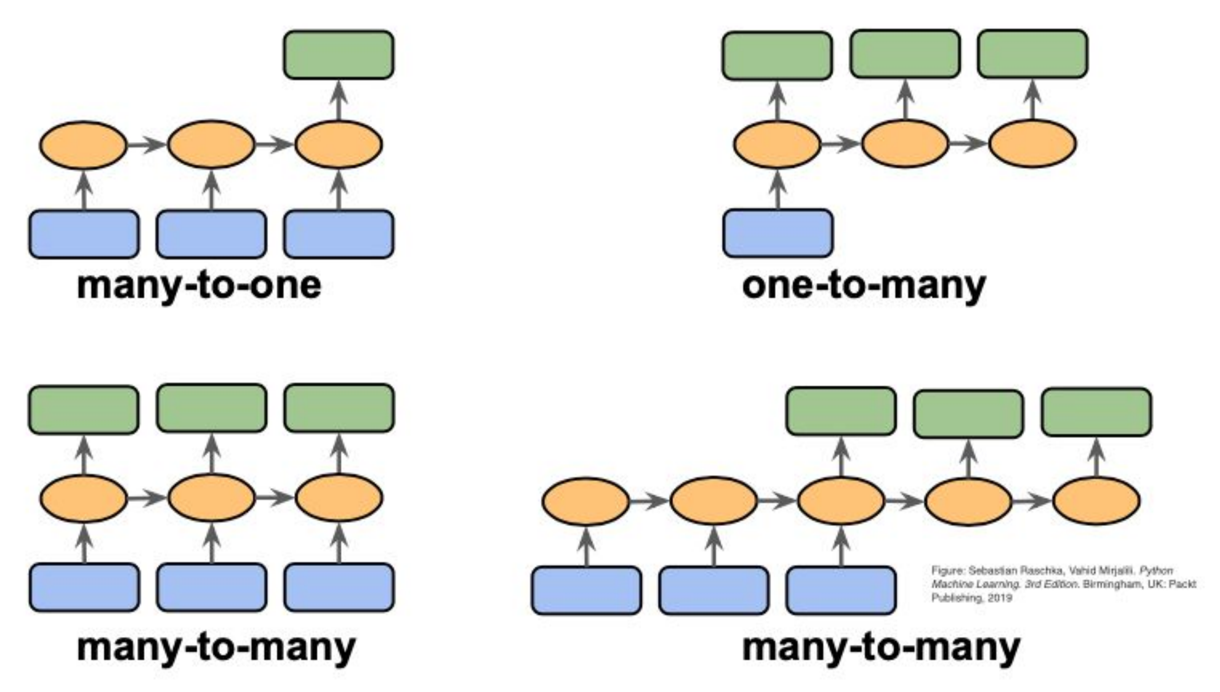


NLP is everywhere around us!



NLP is everywhere around us!

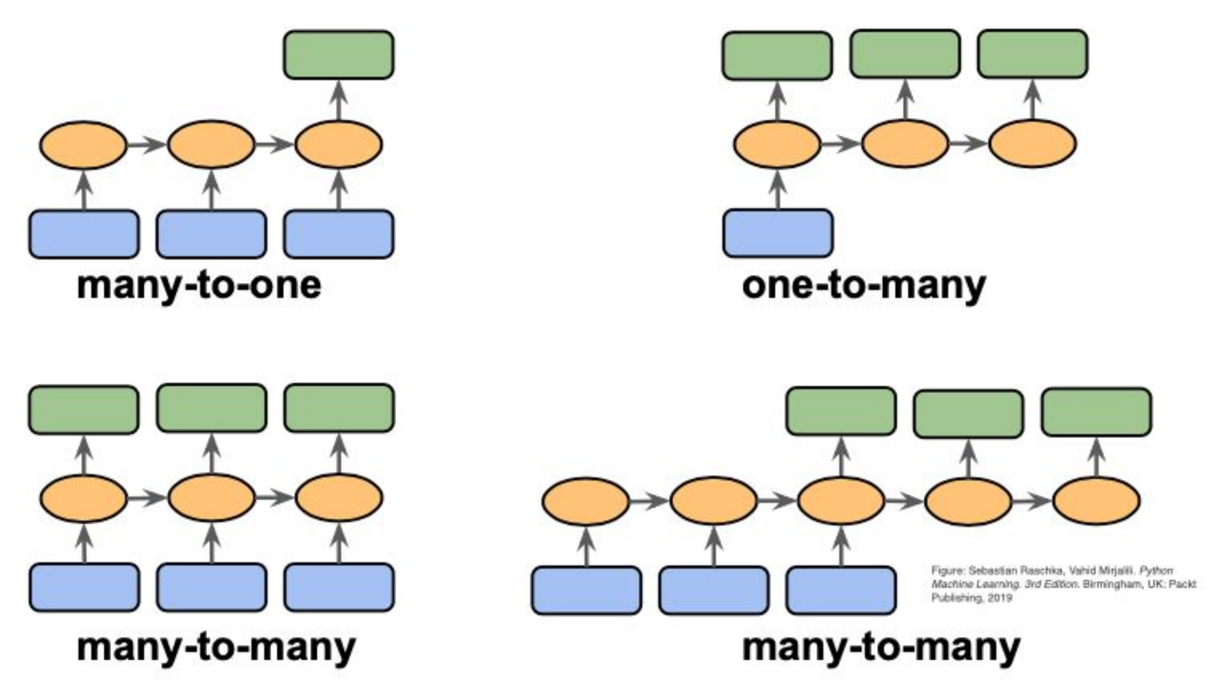
Types of Sequence Modeling



Mila

Types of Sequence Modeling

Example: Text Classification



Types of Sequence Modeling

Example: Text Classification

Example: Image Captioning

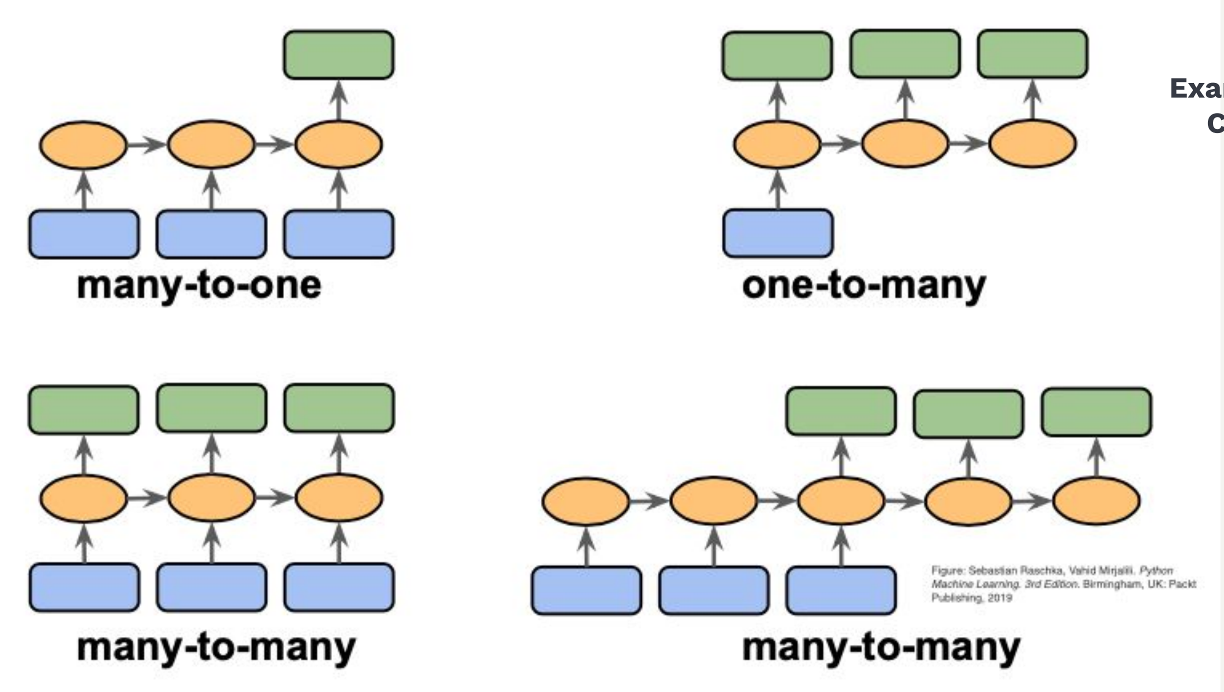


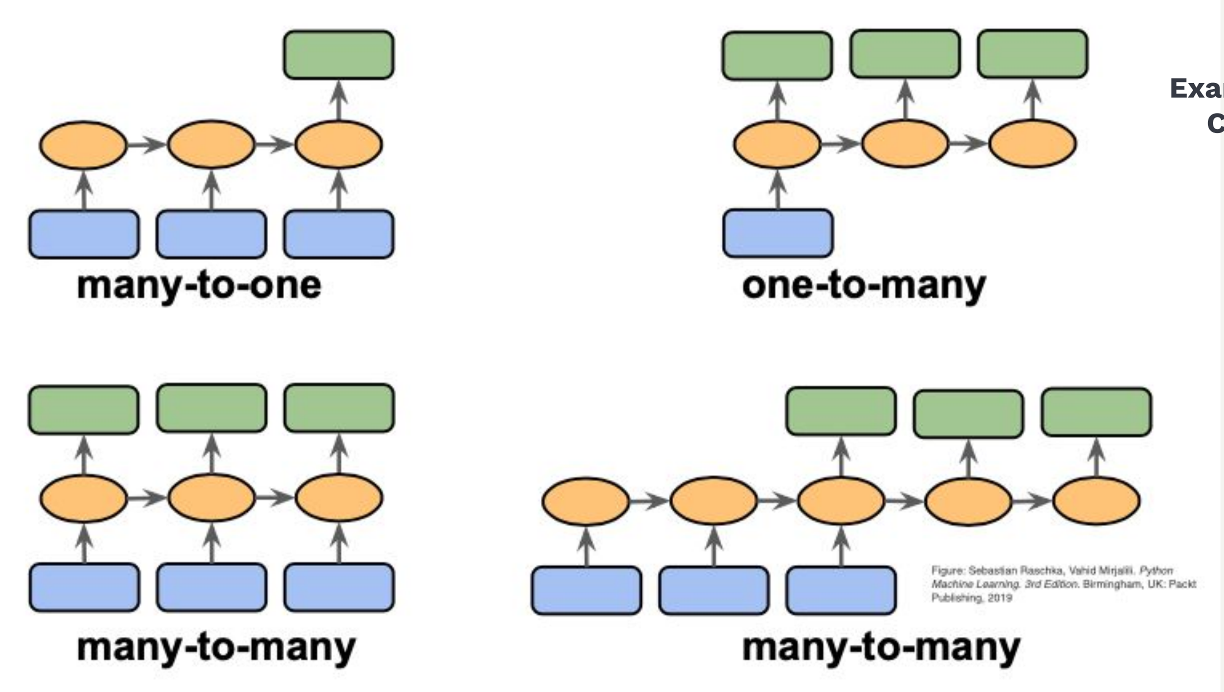
Figure: Sebastian Raschka, Vahid Mirjalili. *Python Machine Learning. 3rd Edition*. Birmingham, UK: Packt Publishing, 2019

