Chapter 5 Natural Language Processing

COMP 6721 Introduction of AI

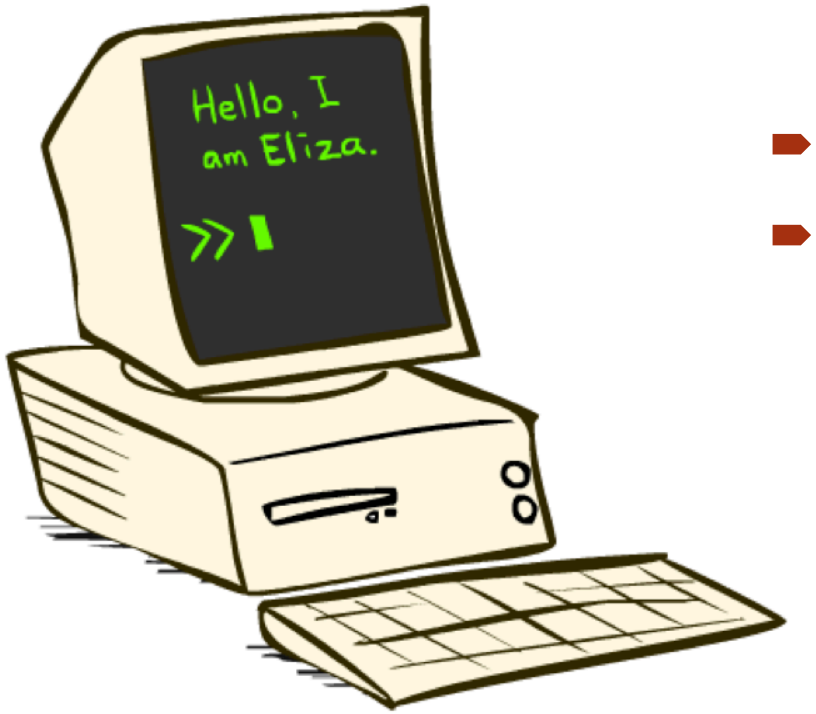
Russell & Norvig – Chapter 23.1 + 23.2 + 23.3

Languages

- Artificial
  - Smaller vocabulary
  - Simple syntactic structures
  - Non-ambiguous
  - Not tolerant to errors (ex. Syntax error)
- Natural
  - Large and open vocabulary (new words everyday)
  - Complex syntactic structures
  - Very ambiguous
  - Robust (ex. forgot a comma, a word… still OK)

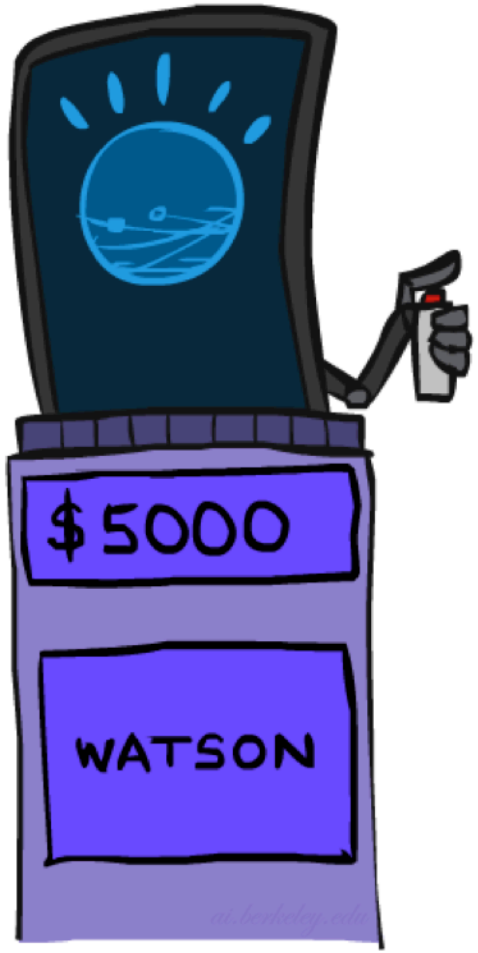
IBM's Watson

IBM's Watson



- A "psychotherapist" agent (Weizenbaum, ~1964)
- Led to a long line of chatterbots
- How does it work:
  - Trivial NLP: string match and substitution
  - Trivial knowledge: tiny script / response database
  - Example: matching "I remember __" results in "Do you often think of __"?
- Can fool some people some of the time?

IBM's Watson



"a camel is a horse designed by"

About 629,000 results (0.27 seconds)

Wiktionary
['wɪkʃənrɪ] *n.*,
a wiki-based Open Content dictionary

Entry Discussion Read Edit History Search

Log in / create account

a camel is a horse designed by a committee

The Phrase Finder

> Discussion Forum

Google™ Custom Search Search

A camel is a horse designed by committee

Posted by Ruben P. Mendez on April 16, 2004

Does anyone know the origin of this maxim? I heard it way back at the United Nations, which is chockfull of committees. It may have originated there, but I'd like an authoritative explanation. Thanks

- Re: A camel is a horse designed by committee **SR** *16/April/04*
  - Re: A camel is a horse designed by committee **Henry** *18/April/04*

If **a camel is a horse designed by** committee then what's this ...

If **a camel is a horse designed by** committee then what's this contemporary Routemaster?

What is in Watson

- A question-answering system (IBM, 2011)
- Designed for the game of Jeopardy
- How does it work:
  - Sophisticated NLP: deep analysis of questions, noisy matching of questions to potential answers
  - Lots of data: onboard storage contains a huge collection of documents (e.g. Wikipedia, etc.), exploits redundancy
  - Lots of computation: 90+ servers
- Can beat all of the people all of the time?

https://www.youtube.com/watch?v=WFR3lOm_xhE

Problems with Dictionary Lookups

顶部	/**top**/roof/
顶端	/summit/peak/**top**/apex/
顶头	/coming directly towards one/**top**/end/
盖	/lid/**top**/cover/canopy/build/Gai/
盖帽	/surpass/**top**/
极	/extremely/pole/utmost/**top**/collect/receive/
尖峰	/peak/**top**/
面	/fade/side/surface/aspect/**top**/face/flour/
摘心	/**top**/topping/

Learning to Translate

CLASSIC SOUPS

			Sm.	Lg.
清燉雞湯	57.	House Chicken Soup (Chicken, Celery, Potato, Onion, Carrot)	1.50	2.75
雞飯湯	58.	Chicken Rice Soup	1.85	3.25
雞麵湯	59.	Chicken Noodle Soup	1.85	3.25
廣東雲吞	60.	Cantonese Wonton Soup	1.50	2.75
蕃茄蛋湯	61.	Tomato Clear Egg Drop Soup	1.65	2.95
雲吞湯	62.	Regular Wonton Soup	1.10	2.10
酸辣湯	63.	Hot & Sour Soup	1.10	2.10
蛋花湯	64.	Egg Drop Soup	1.10	2.10
雲蛋湯	65.	Egg Drop Wonton Mix	1.10	2.10
豆腐菜湯	66.	Tofu Vegetable Soup	NA	3.50
雞玉米湯	67.	Chicken Corn Cream Soup	NA	3.50
蟹肉玉米湯	68.	Crab Meat Corn Cream Soup	NA	3.50
海鮮湯	69.	Seafood Soup	NA	3.50

Example from Adam Lopez

What is NLP

- "Natural Language Processing (NLP) is a subfield of computer science, information engineering, and artificial intelligence concerned with the **interactions between computers and human (natural) languages**, in particular how to program computers to process and analyze large amounts of natural language data."

