07-280
AI & ML I

Natural Language Processing (NLP): N-gram Language Models

Instructors:
Pat Virtue & Nihar Shah

Natural Language Processing (NLP)

Quick NLP Intro

Tokenization

N-grams

Language Models (LMs)

LM Sampling

Natural Language Processing (NLP)

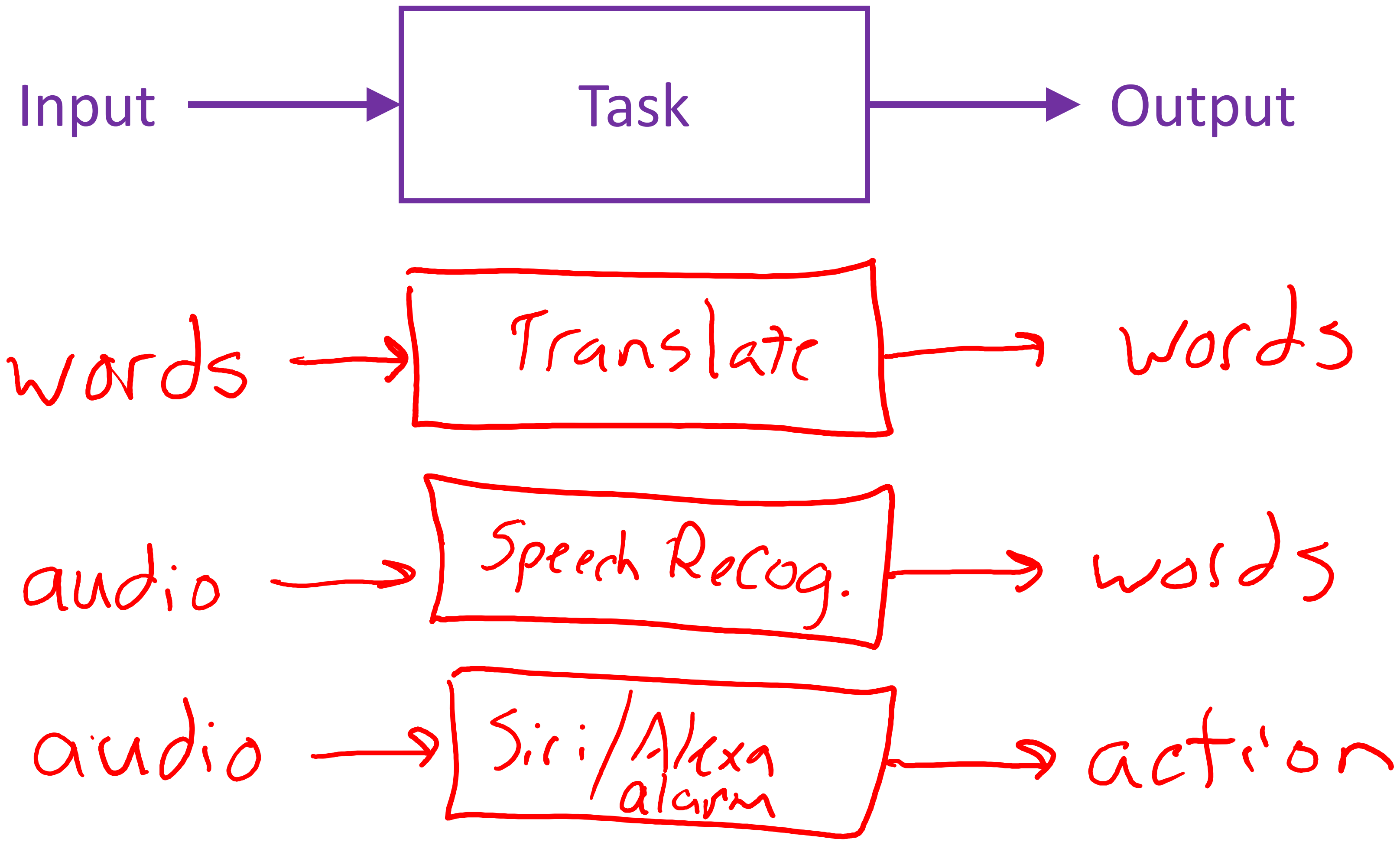
Language is Hard

Emphasis can drastically change meaning

I didn’t eat your dog

NLP Tasks Examples

How many different NLP Input/Output agents can you think of?