Types of Sequence Modeling

Example: Text Classification

Example: Image Captioning

Example: Text to Speech



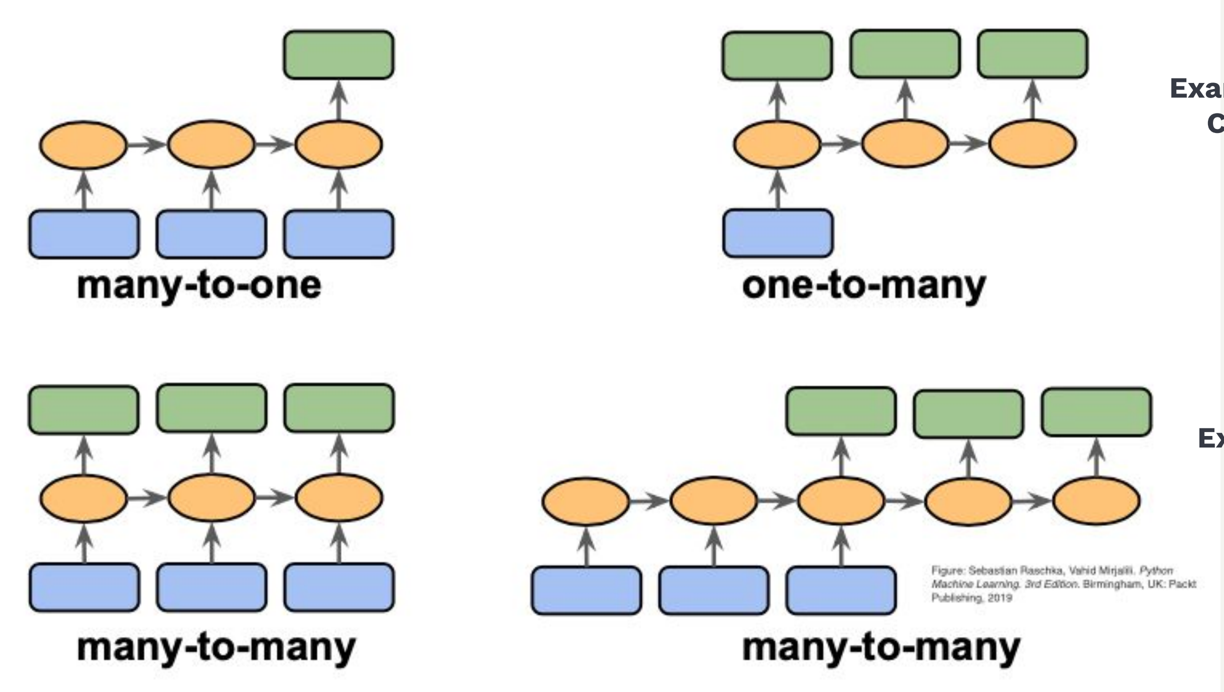
Types of Sequence Modeling

Example: Text Classification

Example: Image Captioning

Example: Text to Speech

Example: Machine Translation



Challenges in NLP

Challenges of NLP: Phrasing Ambiguity

Source: https://blueskiesconsulting.com/how-well-do-you-handle-ambiguity-on-a-project/

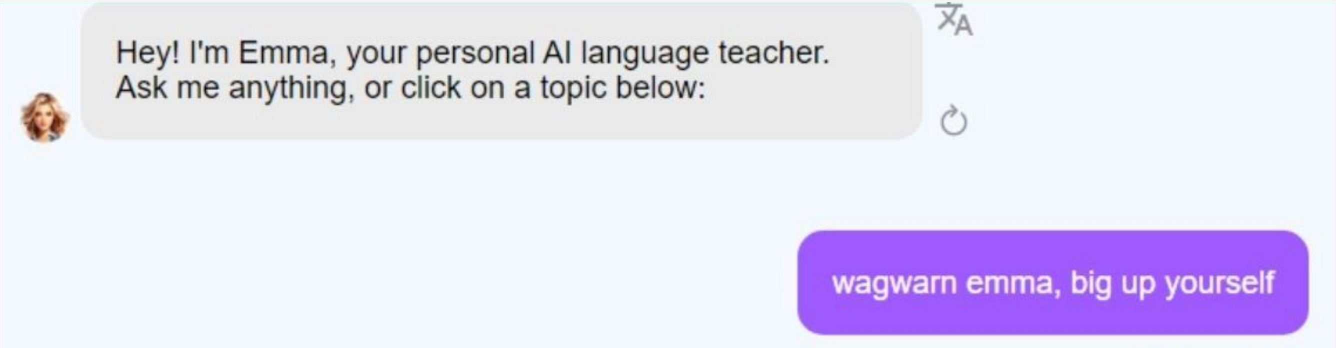
Challenges of NLP: Words with Multiple Meanings

Mila

Challenges of NLP: ~~Mispellings~~

Misspellings

Challenges of NLP: New Vocabulary

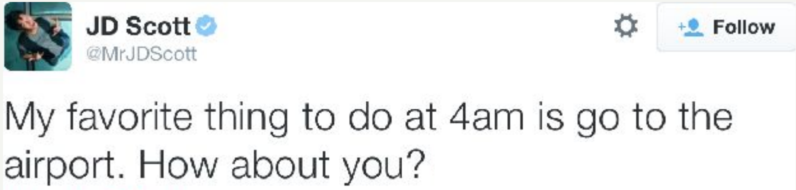


Mila

Challenges of NLP: Specialized Terminology

A 12-year old girl with known hyperagglutinability, presented to the emergency department with a 2-week history of headeaches and facial weakness. Neurologic examination indicated sensorineural hearing loss on the right side with Weber's test lateralizing to the left, and the Rinne's test demonstrating bone conduction greater than air conduction on the right. Magnetic resonance imaging of the head revealed severe structural defects of the right petrous temporal bone. No indication of cerebral infarction.

Challenges of NLP: Tone of Voice



Challenges of NLP: Understanding Context

It’s raining cats and dogs!

Source:
https://medium.com/@InsightfulScribbler/the-curious-history-of-raining-cats-and-dogs-and-interesting-rainy-weather-idioms-from-other-33709f6b7884

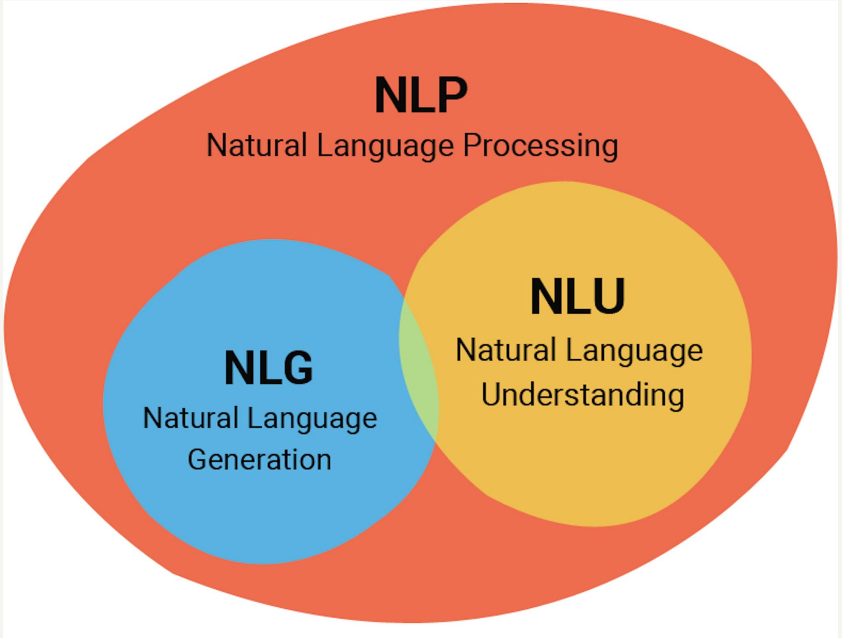
Challenges of NLP: Code Switching

Source: https://www.theinformedslp.com/review/a-little-bit-of-this-un-poquito-of-that

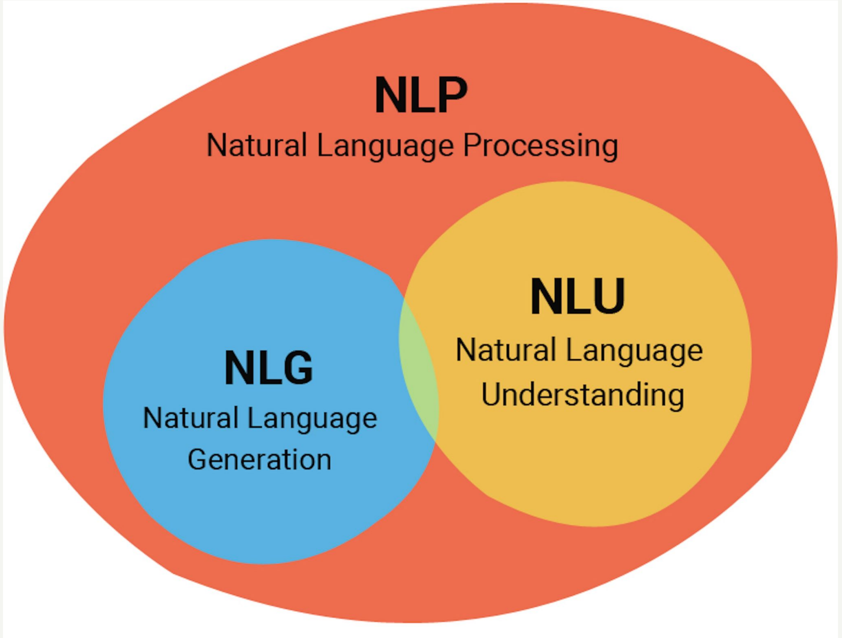
Mila

Terminology

NLP, NLU and NLG



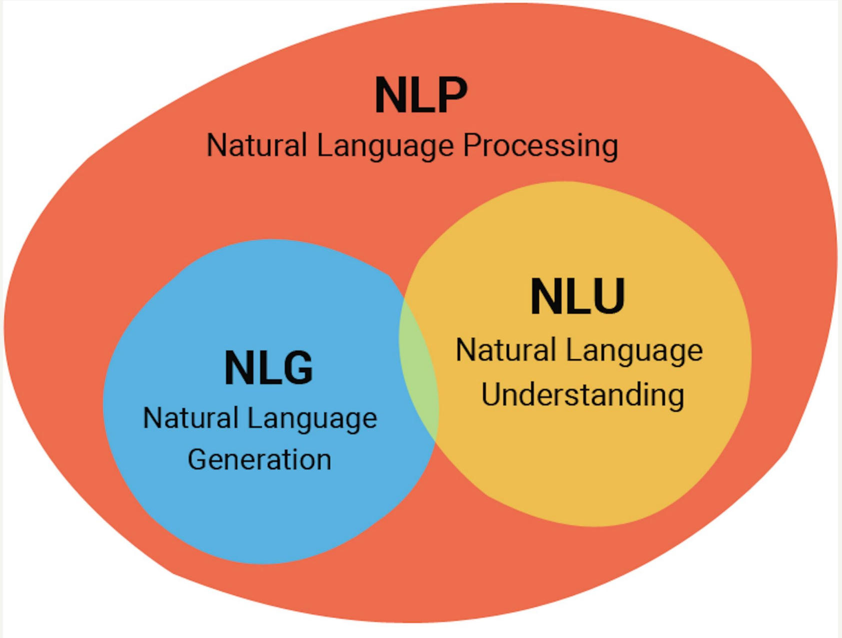
NLP, NLU and NLG



It enables computers to understand and interpret human language.

NLP, NLU and NLG

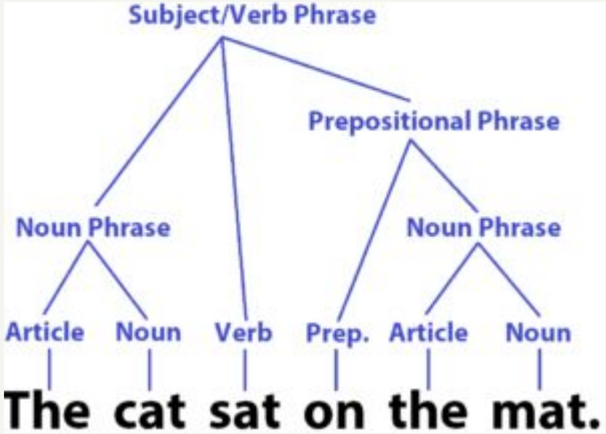
It enables computers to respond, manipulate and generate human language.