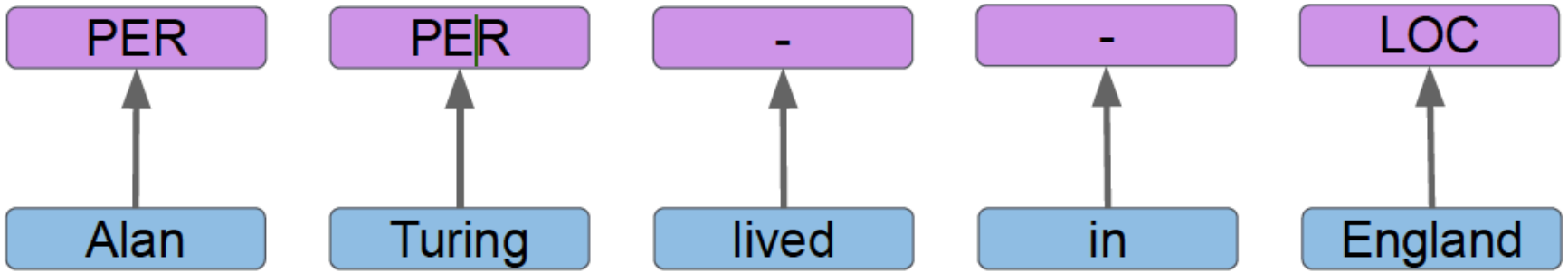
https://en.wikipedia.org/wiki/Natural_language_processing

Some NLP Tasks

- Classification (word-level)
  - **Named Entity Recognition**
  - Part of Speech Tagging
  - Extractive Question Answering
- Classification (sentence-level)
  - Sentiment Analysis
  - Spam Filters
- Classification (sentence pair-level)
  - Entailment
  - Sentence similarity
- Generative
  - Machine Translation
  - Abstractive Text Summarization
  - Abstractive Question Answering

Named Entity Recognition

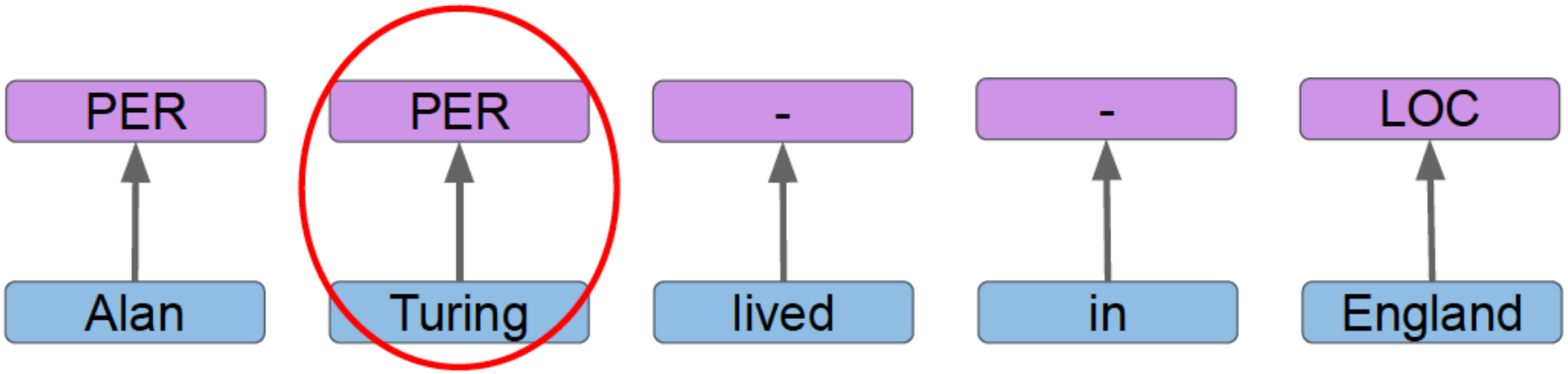
- Goal: assign a label to each word, describing its entity type.
- E.g., let's assume that the list of labels is:
  - "-" (not a named entity),
  - "PER" (person),
  - "LOC" (location),
  - "TIME" (time).



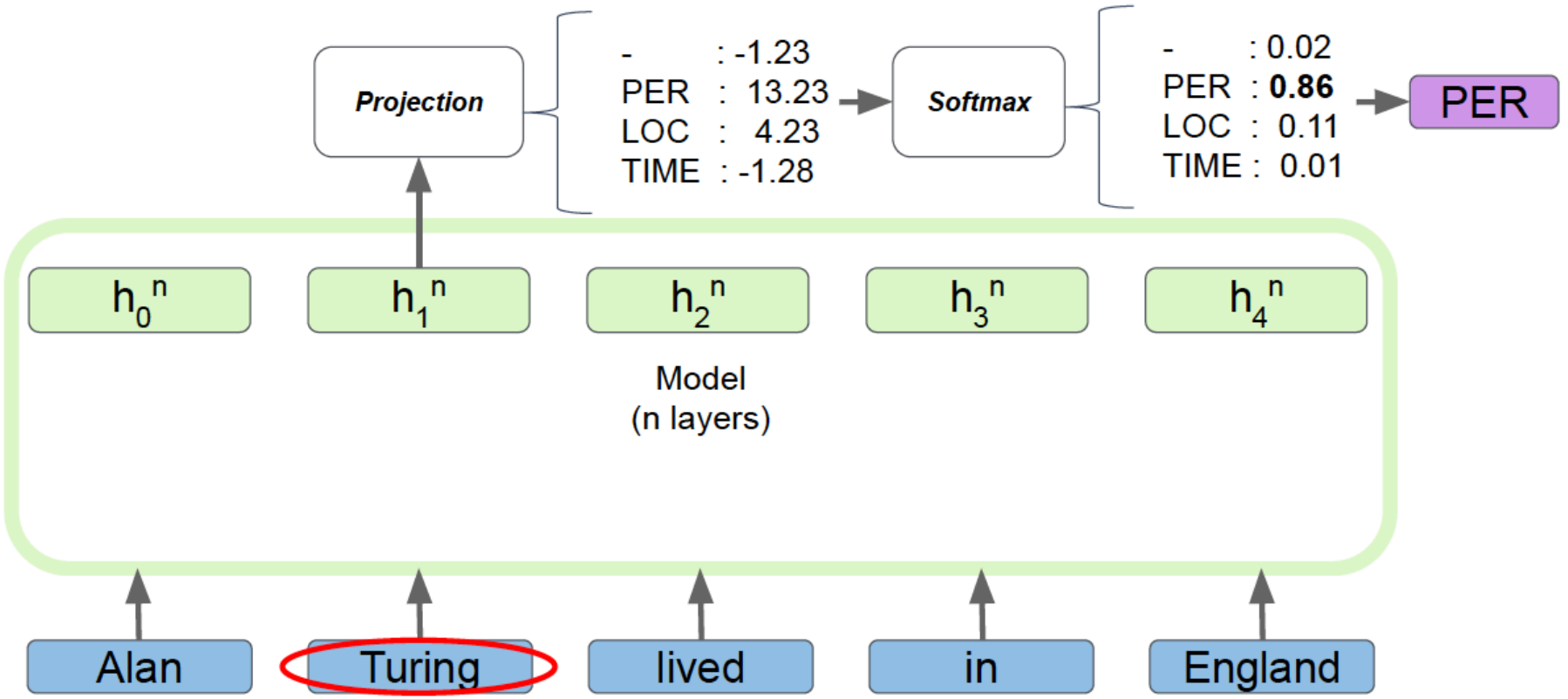
Named Entity Recognition

- Goal: assign a label to each word, describing its entity type.

  E.g., let's assume that the list of labels is: "-", "PER", "LOC", "TIME".
- Let's focus our example on only one step (e.g., the word "Turing").