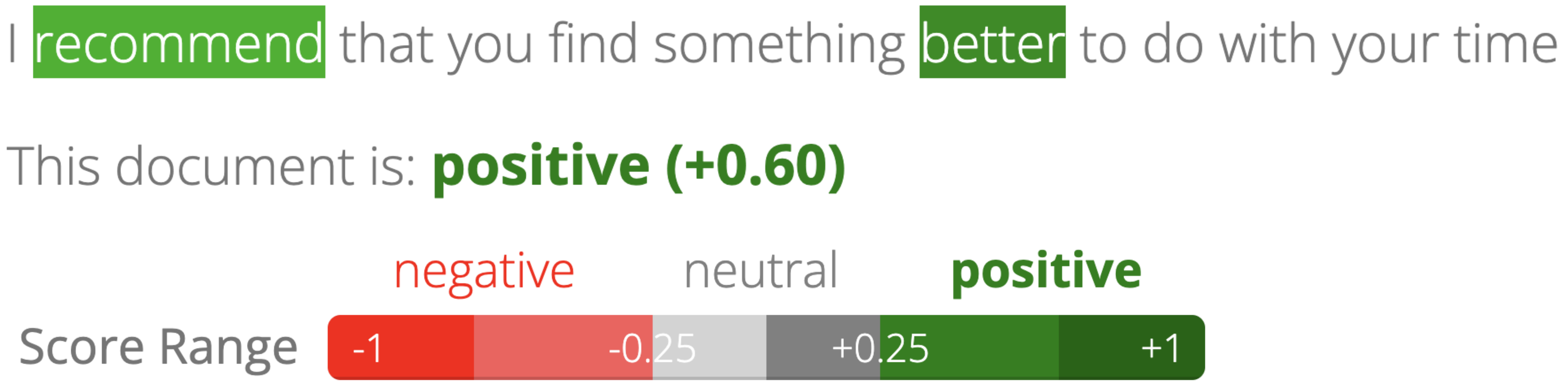


NLP Task: Sentiment Analysis

Sentiment analysis demo

https://text2data.com/Demo

"I recommend that you find something better to do with your time"



Vocab

Corpus

Tokenization/Tokens

Vocabulary

Context

Corpus:

```
The dog ran.
The dog ate.
The dog ran the zoo.
The cat ate the dog!
The cat ran the zoo.
```

→ Tokenizer →

(Simple) Tokenized Corpus:

```
[ 'the', 'dog', 'ran', '.', 'the',
  'dog', 'ate', '.', 'the', 'dog',
  'ran', 'the', 'zoo', '.', 'the',
  'cat', 'ate', 'the', 'dog', '!',
  'the', 'cat', 'ran', 'the',
  'zoo', '.' ]
```

Vocabulary:

```
[ '!',
  '.',
  'ate',
  'cat',
  'dog',
  'ran',
  'the',
  'zoo' ]
```

NLP Vocab

Corpus, Tokenizer

Corpus:

```
The dog ran.
The dog ate.
The dog ran the zoo.
The cat ate the dog!
The cat ran the zoo.
```

→ Tokenizer →

(Simple) Tokenized Corpus:

```
[ 'the', 'dog', 'ran', '.', 'the',
  'dog', 'ate', '.', 'the', 'dog',
  'ran', 'the', 'zoo', '.', 'the',
  'cat', 'ate', 'the', 'dog', '!',
  'the', 'cat', 'ran', 'the',
  'zoo', '.' ]
```

Vocabulary:

```
[ '!',
  '.',
  'ate',
  'cat',
  'dog',
  'ran',
  'the',
  'zoo' ]
```

Tokenization

Tokenization

Text encoding: token → index within vocabulary

What is our "vocabulary", i.e., what are our tokens?

- Characters
- Byte pair encoding
- Words and punctuation

Corpus

```
I am Sam. I am
Sam. Sam I am.
That Sam-I-am.
That Sam-I-am!
```

Tokenization

```
I am Sam. I am Sam. Sam I am.
That Sam-I-am. That Sam-I-am!
```

Text encoding: token → index within vocabulary

What is our "vocabulary", i.e., what are our tokens?

Character token vocabulary

```
 0:  !
 1:  -
 2:  .
 3:  I
 4:  S
 5:  T
 6:  a
 7:  h
 8:  m
 9:  t
10:  _  (space)
```

BPE token vocabulary

```
 0:  !        11:  Sam
 1:  -        12:  Th
 2:  .        13:  am
 3:  I        14:  _Sam
 4:  S        15:  _am
 5:  T
 6:  a
 7:  h
 8:  m
 9:  t
10:  _(space)
```

Word token vocabulary

```
0: !
1: .
2: am
3: I
4: Sam
5: That
6: Sam-I-am
```

Tokenization

```
I am Sam. I am Sam. Sam I am.
That Sam-I-am. That Sam-I-am!
```

Text encoding: token → index within vocabulary

Byte pair encoding (BPE)

Current token vocabulary

(Initialize to characters in corpus)

```
!    h
-    m
.    t
I    _ (space)
S
T
a
```

Pair frequencies

Count frequencies of all pairs of current tokens appearting together in corpus

```
_a: 3
am: 10
_S: 4
Sa: 5
_I: 1
Th: 2
ha: 2
at: 2
```

New tokens

Add most frequent pair to tokens

```
!    h
-    m
.    t
I    _ (space)
S    am
T
a
```

Tokenization

```
I am Sam. I am Sam. Sam I am.
That Sam-I-am. That Sam-I-am!
```

Text encoding: token → index within vocabulary

Byte pair encoding (BPE)

Current token vocabulary

(Initialize to characters in corpus)

```
!       h
-       m
.       t
I       _  (space)
S       am
T
a
```

Pair frequencies

Count frequencies of all pairs of current tokens appearting together in corpus

```
_S:   4
_I:   1
Th:   2
ha:   2
at:   2
_am:  3
Sam:  5
```

New tokens

Add most frequent pair to tokens

```
!       h
-       m
.       t
I       _  (space)
S       am
T       Sam
a
```

Tokenization

```
I am Sam. I am Sam. Sam I am.
That Sam-I-am. That Sam-I-am!
```

Text encoding: token → index within vocabulary

What is our "vocabulary", i.e., what are our tokens?