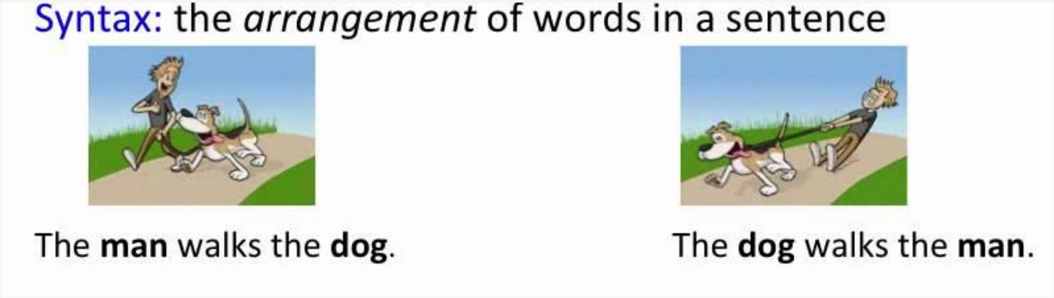


It enables computers to understand and interpret human language.

Syntax

sentence structure and grammar rules

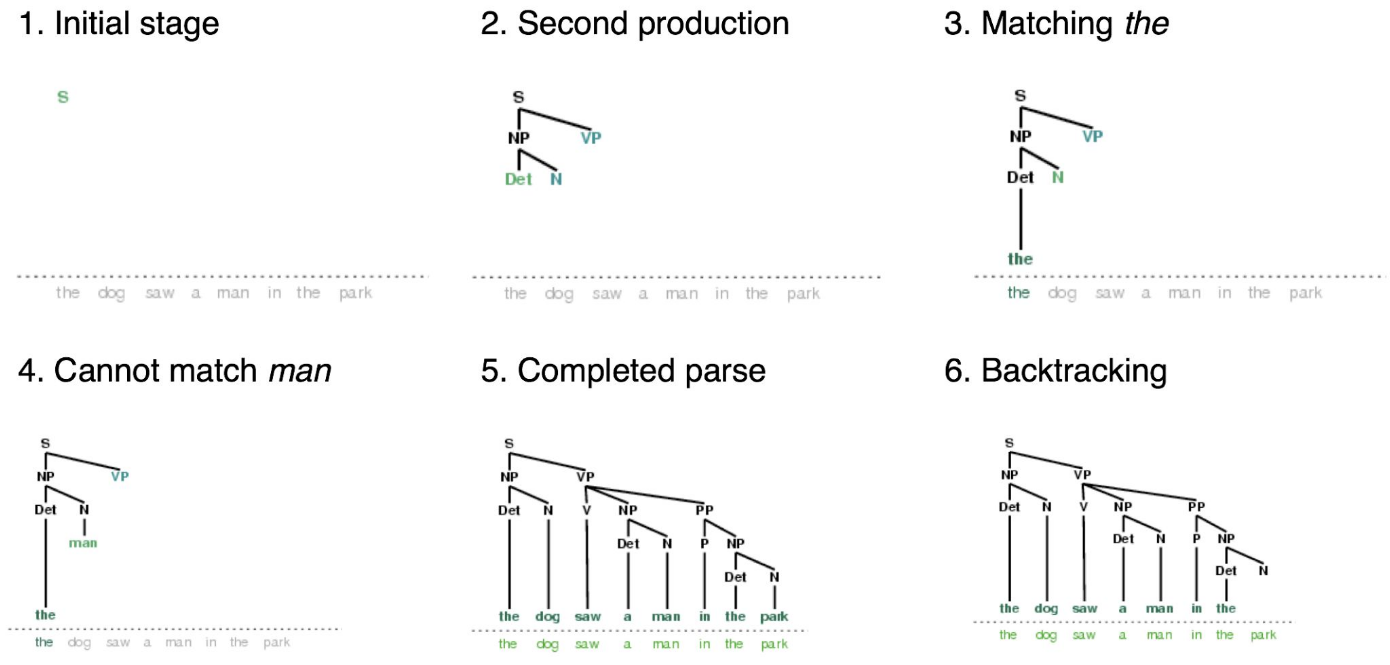




Source: https://www.youtube.com/watch?v=l3mbNkIEcYM

Parsing

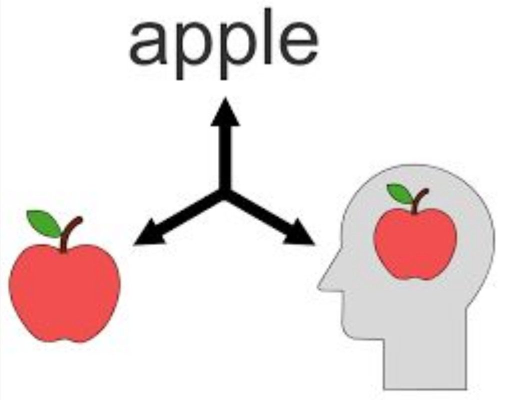
extracting syntax from a sentence



Source: https://www.nltk.org/book/ch08.html

Semantics

meaning of a word



Word	Semantic
pen	a writing tool
pen	a livestock's enclosure
pen	a portable enclosure for a baby
pen	a correctional institution
pen	a female swan

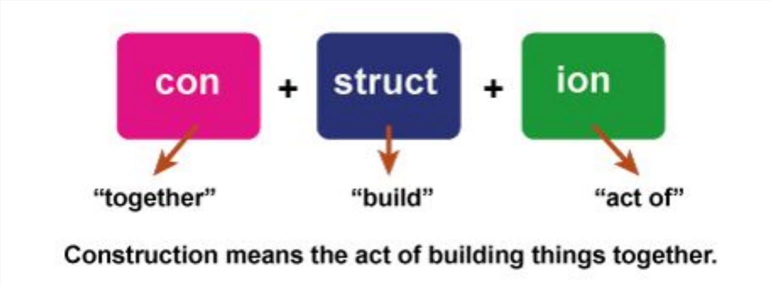
Pragmatics

meaning of a word in context of the sentence

It's hot in here, can you crack a window?

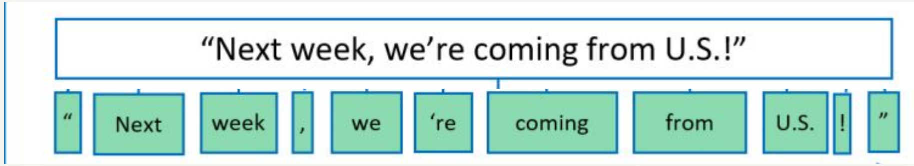
Morphology

the study of how words are formed



Tokenization

splitting text into smaller units (words, phrases, roots, etc.)



Tokenization

Byte Pair Encoding (BPE Tokenization)

Byte Pair Encoding (BPE Tokenization)

low
lower
lowest
new
newest
wider
widest

Byte Pair Encoding (BPE Tokenization)

low = l + o + w
lower = l + o + w + e + r
lowest = l + o + w + e + s + t
new = n + e + w
newest = n + e + w + e + s + t
wider = w + i + d + e + r
widest = w + i + d + e + s + t

Tokens: l, o, w, e, r, s, t, n, i, d

Byte Pair Encoding (BPE Tokenization)

low = **l + o** + w
lower = **l + o** + w + e + r
lowest = **l + o** + w + e + s + t
new = n + e + w
newest = n + e + w + e + s + t
wider = w + i + d + e + r
widest = w + i + d + e + s + t

l + o → 3 times

Tokens: l, o, w, e, r, s, t, n, i, d

Byte Pair Encoding (BPE Tokenization)

low = l + **o + w**
lower = l + **o + w** + e + r
lowest = l + **o + w** + e + s + t
new = n + e + w
newest = n + e + w + e + s + t
wider = w + i + d + e + r
widest = w + i + d + e + s + t

l + o → 3 times
o + w → 3 times

Tokens: l, o, w, e, r, s, t, n, i, d

Byte Pair Encoding (BPE Tokenization)

low = l + o + w
lower = l + o + **w + e** + r
lowest = l + o + **w + e** + s + t
new = n + e + w
newest = n + e + **w + e** + s + t
wider = w + i + d + e + r
widest = w + i + d + e + s + t

l + o → 3 times
o + w → 3 times
w + e → 3 times

Tokens: l, o, w, e, r, s, t, n, i, d

Byte Pair Encoding (BPE Tokenization)

low = l + o + w
lower = l + o + w + e + r
lowest = l + o + w + e + s + t
new = n + e + w
newest = n + e + w + e + s + t
wider = w + i + d + e + r
widest = w + i + d + e + s + t

Tokens: l, o, w, e, r, s, t, n, i, d

l + o → 3 times
o + w → 3 times
w + e → 3 times
e + r → 2 times
e + s → 3 times
s + t → 3 times
n + e → 2 times
e + w → 2 times
w + i → 2 times
i + d → 2 times
d + e → 2 times

Byte Pair Encoding (BPE Tokenization)

low = l + o + w
lower = l + o + w + e + r
lowest = l + o + w + e + s + t
new = n + e + w
newest = n + e + w + e + s + t
wider = w + i + d + e + r
widest = w + i + d + e + s + t

Tokens: l, o, w, e, r, s, t, n, i, d

l + o → 3 times
o + w → 3 times
w + e → 3 times
e + r → 2 times
e + s → 3 times
s + t → 3 times
n + e → 2 times
e + w → 2 times
w + i → 2 times
i + d → 2 times
d + e → 2 times

Byte Pair Encoding (BPE Tokenization)

low = lo + w
lower = lo + w + e + r
lowest = lo + w + e + s + t
new = n + e + w
newest = n + e + w + e + s + t
wider = w + i + d + e + r
widest = w + i + d + e + s + t

Tokens: l, o, w, e, r, s, t, n, i, d, **lo**

Byte Pair Encoding (BPE Tokenization)

low = lo + w
lower = lo + w + e + r
lowest = lo + w + e + s + t
new = n + e + w
newest = n + e + w + e + s + t
wider = w + i + d + e + r
widest = w + i + d + e + s + t

Tokens: l, o, w, e, r, s, t, n, i, d, lo

lo + w → 3 times
w + e → 3 times
e + r → 2 times
e + s → 3 times
s + t → 3 times
n + e → 2 times
e + w → 2 times
w + i → 2 times
i + d → 2 times
d + e → 2 times

Byte Pair Encoding (BPE Tokenization)

low = lo + w
lower = lo + w + e + r
lowest = lo + w + e + s + t
new = n + e + w
newest = n + e + w + e + s + t
wider = w + i + d + e + r
widest = w + i + d + e + s + t

Tokens: l, o, w, e, r, s, t, n, i, d, lo

lo + w → 3 times
w + e → 3 times
e + r → 2 times
e + s → 3 times
s + t → 3 times
n + e → 2 times
e + w → 2 times
w + i → 2 times
i + d → 2 times
d + e → 2 times

Byte Pair Encoding (BPE Tokenization)

low = low
lower = low + e + r
lowest = low + e + s + t
new = n + e + w
newest = n + e + w + e + s + t
wider = w + i + d + e + r
widest = w + i + d + e + s + t

Tokens: l, o, w, e, r, s, t, n, i, d, lo, **low**

Byte Pair Encoding (BPE Tokenization)

low = low
lower = low + e + r
lowest = low + e + s + t
new = n + e + w
newest = n + e + w + e + s + t
wider = w + i + d + e + r
widest = w + i + d + e + s + t

Tokens: l, o, w, e, r, s, t, n, i, d, lo, low

low + e → 2 times
w + e → 1 times
e + r → 2 times
e + s → 3 times
s + t → 3 times
n + e → 2 times
e + w → 2 times
w + i → 2 times
i + d → 2 times
d + e → 2 times

Byte Pair Encoding (BPE Tokenization)

low = low
lower = low + e + r
lowest = low + e + s + t
new = n + e + w
newest = n + e + w + e + s + t
wider = w + i + d + e + r
widest = w + i + d + e + s + t

Tokens: l, o, w, e, r, s, t, n, i, d, lo, low

low + e → 2 times
w + e → 1 times
e + r → 2 times
e + s → 3 times
s + t → 3 times
n + e → 2 times
e + w → 2 times
w + i → 2 times
i + d → 2 times
d + e → 2 times

Mila

Byte Pair Encoding (BPE Tokenization)

low = low
lower = low + e + r
lowest = low + e + s + t
new = n + e + w
newest = n + e + w + e + s + t
wider = w + i + d + e + r
widest = w + i + d + e + s + t

Tokens: l, o, w, e, r, s, t, n, i, d, lo, low

After a few steps…

low + e → 2 times
w + e → 1 times
e + r → 2 times
e + s → 3 times
s + t → 3 times
n + e → 2 times
e + w → 2 times
w + i → 2 times
i + d → 2 times
d + e → 2 times

Byte Pair Encoding (BPE Tokenization)

low = low
lower = low + er
lowest = low + est
new = new
newest = new + est
wider = wid + er
widest = wid + est

Tokens: l, o, w, e, r, s, t, n, i, d, lo, low, es, est, er, ne, new, wi, wid

Byte Pair Encoding (BPE Tokenization)

low = low
lower = low + er
lowest = low + est
new = new
newest = new + est
wider = wid + er
widest = wid + est

newer = new + er
wide = wid + e

lost = lo + s + t
worst = w + o + r + s + t
wise = wi + s + e

Tokens: l, o, w, e, r, s, t, n, i, d, lo, low, es, est, er, ne, new, wi, wid

Before Tokenization: Text Preprocessing

- **Lower Casing:** lOOK at that DUck! → look at that duck!