Named Entity Recognition

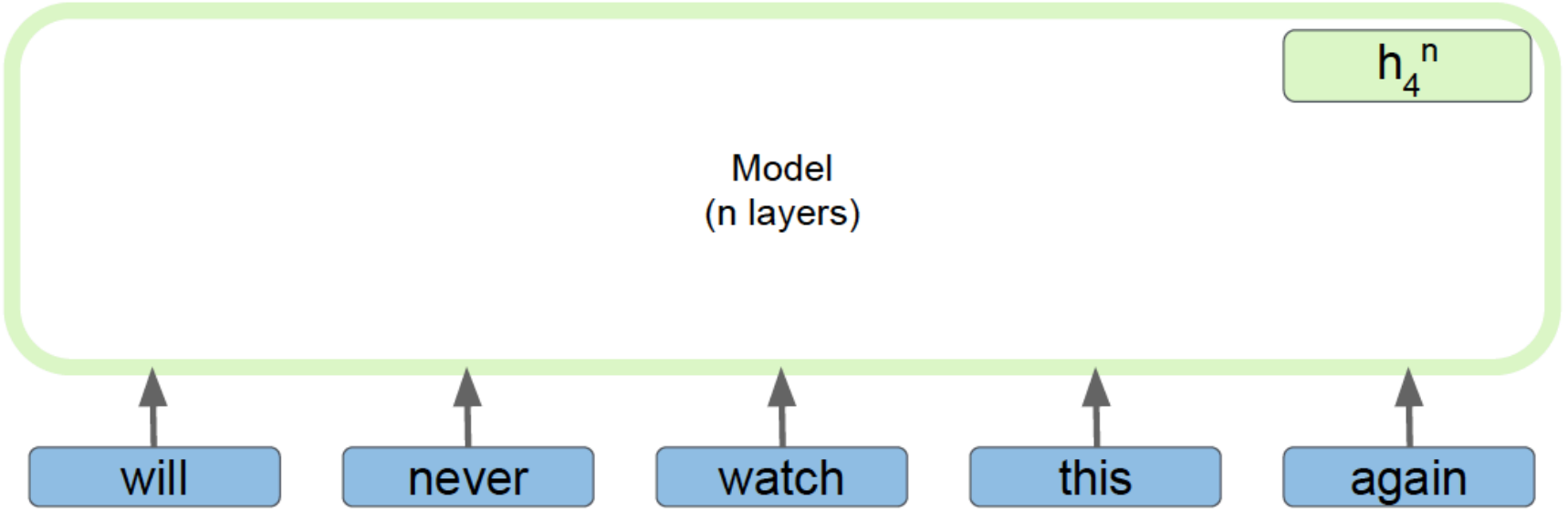


Some NLP Tasks

- Classification (word-level)
  - **Named Entity Recognition**
  - Part of Speech Tagging
  - Extractive Question Answering
- Classification (sentence-level)
  - **Sentiment Analysis**
  - Spam Filters
- Classification (sentence pair-level)
  - Entailment
  - Sentence similarity
- Generative
  - Machine Translation
  - Abstractive Text Summarization
  - Abstractive Question Answering

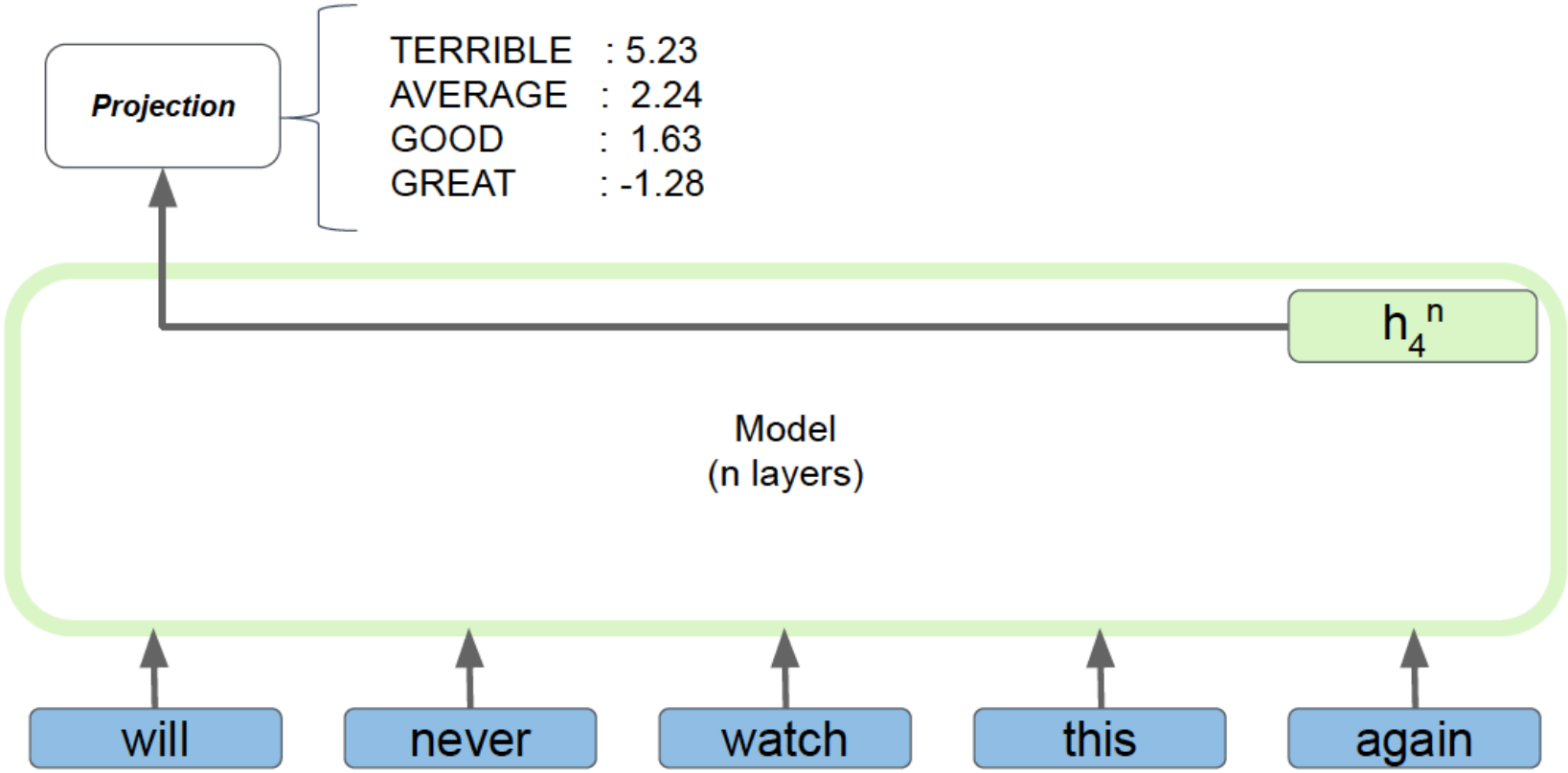
Sentiment Analysis

- Goal: assign one label to a **sentence** describing the sentiment that it conveys.
- Note that we only need the last model output (h_4^n).