Character token vocabulary

```
 0:  !
 1:  -
 2:  .
 3:  I
 4:  S
 5:  T
 6:  a
 7:  h
 8:  m
 9:  t
10:  _  (space)
```

BPE token vocabulary

```
 0:  !       11:  Sam
 1:  -       12:  Th
 2:  .       13:  am
 3:  I       14:  _Sam
 4:  S       15:  _am
 5:  T
 6:  a
 7:  h
 8:  m
 9:  t
10:  _(space)
```

Word token vocabulary

```
0: !
1: .
2: am
3: I
4: Sam
5: That
6: Sam-I-am
```

Language Models

Human Language Model

[Wheel of Fortune]

What is a language model (LM)?

LM Applications: Speech Recognition

Example: Speech Recognition

“artificial ……….”

Find most probable next word given “artificial” and the audio for second word.

LM Applications: Speech Recognition

Example: Speech Recognition

“artificial ……….”

Find most probable next word given “artificial” and the audio for second word.

Which second word gives the highest probability?

$P(\textbf{limb} \mid \text{artificial, audio})$

$P(\textbf{intelligence} \mid \text{artificial, audio})$

$P(\textbf{flavoring} \mid \text{artificial, audio})$

Break down problem

n-gram probability * audio probability

$P(\textbf{limb} \mid \text{artificial}) * P(\text{audio} \mid \textbf{limb})$

$P(\textbf{intelligence} \mid \text{artificial}) * P(\text{audio} \mid \textbf{intelligence})$

$P(\textbf{flavoring} \mid \text{artificial}) * P(\text{audio} \mid \textbf{flavoring})$

LM Applications: Text Generation

Given a LM that can model the joint distribution any 10-token sequence in the vocabulary, $p(w_1, w_2, \ldots, w_{10})$, how can we get the probability of a 6th token, given 5 previous tokens

Language Models

N-grams

N-gram Worksheet

[Worksheet linked on website]

N-gram Worksheet

Corpus:

```
The dog ran.
The dog ate.
The dog ran the zoo.
The cat ate the dog!
The cat ran the zoo.
```

(Simple) Tokenized Corpus:

```
[ 'the', 'dog', 'ran', '.', 'the',
  'dog', 'ate', '.', 'the', 'dog',
  'ran', 'the', 'zoo', '.', 'the',
  'cat', 'ate', 'the', 'dog', '!',
  'the', 'cat', 'ran', 'the',
  'zoo', '.' ]
```

Vocabulary:

```
[ '!',
  '.',
  'ate',
  'cat',
  'dog',
  'ran',
  'the',
  'zoo' ]
```

Compute the unigram probabilities given the tokenized corpus above

x_t	!	.	ate	cat	dog	ran	the	zoo
$p(x_t)$								

Compute the bigram probabilities given the tokenized corpus above

$p(x_{t+1}\|x_t)$	x_{t+1}	x_t
	!	!
	.	!
	ate	!
	cat	!
	dog	!
	ran	!
	the	!
	zoo	!

$p(x_{t+1}\|x_t)$	x_{t+1}	x_t
	!	.
	.	.
	ate	.
	cat	.
	dog	.
	ran	.
	the	.
	zoo	.

$p(x_{t+1}\|x_t)$	x_{t+1}	x_t
	!	ate
	.	ate
	ate	ate
	cat	ate
	dog	ate
	ran	ate
	the	ate
	zoo	ate

$p(x_{t+1}\|x_t)$	x_{t+1}	x_t
	!	cat
	.	cat
	ate	cat
	cat	cat
	dog	cat
	ran	cat
	the	cat
	zoo	cat

$p(x_{t+1}\|x_t)$	x_{t+1}	x_t
	!	dog
	.	dog
	ate	dog
	cat	dog
	dog	dog
	ran	dog
	the	dog
	zoo	dog

$p(x_{t+1}\|x_t)$	x_{t+1}	x_t
	!	ran
	.	ran
	ate	ran
	cat	ran
	dog	ran
	ran	ran
	the	ran
	zoo	ran

$p(x_{t+1}\|x_t)$	x_{t+1}	x_t
	!	the
	.	the
	ate	the
	cat	the
	dog	the
	ran	the
	the	the
	zoo	the

$p(x_{t+1}\|x_t)$	x_{t+1}	x_t
	!	zoo
	.	zoo
	ate	zoo
	cat	zoo
	dog	zoo
	ran	zoo
	the	zoo
	zoo	zoo