Before Tokenization: Text Preprocessing

- **Lower Casing:** lOOK at that DUck! → look at that duck!
- **Removing punctuations, stop words, special characters, etc.:** Holy sh!t, look at that duck!!! → look duck

Before Tokenization: Text Preprocessing

- **Lower Casing:** lOOK at that DUck! → look at that duck!
- **Removing punctuations, stop words, special characters, etc.:** Holy sh!t, look at that duck!!! → look duck
- **Normalization and Spell Correction:** U.K., UK, U K → United Kingdom

Before Tokenization: Text Preprocessing

- **Lower Casing:** lOOK at that DUck! → look at that duck!
- **Removing punctuations, stop words, special characters, etc.:** Holy sh!t, look at that duck!!! → look duck
- **Normalization and Spell Correction:** U.K., UK, U K → United Kingdom
- **Stemming and Lemmatization:** running → run
  fast, faster, fastest → fast

Before Tokenization: Text Preprocessing

- **Lower Casing:** lOOK at that DUck! → look at that duck!
- **Removing punctuations, stop words, special characters, etc.:** Holy sh!t, look at that duck!!! → look duck
- **Normalization and Spell Correction:** U.K., UK, U K → United Kingdom
- **Stemming and Lemmatization:** running → run
  fast, faster, fastest → fast

Just a lot of cleaning!
A relic of NLP pre-deep learning

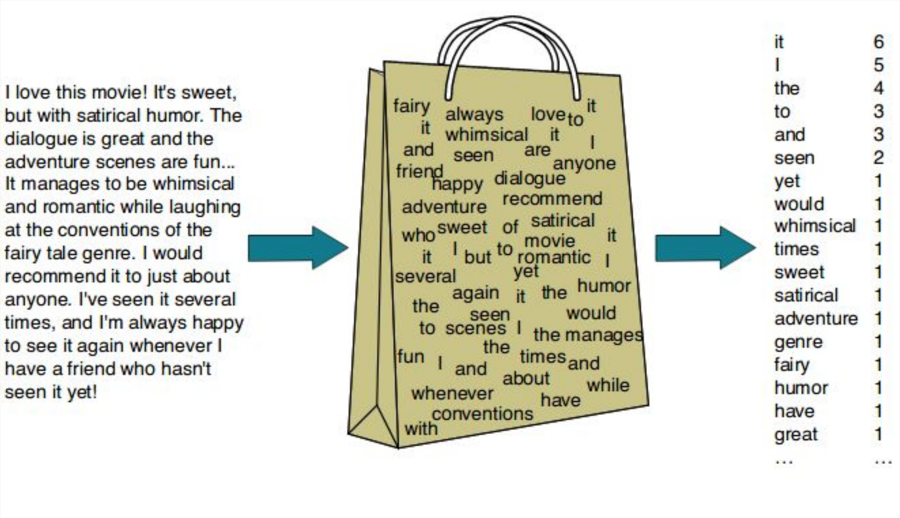
Modeling Language

Bag of Words

Order of the words doesn't matter, only their occurrence matters.

Bag of Words

Order of the words doesn't matter, only their occurrence matters.



Source: https://koushik1102.medium.com/nlp-bag-of-words-and-tf-idf-explained-fd1f49dce7c4

Bag of Words

Order of the words doesn’t matter, only their occurrence matters.

- Simple, efficient, and a decent baseline.

Positive or Negative Sentiment?

Bag of Words

Order of the words doesn't matter, only their occurrence matters.

- Simple, efficient, and a decent baseline.
- **Ignores context!**

Mila

Bag of Words

Order of the words doesn't matter, only their occurrence matters.

- Simple, efficient, and a decent baseline.
- **Ignores context!**

Other reviews said it was disappointing, but I felt it was good.

Other reviews said it was good, but I felt it was disappointing.

n-gram Models

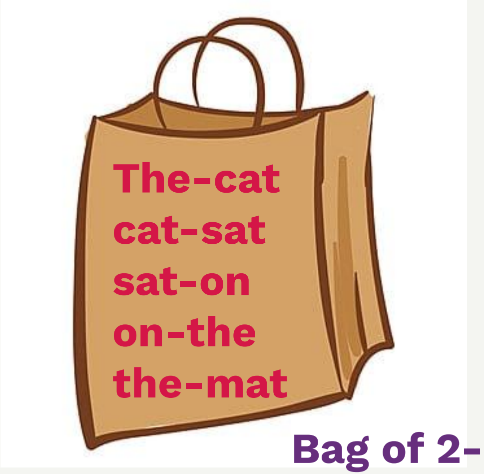
Order of the **n-grams** doesn't matter, only their occurrence matters.

n-gram Models

Order of the **n-grams** doesn't matter, only their occurrence matters.

The cat sat on the mat.

Bag of Words



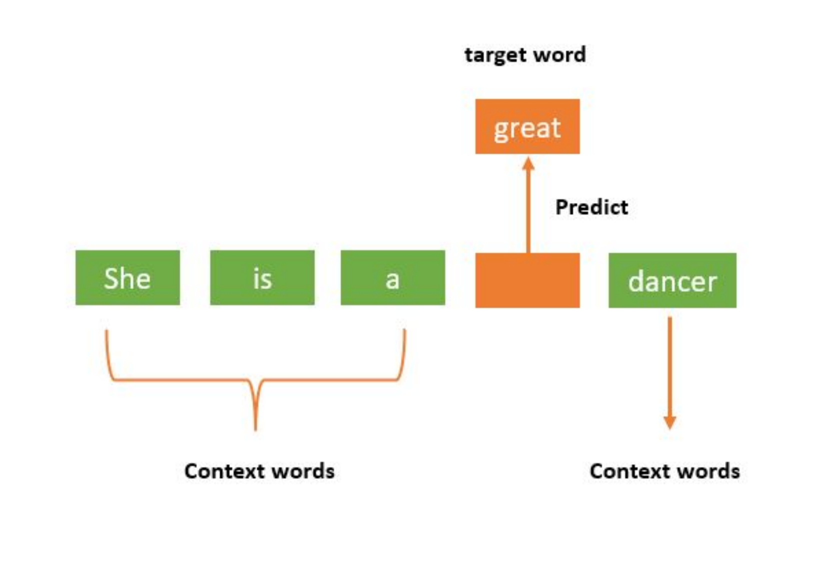
Bag of 2-grams

Continuous Bag of Words

"You shall know a word by the company it keeps" - J.R. Firth

Continuous Bag of Words

"You shall know a word by the company it keeps" - J.R. Firth



Masked Language Modeling

Sentence:

The keys to the cabinet [MASK] on the table.

Mask 1 Predictions:

70.3% **were**
10.1% **lay**

Sentence:

The [MASK] to the cabinet were on the table.

Mask 1 Predictions:

89.7% **keys**
1.7% **contents**

Sentence:

The [MASK] to the cabinet [MASK] on the table.

Mask 1 Predictions:

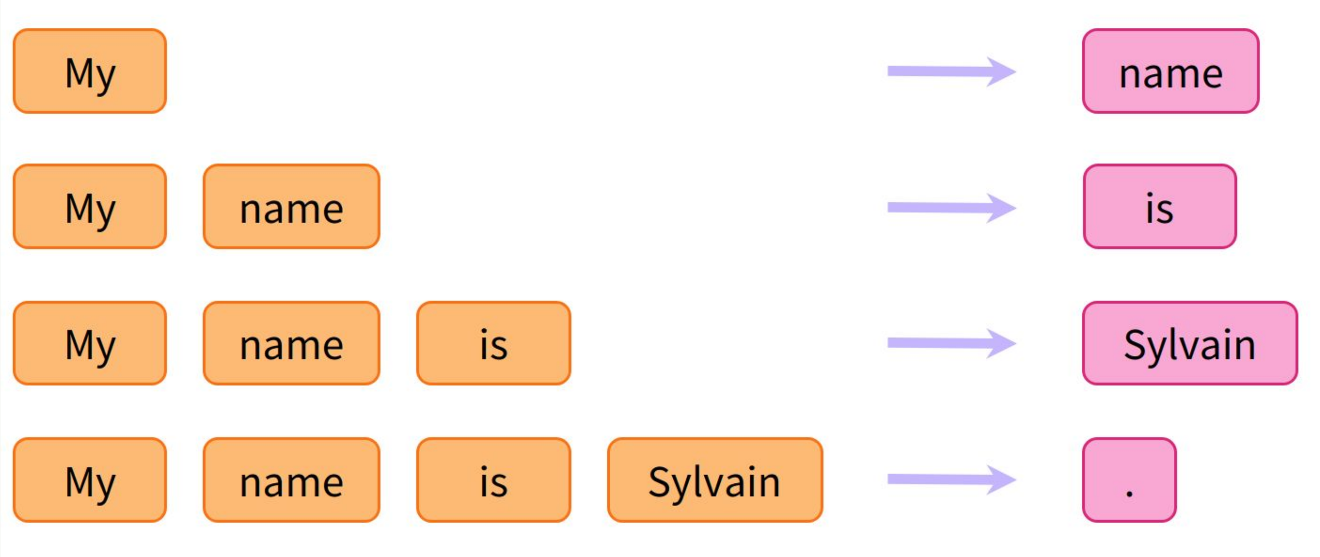
70.8% **keys**
18.2% **key**

Mask 2 Predictions:

36.6% **was**
9.0% **were**

Causal Language Modeling

Predicting the next word based on previous words.



Mila

Causal Language Modeling

Embeddings

Why Embeddings?

Why Embeddings?

Blood pressure = w*Dosage + b ⇒ **134 = 0.7*20 + 120**

Makes sense

Why Embeddings?

Blood pressure = w*Dosage + b ⇒ **134 = 0.7*20 + 120**

Makes sense

```
[112, 111,  98,  79,  97, 130, 124, 122, 127,  72]
[142, 124, 103, 104,  91,  92, 110, 100, 114,  65]
[137, 137, 119, 100,  98,  85,  98,  86,  94,  55]
[147, 142, 145, 129, 113,  99,  86,  81,  87,  62]
[143, 140, 141, 139, 137, 135, 153,  98,  87,  55]
[147, 151, 150, 148, 115, 163, 241, 170, 111,  82]
[152, 153, 152, 141,  60, 129, 202, 150, 127, 146]
[134, 142, 152,  99,  52,  90, 128, 107, 134, 148]
[136, 135, 131,  56,  74,  94, 119, 133, 144, 143]
[133, 138, 105,  50,  79,  87,  93, 137, 146, 145]
[131, 136,  90,  64,  80,  89,  80, 130, 135, 137]
[127, 125,  67,  80,  71,  85,  92, 134, 137, 131]
[118, 119,  48,  76,  73,  60,  88, 134, 133, 136]
[119, 114,  52, 100,  60,  10,  42, 101, 123, 132]
[114, 106,  81, 113,  22,  14,  59, 120, 131, 126]
[107, 109,  92,  65,  20,  64, 121, 125, 128, 134]
[107, 110,  35,  37,  75, 123, 136, 127, 124, 130]
[104, 121,  94, 111, 124, 124, 129, 130, 118, 124]
[102, 119, 127, 125, 122, 128, 129, 125, 126, 114]
[114, 117, 116, 115, 108, 116, 124, 120, 131, 128]
[113, 111, 108, 101, 104, 112, 112, 120, 118, 114]
[101, 105, 101,  91,  98, 107, 103, 109, 108, 109]
[ 98,  94,  96,  94,  93,  99,  90, 100, 110, 115]
[103,  99,  92,  91,  96,  98,  87,  89,  91, 103]
[102,  95,  90,  90,  92,  99,  91,  97,  95,  92]
```

"Applying a filter"

$$\underbrace{\begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}}_{\text{filter}} * \underbrace{\begin{bmatrix} 1 & 3 & 4 \\ 2 & 1 & 1 \\ 2 & 5 & 2 \end{bmatrix}}_{\text{input}} = \begin{matrix} 1*1 + 0*3 + (-1)*4 \\ + \; 2*2 + 0*1 + (-2)*1 = -1 \\ + \; 1*2 + 0*5 + (-1)*2 \end{matrix}$$

Makes sense

Mila

Why Embeddings?