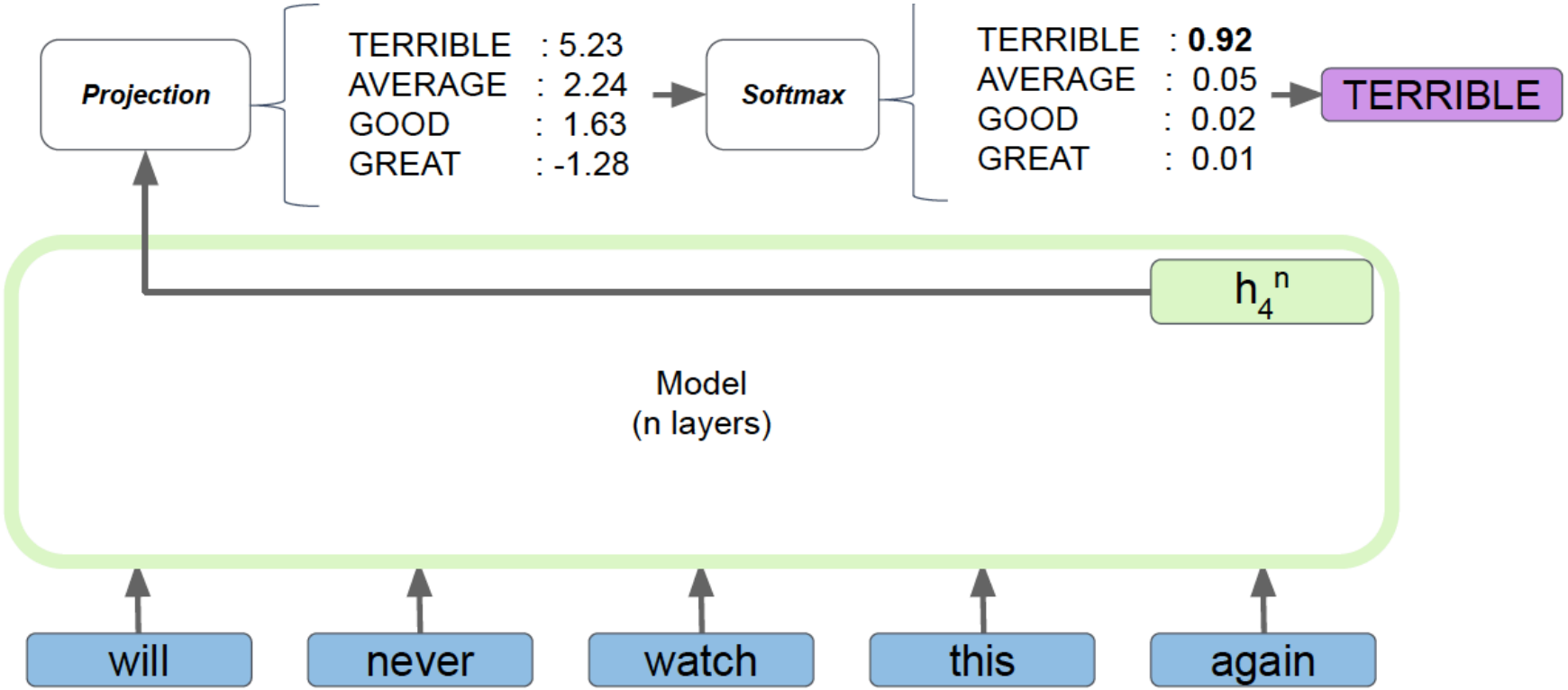
Sentiment Analysis

- Goal: assign one label to a **sentence** describing the sentiment that it conveys.



Sentiment Analysis

- Goal: assign one label to a **sentence** describing the sentiment that it conveys.



Some NLP Tasks

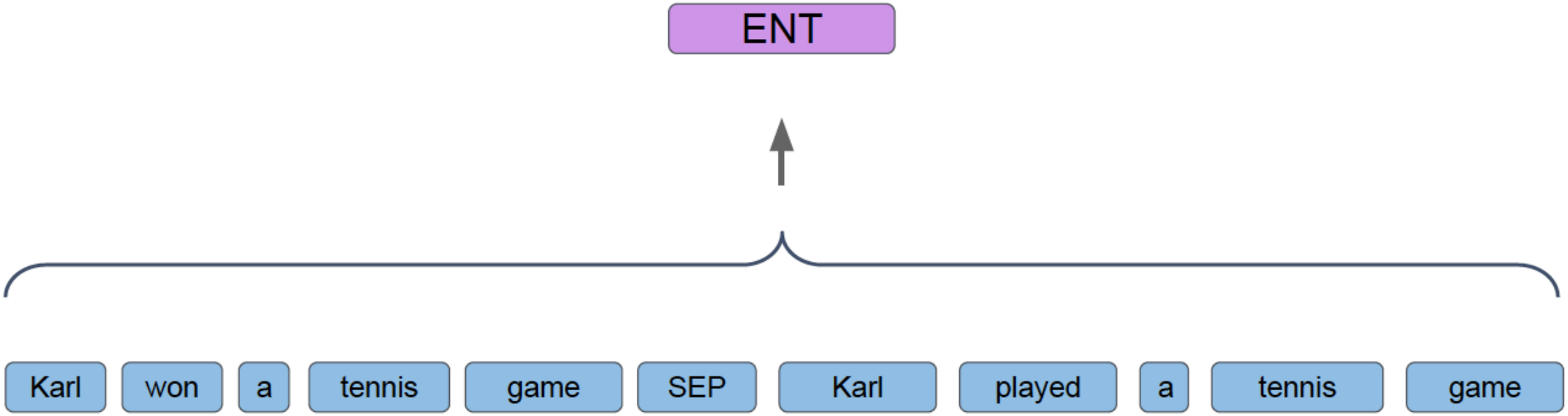
- Classification (word-level)
  - **Named Entity Recognition**
  - Part of Speech Tagging
  - Extractive Question Answering
- Classification (sentence-level)
  - **Sentiment Analysis**
  - Spam Filters
- Classification (sentence pair-level)
  - **Entailment**
  - Sentence similarity
- Generative
  - Machine Translation
  - Abstractive Text Summarization
  - Abstractive Question Answering

Entailment

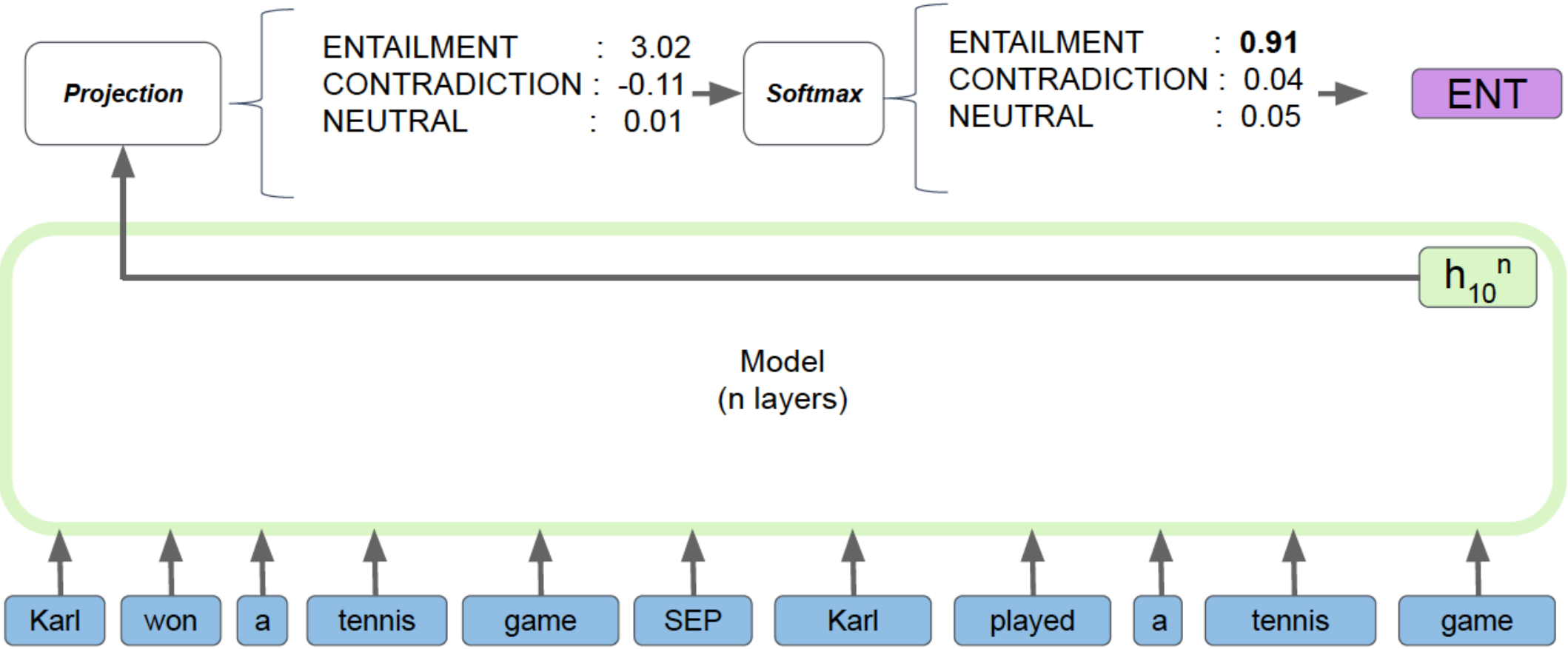
- Task: given two sentences, does the first one entail the second one? E.g.,
  - (Input) Sentence #1: "Karl won a tennis game"
  - (Input) Sentence #2: "Karl played a tennis game"
  - Target: "entailment"
- There are three possible labels:
  - "entailment"
  - "contradiction"
  - "neutral"

Entailment

- SEP = separator indicating the end of the first sentence.