Blood pressure = w*Dosage + b ⇒ **134 = 0.7*20 + 120**

Makes sense

```
[112, 111,  98,  79,  97, 130, 124, 122, 127,  72]
[142, 124, 103, 104,  91,  92, 110, 100, 114,  65]
[137, 137, 119, 100,  98,  85,  98,  86,  94,  55]
[147, 142, 145, 129, 113,  99,  86,  81,  87,  62]
[143, 140, 141, 139, 137, 135, 153,  98,  87,  55]
[147, 151, 150, 148, 115, 163, 241, 170, 111,  82]
[152, 153, 152, 141,  60, 129, 202, 150, 127, 146]
[134, 142, 152,  99,  52,  90, 128, 107, 134, 148]
[136, 135, 131,  56,  74,  94, 119, 133, 144, 143]
[133, 138, 105,  50,  79,  87,  93, 137, 146, 145]
[131, 136,  90,  64,  80,  89,  80, 130, 135, 137]
[127, 125,  67,  80,  71,  85,  92, 134, 137, 131]
[118, 119,  48,  76,  73,  60,  88, 134, 133, 136]
[119, 114,  52, 100,  60,  10,  42, 101, 123, 132]
[114, 106,  81, 113,  22,  14,  59, 120, 131, 126]
[107, 109,  92,  65,  20,  64, 121, 125, 128, 134]
[107, 110,  35,  37,  75, 123, 136, 127, 124, 130]
[104, 121,  94, 111, 124, 124, 129, 130, 118, 124]
[102, 119, 127, 125, 122, 128, 129, 125, 126, 114]
[114, 117, 116, 115, 108, 116, 124, 120, 131, 128]
[113, 111, 108, 101, 104, 112, 112, 120, 118, 114]
[101, 105, 101,  91,  98, 107, 103, 109, 108, 109]
[ 98,  94,  96,  94,  93,  99,  90, 100, 110, 115]
[103,  99,  92,  91,  96,  98,  87,  89,  91, 103]
[102,  95,  90,  90,  92,  99,  91,  97,  95,  92]
```

"Applying a filter"

$$\underbrace{\begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}}_{\text{filter}} * \underbrace{\begin{bmatrix} 1 & 3 & 4 \\ 2 & 1 & 1 \\ 2 & 5 & 2 \end{bmatrix}}_{\text{input}} = \begin{array}{l} 1*1 + 0*3 + (-1)*4 \\ + \; 2*2 + 0*1 + (-2)*1 = -1 \\ + \; 1*2 + 0*5 + (-1)*2 \end{array}$$

Makes sense

The cat sat on the mat ⇒ The*0.7 + cat*1.3 + ...

????

Mila

Why Embeddings?

Blood pressure = w*Dosage + b ⇒ **134 = 0.7*20 + 120**

Makes sense

```
[112, 111,  98,  79,  97, 130, 124, 122, 127,  72]
[142, 124, 103, 104,  91,  92, 110, 100, 114,  65]
[137, 137, 119, 100,  98,  85,  98,  86,  94,  55]
[147, 142, 145, 129, 113,  99,  86,  81,  87,  62]
[143, 140, 141, 139, 137, 135, 153,  98,  87,  55]
[147, 151, 150, 148, 115, 163, 241, 170, 111,  82]
[152, 153, 152, 141,  60, 129, 202, 150, 127, 146]
[134, 142, 152,  99,  52,  90, 128, 107, 134, 148]
[136, 135, 131,  56,  74,  94, 119, 133, 144, 143]
[133, 138, 105,  50,  79,  87,  93, 137, 146, 145]
[131, 136,  90,  64,  80,  89,  80, 130, 135, 137]
[127, 125,  67,  80,  71,  85,  92, 134, 137, 131]
[118, 119,  48,  76,  73,  60,  88, 134, 133, 136]
[119, 114,  52, 100,  60,  10,  42, 101, 123, 132]
[114, 106,  81, 113,  22,  14,  59, 120, 131, 126]
[107, 109,  92,  65,  20,  64, 121, 125, 128, 134]
[107, 110,  35,  37,  75, 123, 136, 127, 124, 130]
[104, 121,  94, 111, 124, 124, 129, 130, 118, 124]
[102, 119, 127, 125, 122, 128, 129, 125, 126, 114]
[114, 117, 116, 115, 108, 116, 124, 120, 131, 128]
[113, 111, 108, 101, 104, 112, 112, 120, 118, 114]
[101, 105, 101,  91,  98, 107, 103, 109, 108, 109]
[ 98,  94,  96,  94,  93,  99,  90, 100, 110, 115]
[103,  99,  92,  91,  96,  98,  87,  89,  91, 103]
[102,  95,  90,  90,  92,  99,  91,  97,  95,  92]
```

"Applying a filter"

$$\underbrace{\begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}}_{\text{filter}} * \underbrace{\begin{bmatrix} 1 & 3 & 4 \\ 2 & 1 & 1 \\ 2 & 5 & 2 \end{bmatrix}}_{\text{input}} = \begin{array}{l} 1*1 + 0*3 + (-1)*4 \\ + \; 2*2 + 0*1 + (-2)*1 = -1 \\ + \; 1*2 + 0*5 + (-1)*2 \end{array}$$

Makes sense

We need a way to numerically represent language

The cat sat on the mat ⇒ The*0.7 + cat*1.3 + ...

????

Mila

Embeddings as Sequential Numbering

The cat sat on the mat
1 2 3 4 1 5

Will this work?

Embeddings as Sequential Numbering

The cat sat on the mat
1 2 3 4 1 5

Will this work?

Are the words ‘the’ and ‘cat’ similar? 2-1 = 1. Yes
Are the words ‘the’ and ‘mat’ similar? 5-1 = 4. No

We have encoded *wrong* similarity information into these embeddings without even wanting to!

Embeddings as One Hot Encoding

$$\begin{pmatrix} the \\ cat \\ sat \\ on \\ the \\ mat \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Is this better?

Embeddings as One Hot Encoding

$$\begin{pmatrix} the \\ cat \\ sat \\ on \\ the \\ mat \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Is this better?

Better. Distance or 'similarity' between any 2 feature vectors is now the same!
But we're not done yet.

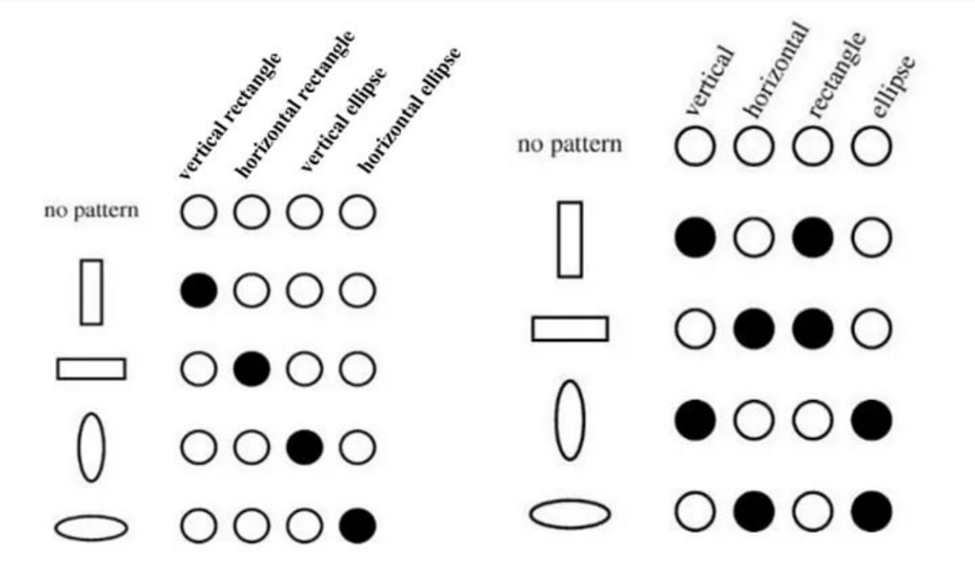
This representation does not have the problems of sequential numbering but it also **holds no similarity information** about the relationship between words.

Embeddings as Distributed Representation

Numerical representation with **correct** comparative value!

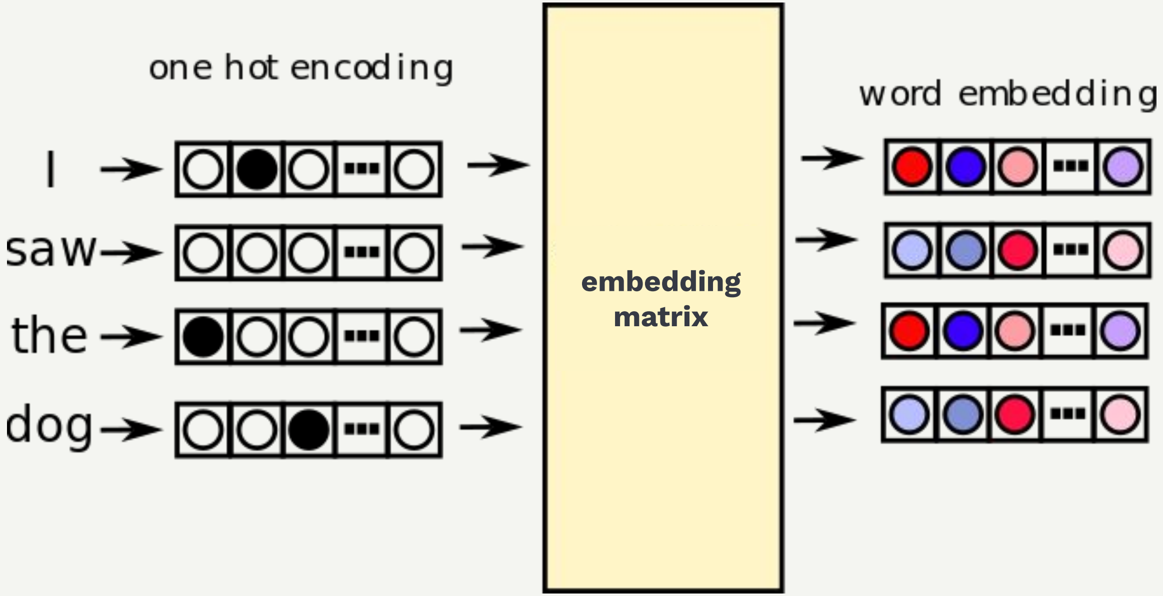
Embeddings as Distributed Representation

Numerical representation with **correct** comparative value!



One hot encoding   Distributed Representation

Embedding Matrix



Mila

Embedding Matrix

Once we have numerical representation of the language, we can use the learning methods we studied earlier.

And some special methods designed just for NLP!

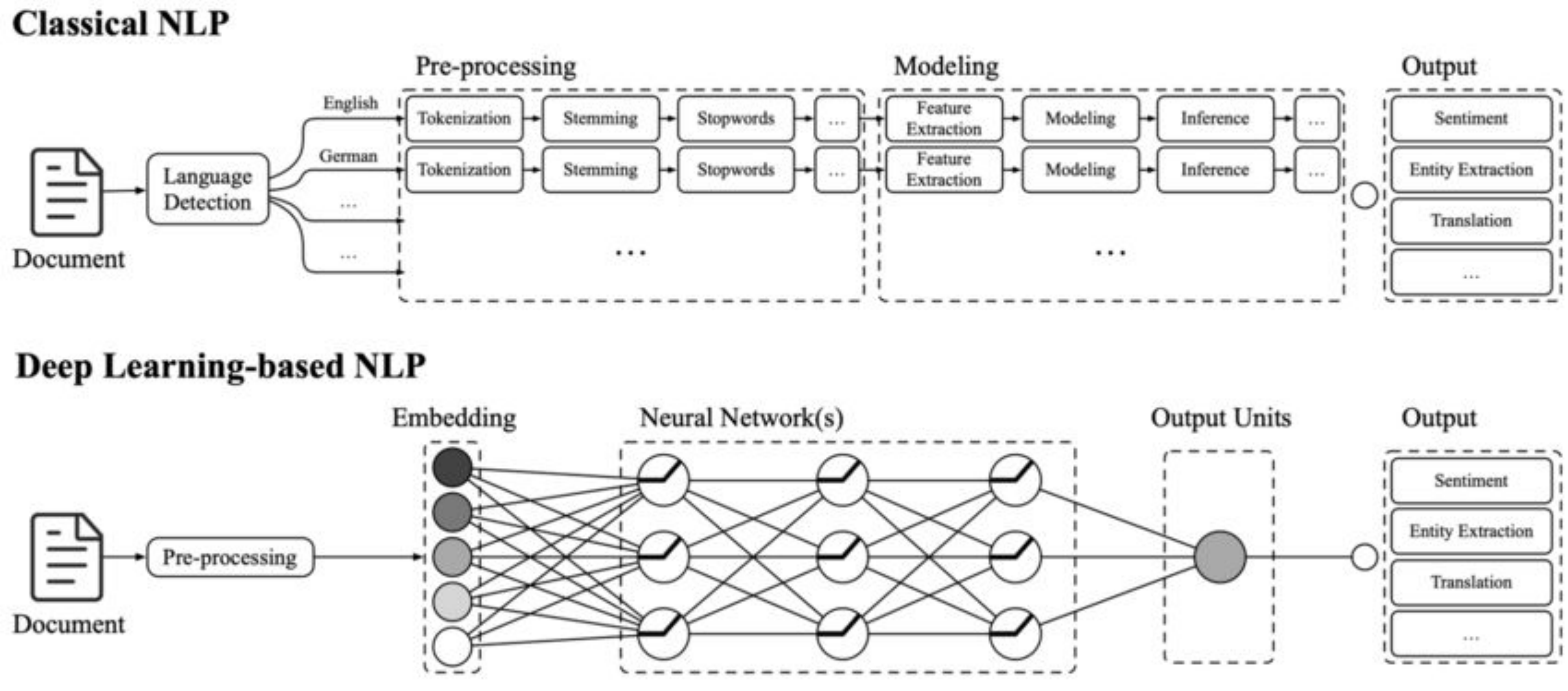
NLP with Deep Learning

Why deep learning?

Why deep learning?

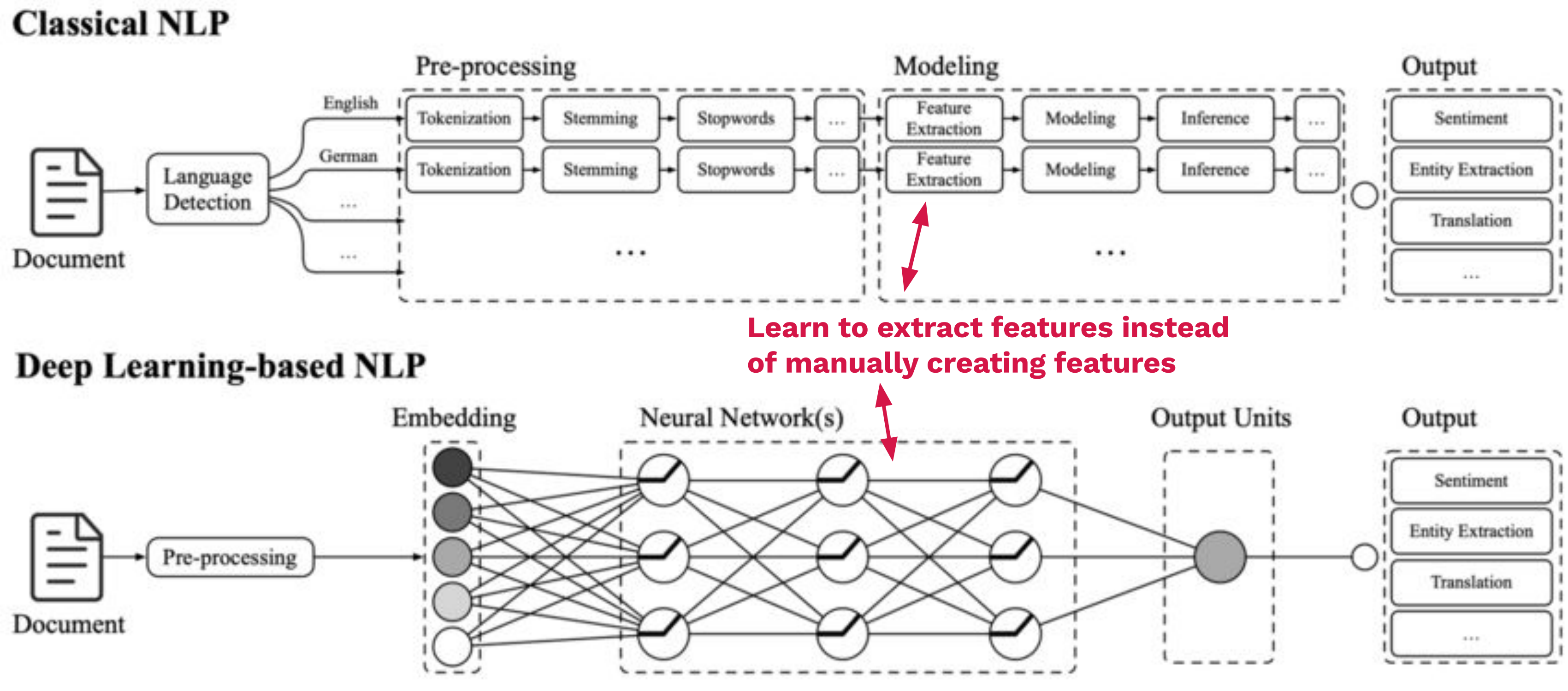
- Learn to extract features
- Data-driven learning
- End-to-end learning
- Scalable
- High Performance

Why deep learning?



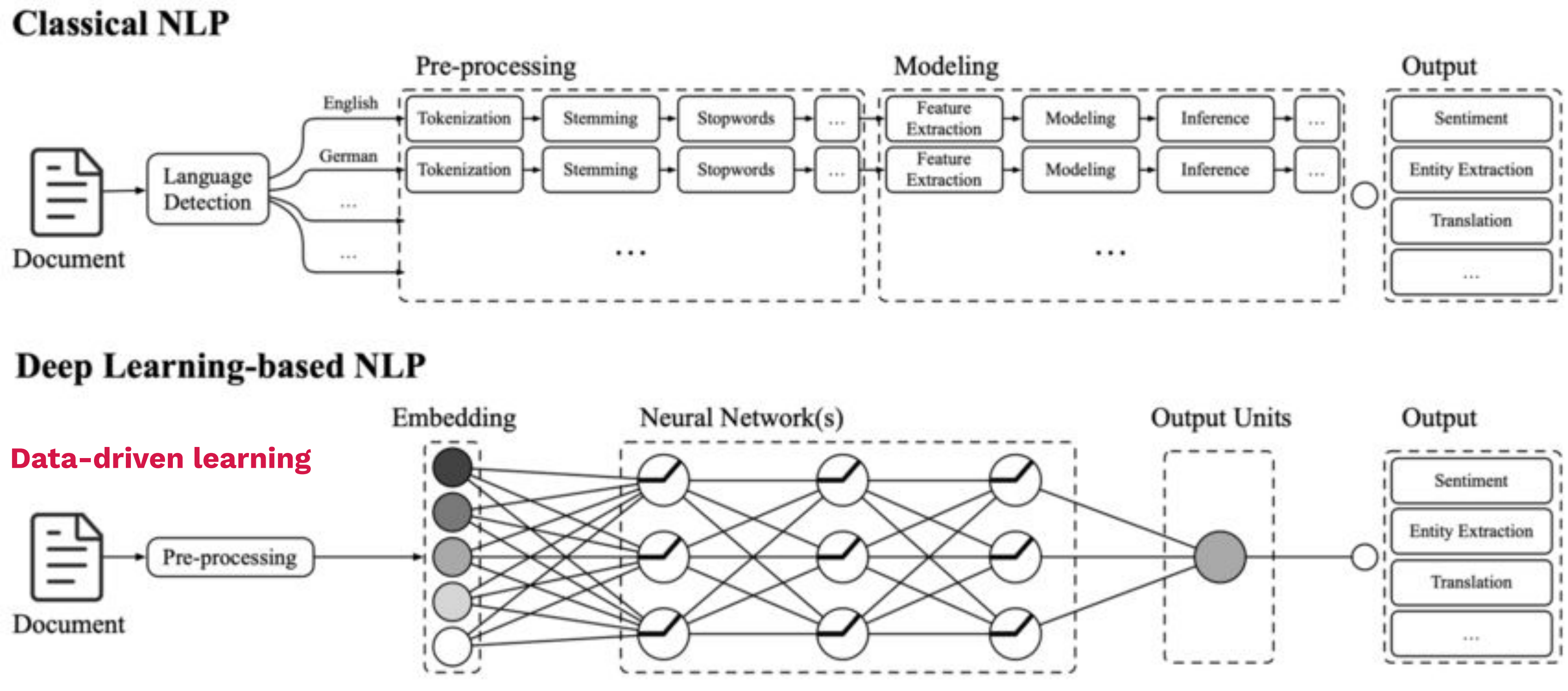
Source: Landolt, Severin, Thiemo Wambsganss, and Matthias Söllner. "A taxonomy for deep learning in natural language processing." HICSS. 2021.

Why deep learning?



Source: Landolt, Severin, Thiemo Wambsganss, and Matthias Söllner. "A taxonomy for deep learning in natural language processing." HICSS. 2021.