Entailment


Projection
ENTAILMENT : 3.02
CONTRADICTION : -0.11
NEUTRAL : 0.01
Softmax
ENTAILMENT : 0.91
CONTRADICTION : 0.04
NEUTRAL : 0.05
ENT
h_{10}^{n}
Model
(n layers)
Karl
won
a
tennis
game
SEP
Karl
played
a
tennis
game

Some NLP Tasks

- Classification (word-level)
  - **Named Entity Recognition**
  - Part of Speech Tagging
  - Extractive Question Answering
- Classification (sentence-level)
  - **Sentiment Analysis**
  - Spam Filters
- Classification (sentence pair-level)
  - **Entailment**
  - Sentence similarity
- Generative
  - **Machine Translation**
  - Abstractive Text Summarization
  - Abstractive Question Answering

Machine Translation

- Task: given a sentence, translate it into another language. E.g.,
  - Input: “Je suis malade”
  - Target: “I am sick”
- Machine translation requires a sequence-to-sequence (encoder plus decoder) model.
  - The encoder parses the input.
  - The decoder produces the output (using an autoregressive approach).
  - Attention (between encoder and decoder) greatly improves results.

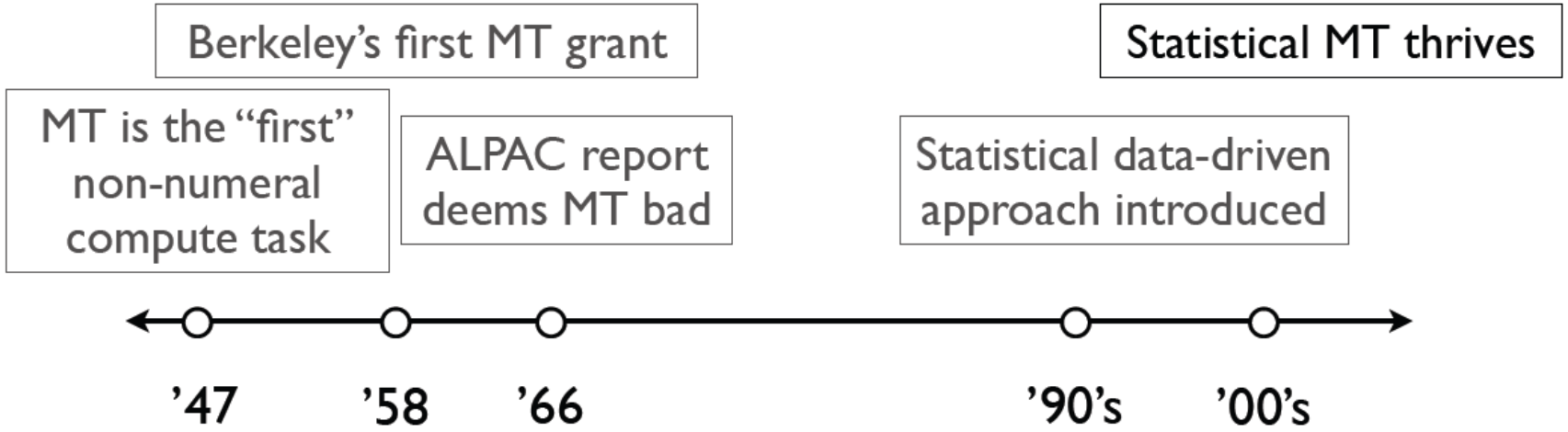
Machine Translation: 60 Years in 60s

When I look at an article in Russian, I say: "This is really written in English, but it has been coded in some strange symbols. I will now proceed to decode."

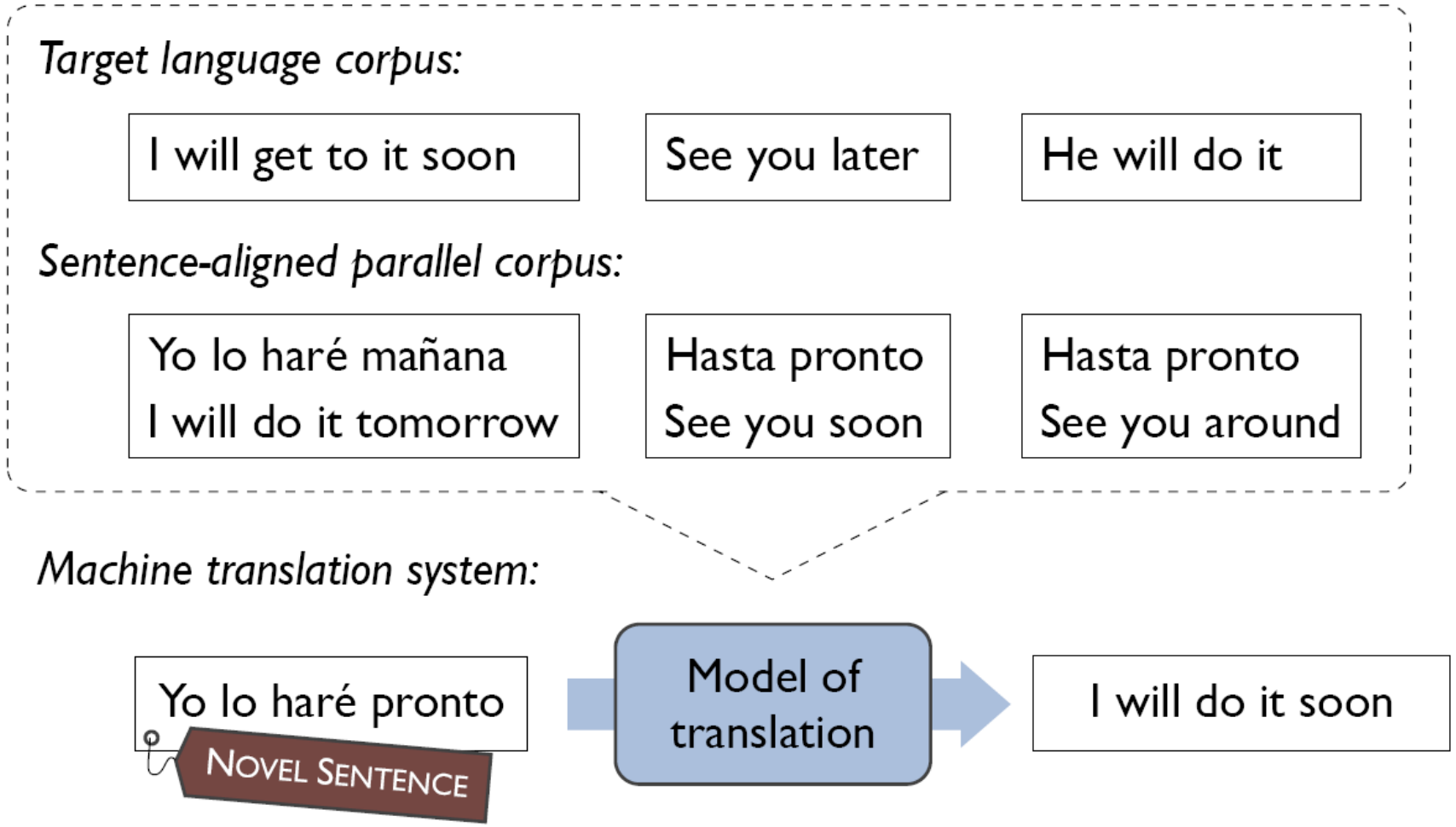
Warren Weaver

"Machine Translation" presumably means going by algorithm from machine-readable source text to useful target text... In this context, there has been no machine translation...