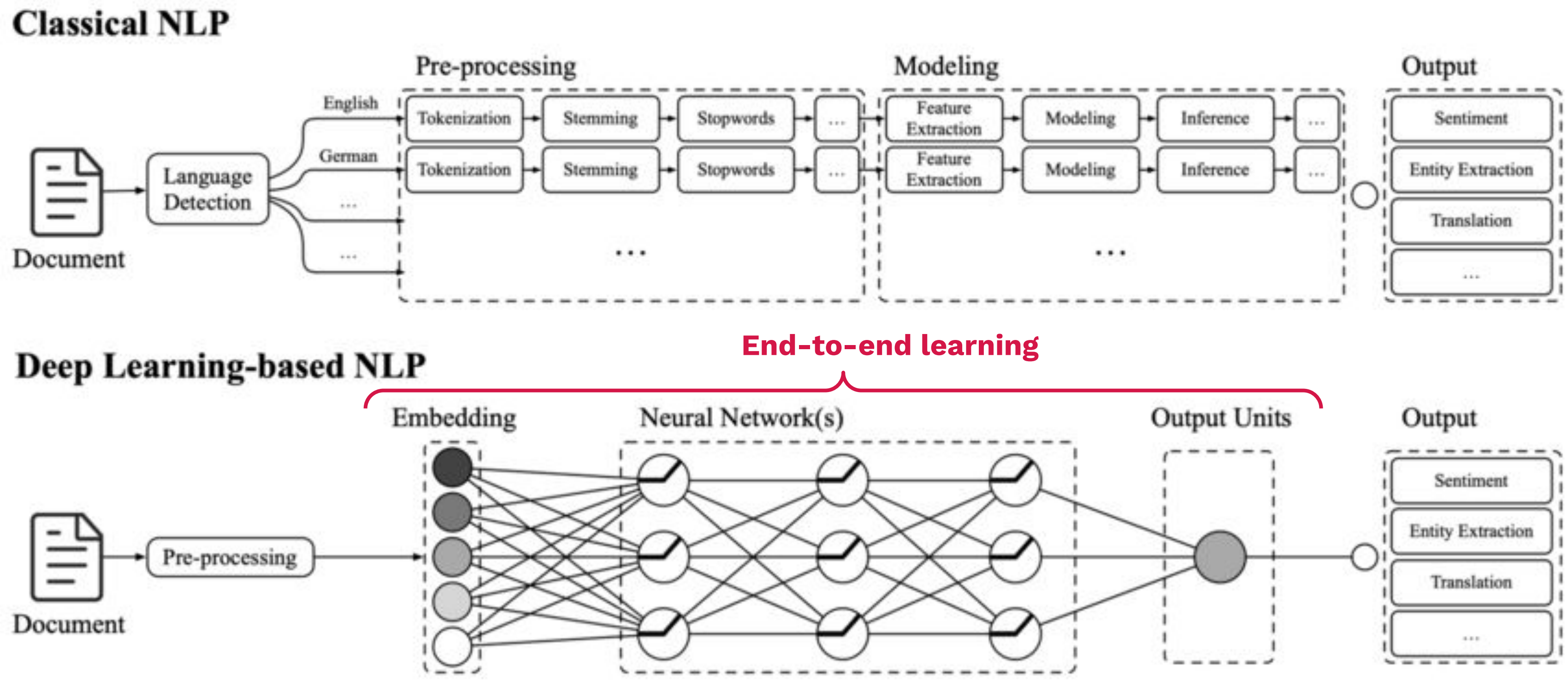
Why deep learning?



Data-driven learning

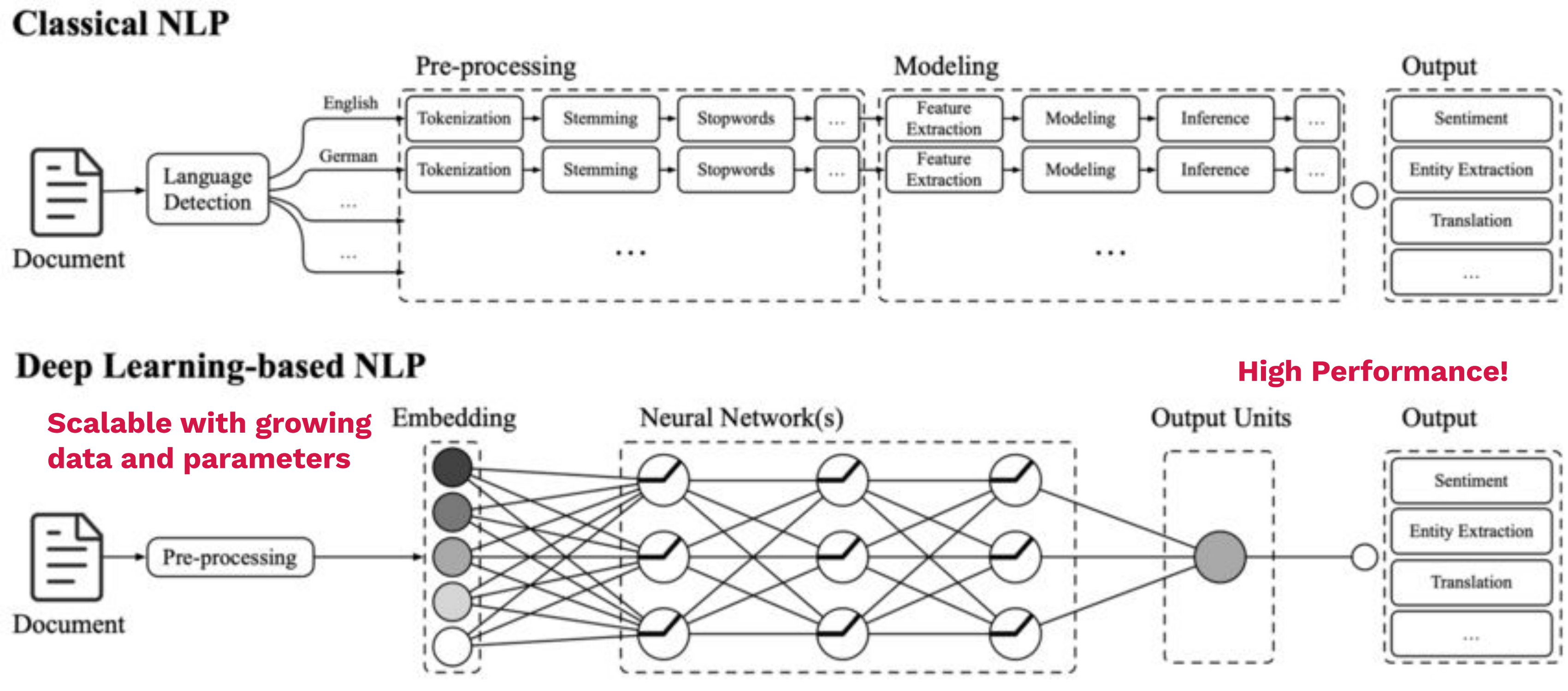
Source: Landolt, Severin, Thiemo Wambsganss, and Matthias Söllner. "A taxonomy for deep learning in natural language processing." HICSS. 2021.

Why deep learning?



Source: Landolt, Severin, Thiemo Wambsganss, and Matthias Söllner. "A taxonomy for deep learning in natural language processing." HICSS. 2021.

Why deep learning?



High Performance!

Scalable with growing data and parameters

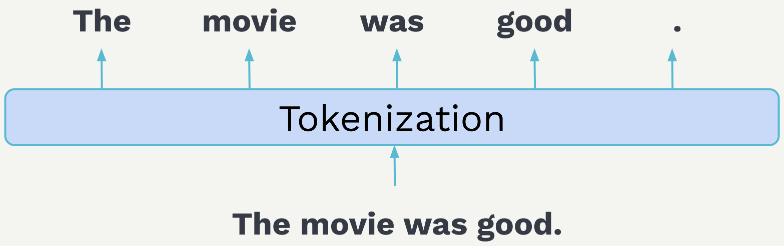
Source: Landolt, Severin, Thiemo Wambsganss, and Matthias Söllner. "A taxonomy for deep learning in natural language processing." HICSS. 2021.

NLU Pipeline

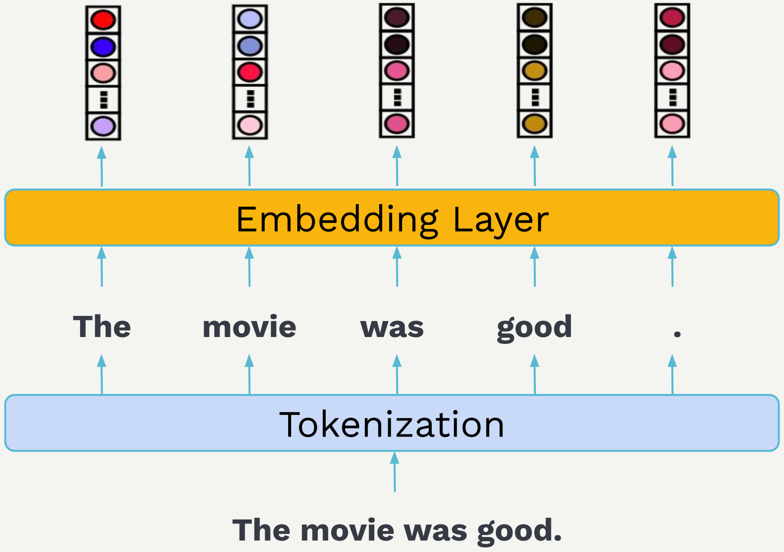
NLU Pipeline

The movie was good.