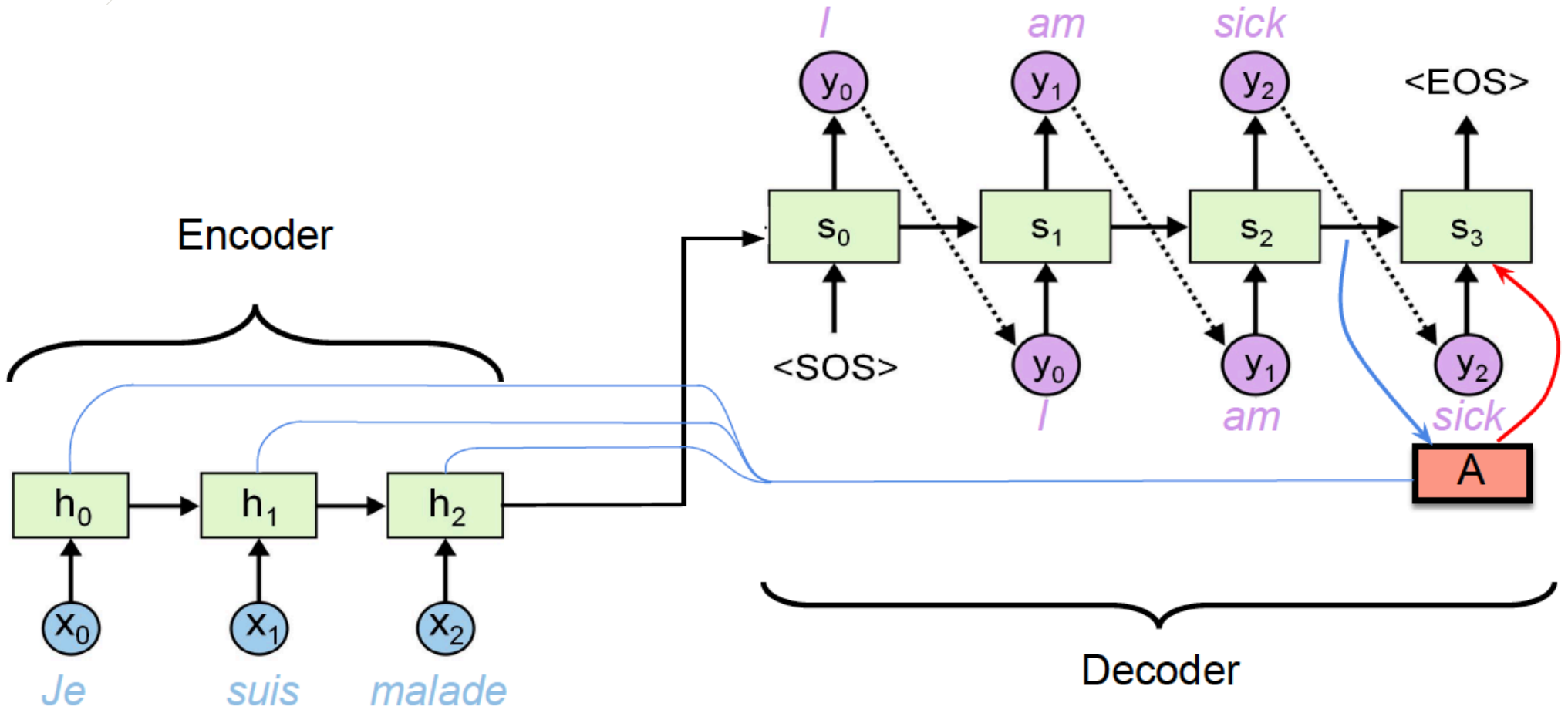
John Pierce



Data-Driven Machine Translation



Machine Translation



Where we are today

- mostly solved.

Spam detection

Let's go to Agra! ✓

Buy V1AGRA ... ✗

Part-of-speech (POS) tagging

ADJ	ADJ	NOUN	VERB	ADV
Colorless	green	ideas	sleep	furiously.

Named entity recognition (NER)

PERSON	ORG	LOC
Einstein met with UN officials in Princeton		

Making good progress.

Sentiment analysis

Best roast chicken in San Francisco!

The waiter ignored us for 20 minutes.

Coreference resolution

Carter told Mubarak he shouldn't run again.

Word sense disambiguation (WSD)

I need new batteries for my *mouse*.

Parsing

I can see Alcatraz from the window!

Machine translation (MT)

第13届上海国际电影节开幕...

The 13th Shanghai International Film Festival...

Information extraction (IE)

You're invited to our dinner party, Friday May 27 at 8:30

Party May 27 add

Good progress by DL

Question answering (QA)

Q. How effective is ibuprofen in reducing fever in patients with acute febrile illness?

Paraphrase

XYZ acquired ABC yesterday

ABC has been taken over by XYZ

Summarization

The Dow Jones is up

The S&P500 jumped

Housing prices rose

→ Economy is good

Dialog

Where is Citizen Kane playing in SF?

Castro Theatre at 7:30. Do you want a ticket?

Why NLP is hard

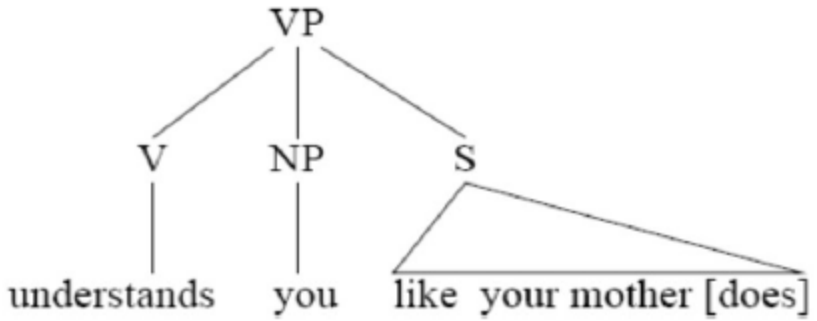
"At last, a computer that understands you like your mother"

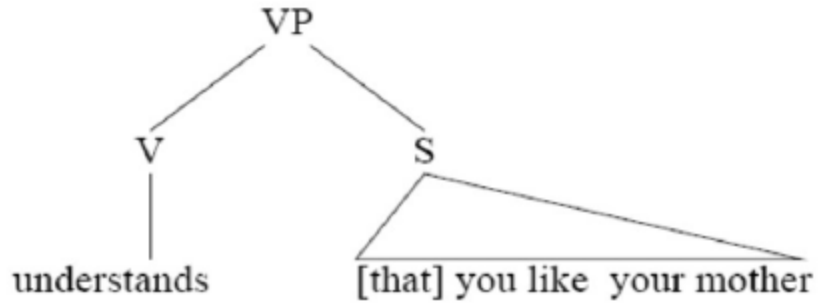
- Because it is ambiguous:

1. The computer understands you as well as your mother understands you.

2. The computer understands that you like (love) your mother.

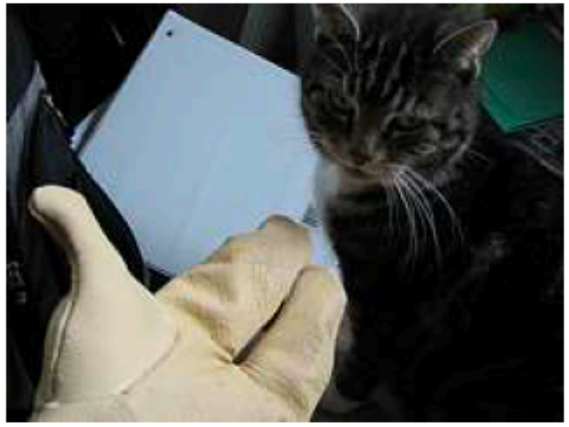
3. The computer understands you as well as it understands your mother.





Why NLP is hard

- Even simple sentences are highly ambiguous
- "Get the cat with the gloves"



More Examples of Ambiguity

- Iraqi Head Seeks Arms
- Ban on Nude Dancing on Governor's Desk
- Juvenile Court to Try Shooting Defendant
- Teacher Strikes Idle Kids
- Kids Make Nutritious Snacks
- British Left Waffles on Falkland Islands
- Red Tape Holds Up New Bridges
- Bush Wins on Budget, but More Lies Ahead
- Hospitals are Sued by 7 Foot Doctors
- Stolen Painting Found by Tree
- Local HS Dropouts Cut in Half

NLP vs Speech Processing

- Natural Language Processing

  = automatic processing of written texts

  1. Natural Language Understanding

  Input = text

  2. Natural Language Generation

  Output = text

- Speech Processing

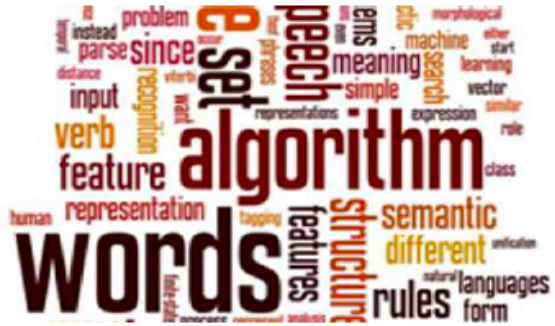
  = automatic processing of speech

  1. Speech Recognition

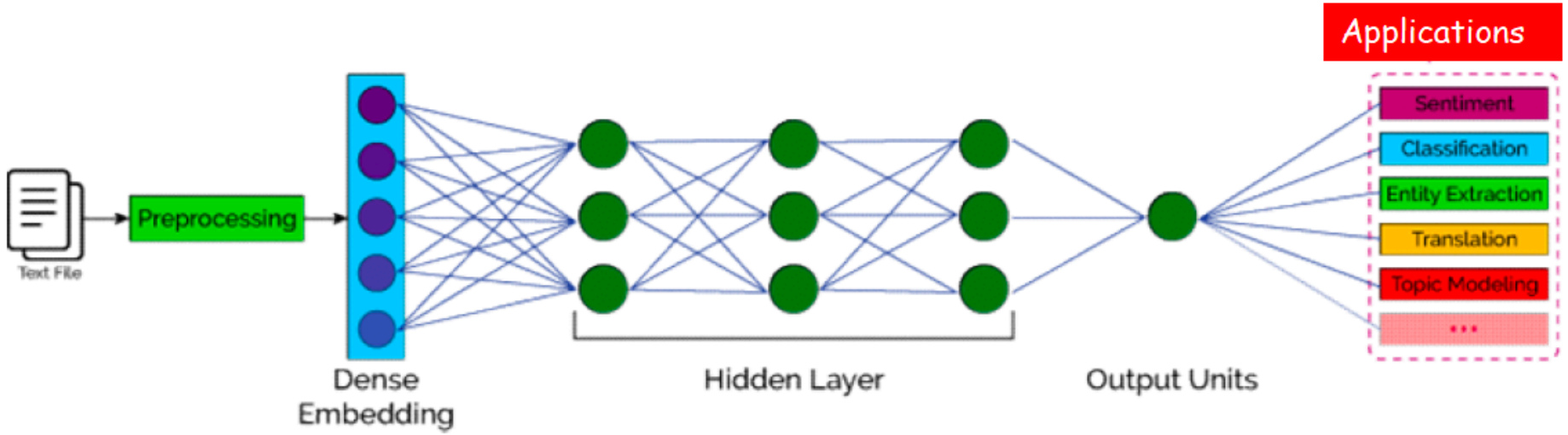
  Input = acoustic signal

  2. Speech Synthesis

  Output = acoustic signal



Natural Language Processing (circa 2010-today)



- Deep Neural Networks applied to NLP problems
  - Rules are developed automatically (using machine learning)
  - And the linguistic features are found automatically!

Topics

- Words and Semantics
- Bag of word model
- Classical Approaches

Words and Semantics

- In all NLP tasks, we need to "access" the word/sentence semantics in order to solve a given task.
- Finding a link between words and semantics is not always trivial.

Example

- Example: question answering.

How is the weather now?

It's raining cats and dogs.

https://commons.wikimedia.org/wiki/File:Weather_symbols_p.png
https://unsplash.com/photos/F-t5EpfQNpk

Example

- Example: question answering.
- Note how we are interested in the semantics…

How is the weather now?

It's raining cats and dogs.

https://commons.wikimedia.org/wiki/File:Weather_symbols_p.png
https://unsplash.com/photos/F-t5EpfQNpk

Example

- Example: question answering.
- Note how we are interested in the semantics…
- but we only have access to the words.

How is the weather now?

It's raining cats and dogs.

https://commons.wikimedia.org/wiki/File:Weather_symbols_p.png
https://unsplash.com/photos/F-t5EpfQNpk

Linking Words and Semantics

- We need some way to link words and semantics:
  - Distributional Hypothesis
  - Principle of Compositionality

Distributional Hypothesis

- "Words that occur in the same contexts tend to have similar meanings (Harris, 1954)."
- "You shall know a word by the company it keeps" Firth, J. R. 1957:11
- The distributional hypothesis suggests that the more semantically similar two words are, the more distributionally similar they will in turn be, and thus the more that they will tend to occur in similar linguistic contexts.

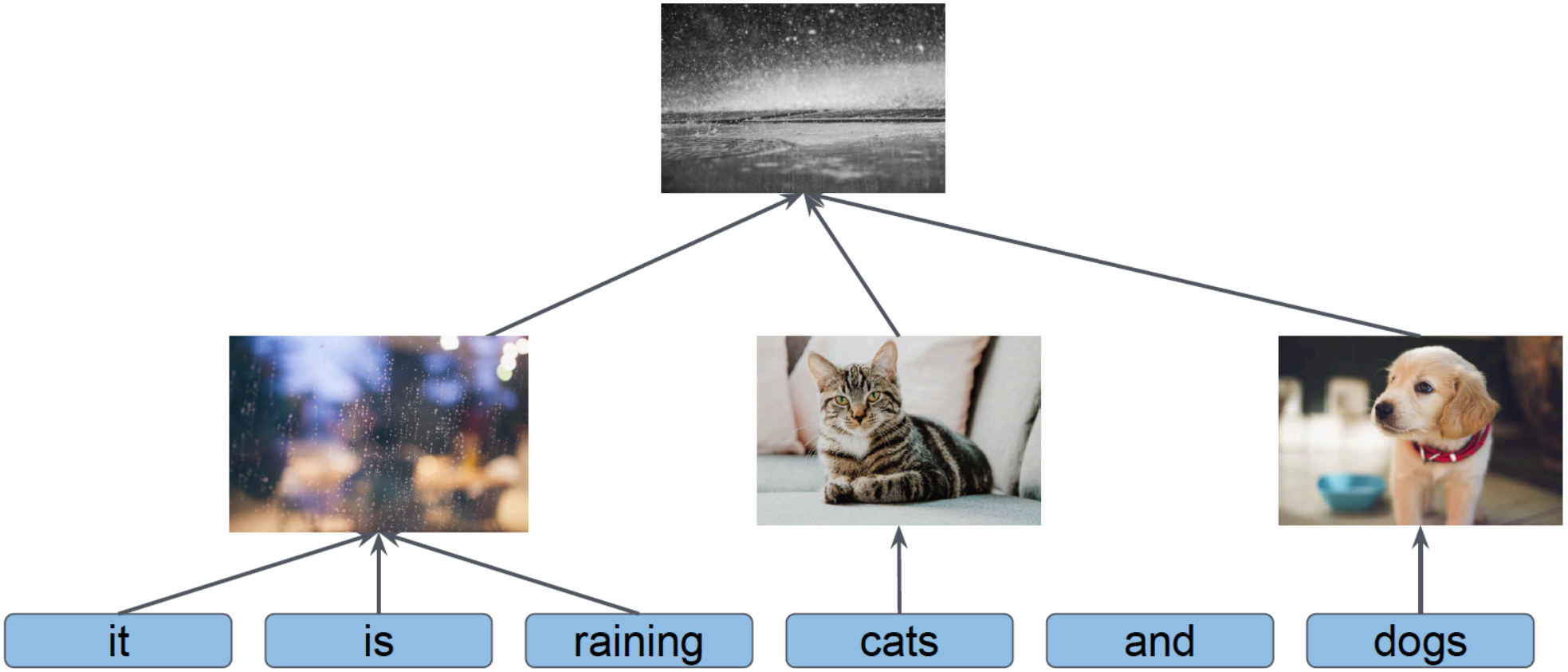
Distributional Hypothesis

- "Through their **intelligence**, humans possess the cognitive abilities to learn, form concepts [...]"
- "**Intelligence** is what makes humans the most successful [...]"
- "Human **intelligence** is essential to better understand [...]"
- Note how the word "human" is often in the context of the word "intelligence".

Principle of Compositionality

- "In mathematics, semantics, and philosophy of language, the principle of compositionality is the principle that the meaning of a complex expression is determined by the meanings of its constituent expressions and the rules used to combine them."