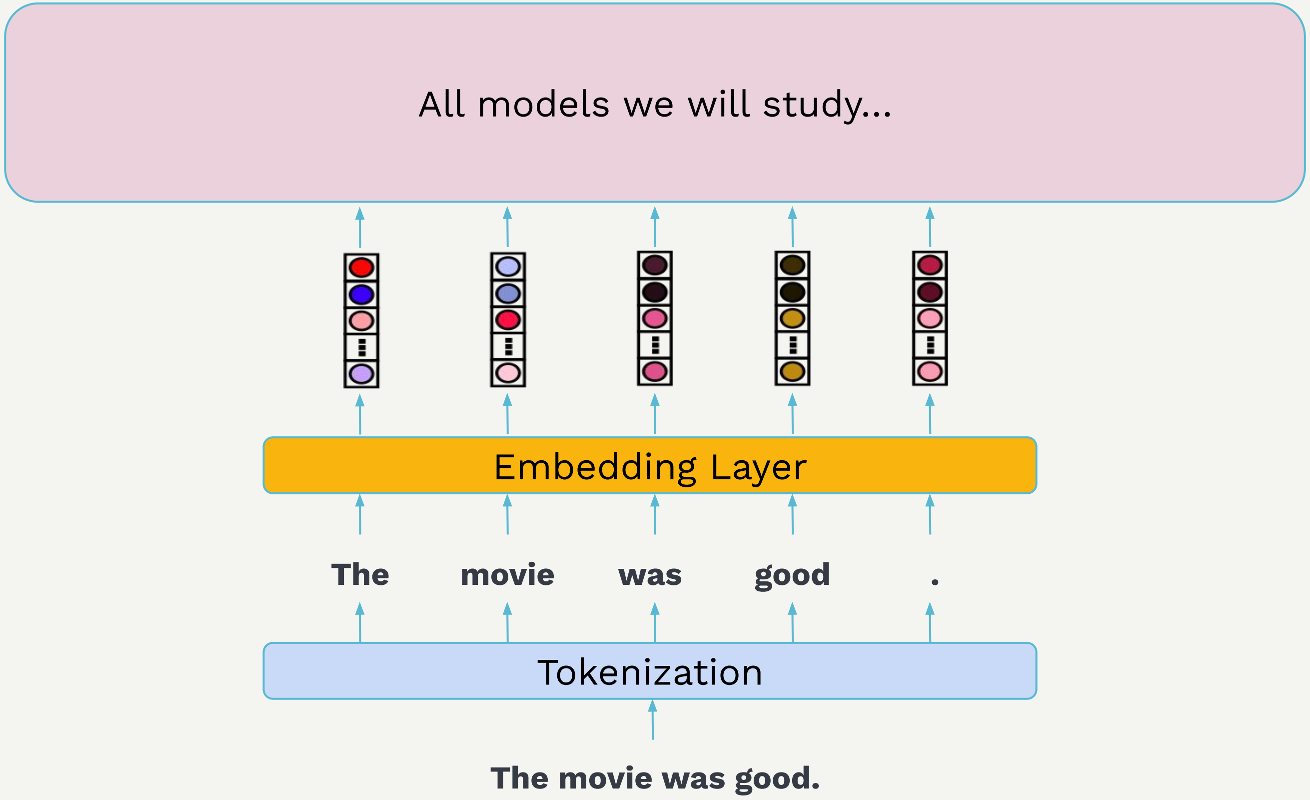
NLU Pipeline



NLU Pipeline



NLU Pipeline

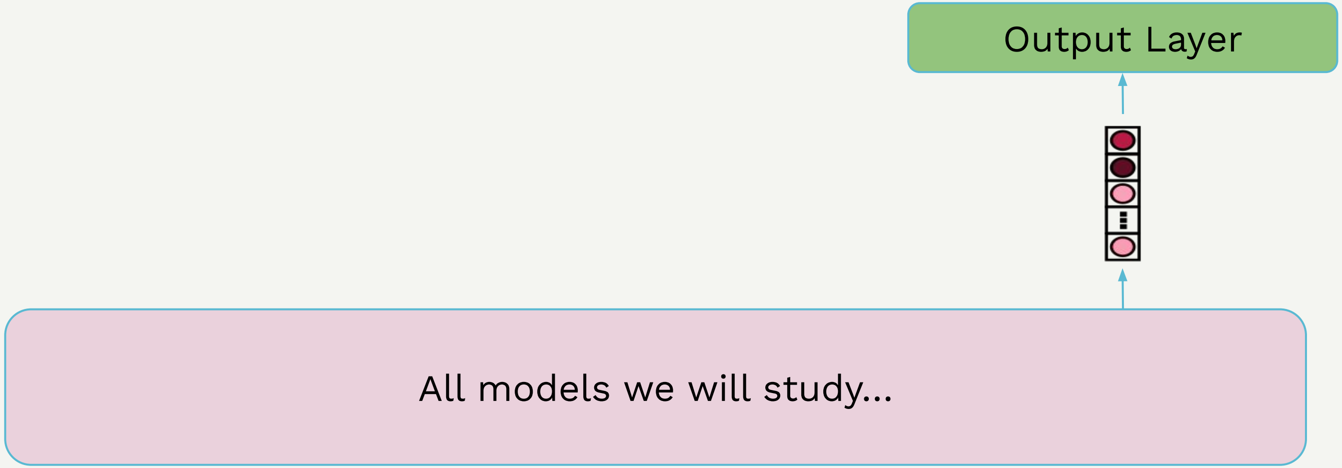


NLG Pipeline

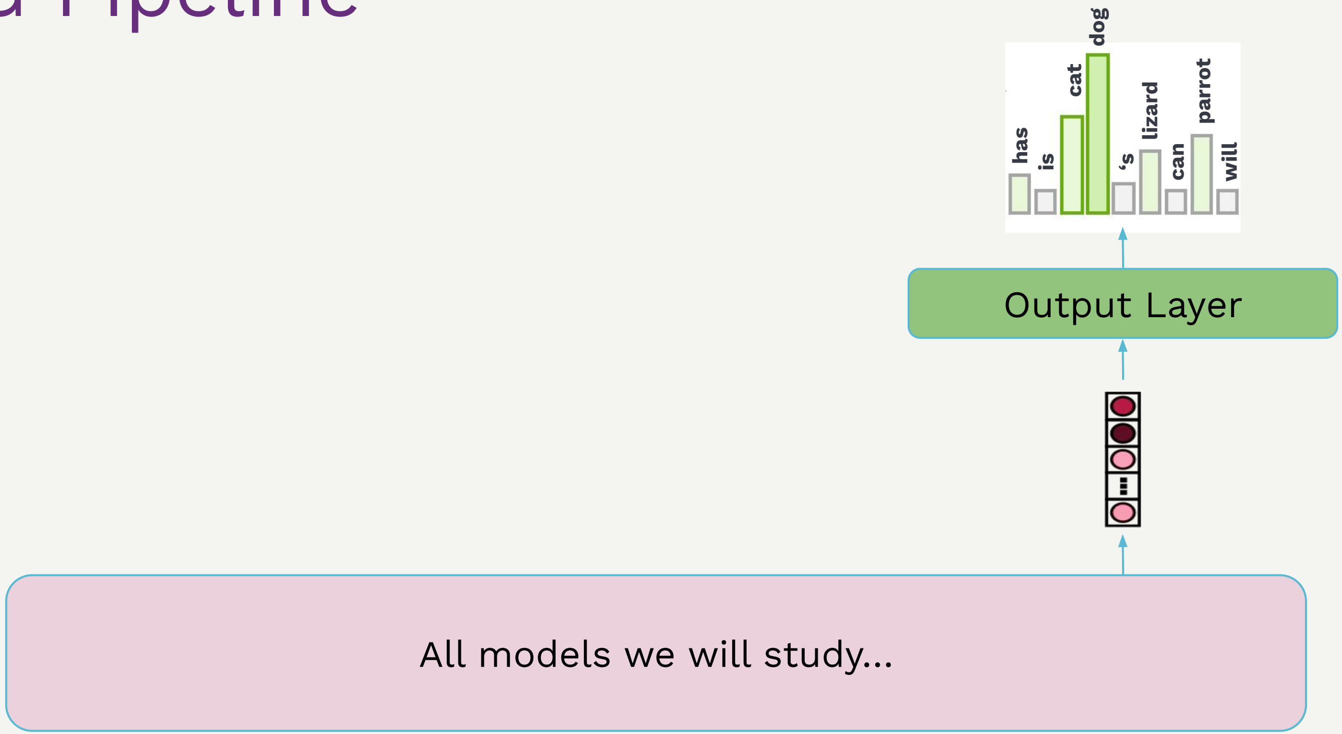
NLG Pipeline

All models we will study...

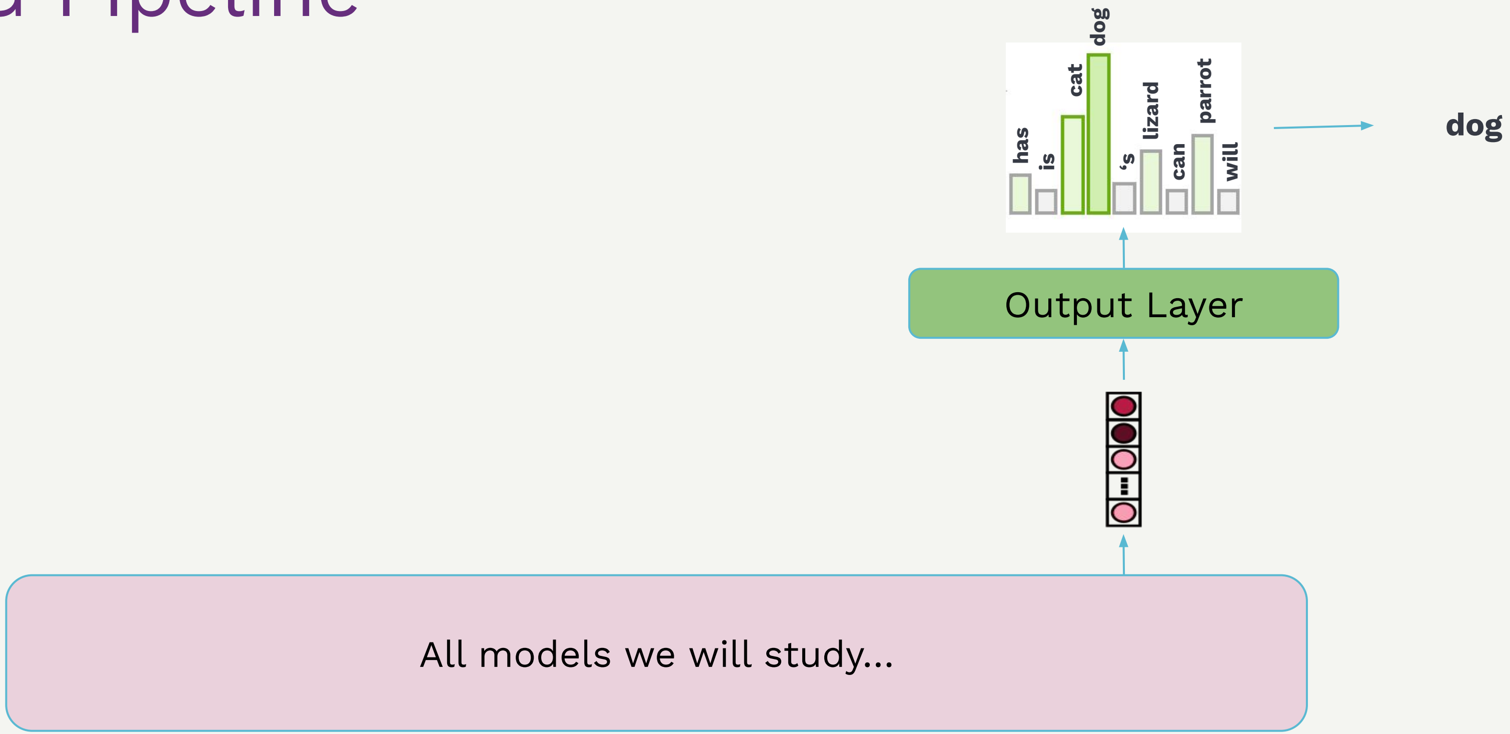
NLG Pipeline



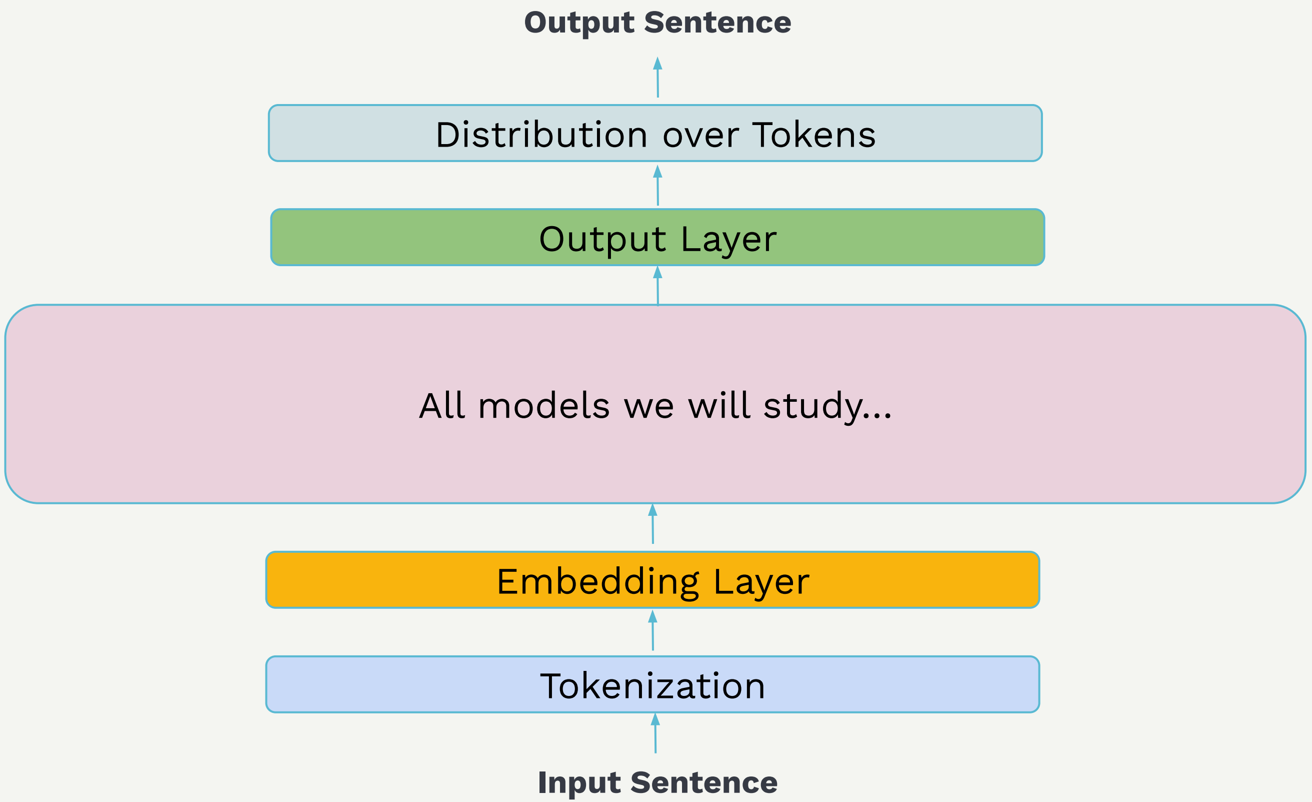
NLG Pipeline



NLG Pipeline



NLP Pipeline



Sneak Peek

- **RNNs, LSTMs, Attention, Transformers**
- **Large Language Models (LLMs) - ChatGPT, Claude, etc.**
- **Responsible NLP**

In the next two classes