Principle of Compositionality



https://unsplash.com/photos/ngqyo2AYYnE, https://unsplash.com/photos/IbPxGLgJiMI,
https://unsplash.com/photos/VR0s3Yqm2RA, https://unsplash.com/photos/F-t5EpfQNpk

Linking Words and Semantics

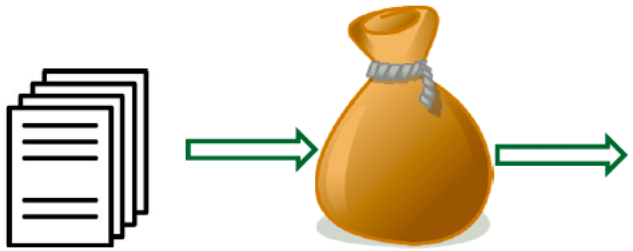
- The distributional hypothesis is a promising way to generate semantics for a word by looking at the context where the word appears.
- The principle of compositionality allows us to tackle the NLP tasks in a hierarchical way.
- Both can help us in creating algorithms to solve NLP tasks.
- Before doing so though, we can start by looking at older/classical approaches to better understand the evolution of some NLP algorithms.

Topics

- ***Words and Semantics***
- Bag of word model
- Classical Approaches

Bag-of-word Model (BOW)

- A simple model where word order is ignored

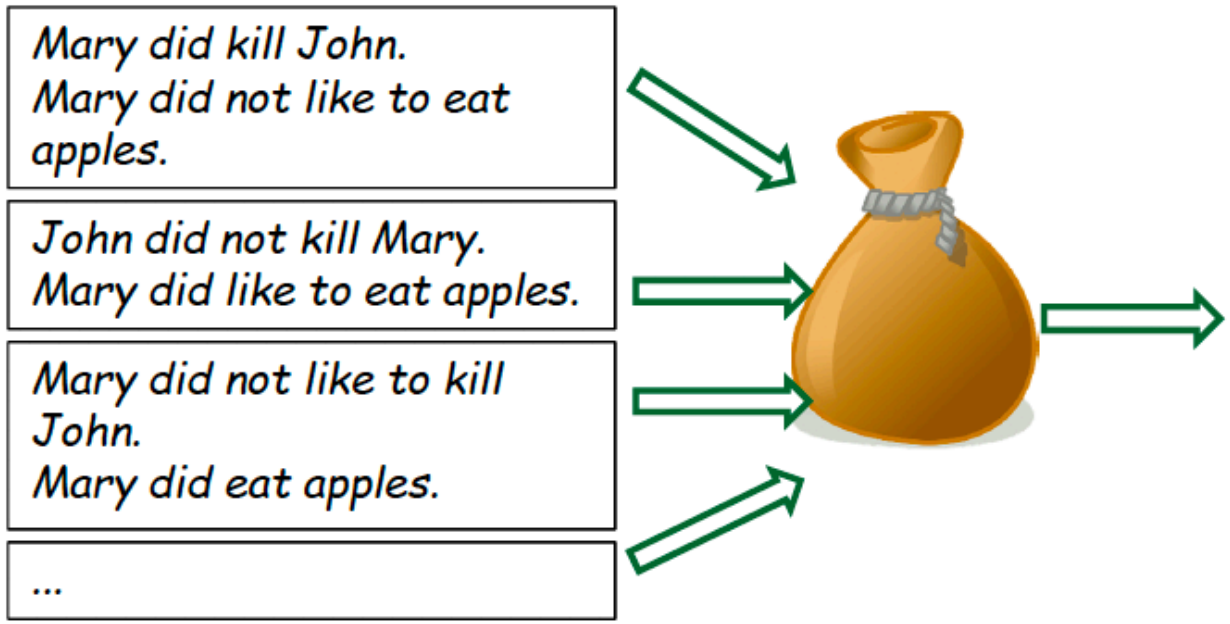


Word	Freq.
Mary	2
apples	1
did	2
eat	1
John	1
kill	1
like	1
not	1
to	1

- used in many applications:
  - NB spam filter seen in class a few weeks ago
  - Information Retrieval (e.g. google search)
  - ...
- But has severe limits to understand meaning of text...
- Maybe we should take word order into account...

Limits of BOW Model

- word order is ignored ==> meaning of text is lost.



Word	Freq.
Mary	2
apples	1
did	2
eat	1
John	1
kill	1
like	1
not	1
to	1

- n-grams take [a bit of] word order into account

Topics

- ***Words and Semantics***
- ***Bag of word model***
- Classical Approaches

Tokens

- A token is a unit in a text sequence.
- It can be composed of:
  - characters, e.g., “a”, “b”, ...
  - subwords, e.g., “er”, “est”, ...
  - words, e.g., “cat”, “house”, ...

Tokens – Vocabulary Size

- The longer the token string, the bigger the vocabulary.
- words > subwords > characters
- E.g.,

  words: ~80k

  subwords: ~20k

  characters: < 100

- A small vocabulary is usually better (both computationally and result wise).

Tokens – Sentence Length

- The smaller the token unit, the longer the representation of a sentence.
- characters > subwords > words
- E.g.,

  “hi there” (2 words)

  “hi _ the re” (4 subwords)

  “h i _ t h e r e” (8 characters)
- A short representation is usually better (both computationally and result wise).

Tokens – Out Of Vocabulary

- An out-of-vocabulary (OOV) token is a token that does not appear in training.
- In general, there is a big chance that the training data will not contain all the possible words for a given task.
  - The amount of OOV tokens can be quite large when using word-based
- vocabularies.
- On the other hand, the chance that a subword or a character does not appear in training is very small.
  - OOV tokens are very few (or absent) when using subwords or characters.

Tokens – Summary

- Every choice (word / subword / character) has its own advantages and disadvantages.
- Subwords are usually a good compromise that avoids the two extreme cases of having a too big vocabulary and having too long sentences.
- They are also less affected by the OOV problem.
- In the rest of this slide, for the sake of simplicity, we will assume that a token is a word.

Tokens Representation

- In a **one hot** representation, each token is associated with a vector whose elements are all equal to 0, except one, which has a value equal to 1.
  - The vector size is equal to the vocabulary size.
  - The vector is formed of bits (which can take values 0 / 1), with each bit corresponding to a specific word.
- Example #1 - with a vocabulary of 3 words ('cat', 'dog', 'house'):

  'cat' = [0, 0, 1]

  'dog' = [0, 1, 0]

  'house' = [1, 0, 0]

Tokens Representation

- Example #2 - with a vocabulary of 4 words ('cat', 'dog', 'house', 'pc'):

  'cat' = [0, 0, 0, 1]

  'dog' = [0, 0, 1, 0]

  'house' = [0, 1, 0, 0]

  'pc' = [1, 0, 0, 0]

- Note how vector size = vocabulary size

(this can be a problem if the vocabulary size is big)

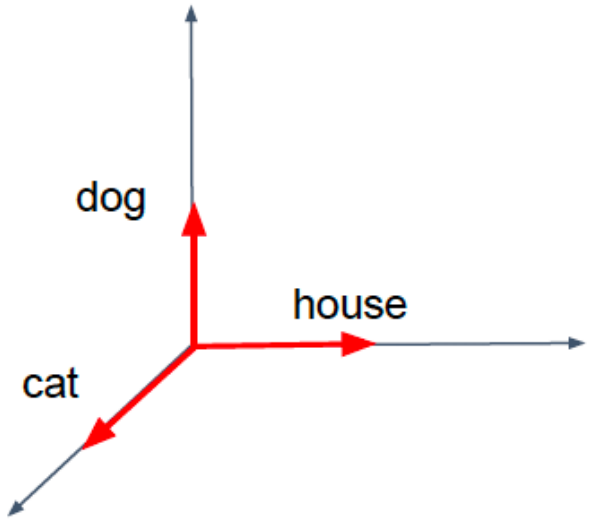
Tokens Representation

- Words represented as one hot vectors are all at the same distance of each other.
- This is not desirable.
- For example, we would prefer to have ‘dog’ and ‘cat’ closer to each other than they are to ‘house’ (which is not the case here)

‘cat’ = [0, 0, 1]

‘dog’ = [0, 1, 0]

‘house’ = [1, 0, 0]



Tokens Representation

- A **bag of word** representation corresponds to the sum of one hot vectors.
- Example: given the following one hot vectors

  'cat' = [0, 0, 1]

  'dog' = [0, 1, 0]

  'house' = [1, 0, 0]

the bag of word representation of the sequence 'cat dog house' is: 'cat dog house' = [1, 1, 1]

- The order of the words is lost!
- E.g., "drink water not poison" = "drink poison not water"

Topics

- ***Words and Semantics***
- ***Bag of word model***
- **Classical Approaches**

The End