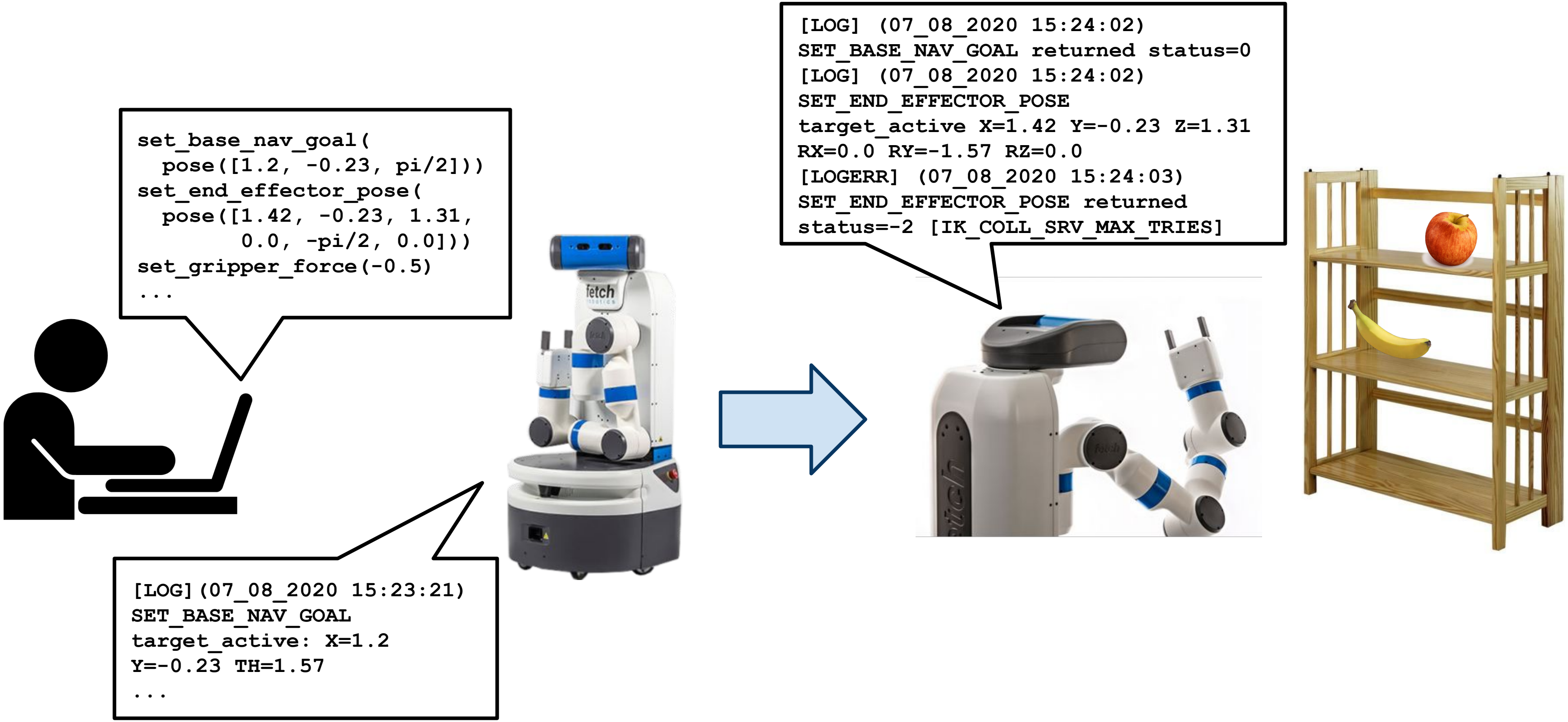
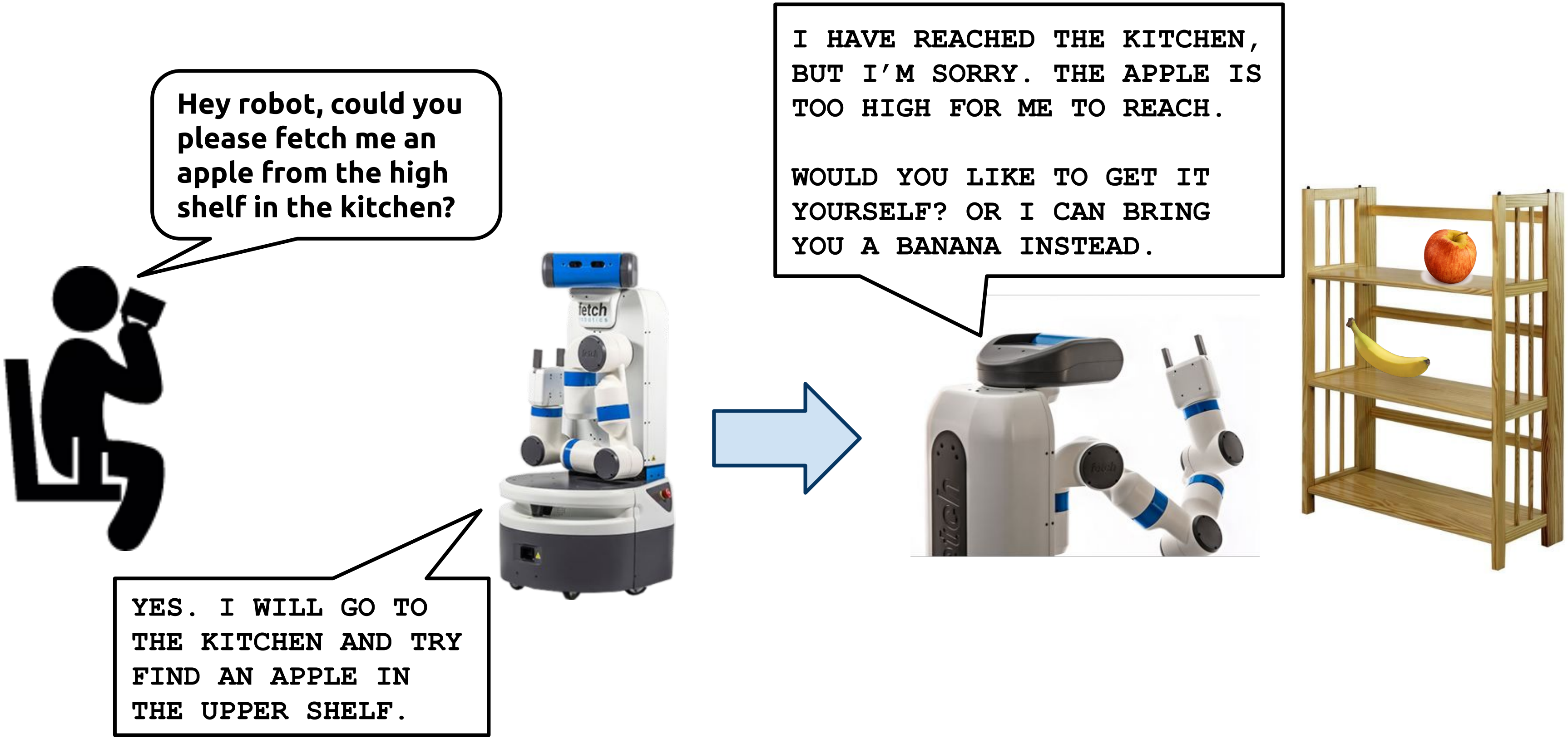
Introduction to Natural Language Processing (NLP)

Sebastian Castro

Robotics Software Engineer
MIT Computer Science & Artificial Intelligence Laboratory



set_base_nav_goal(
pose([1.2, -0.23, pi/2]))
set_end_effector_pose(
pose([1.42, -0.23, 1.31,
0.0, -pi/2, 0.0]))
set_gripper_force(-0.5)
...
[LOG](07_08_2020 15:23:21)
SET_BASE_NAV_GOAL
target_active: X=1.2
Y=-0.23 TH=1.57
...
[LOG] (07_08_2020 15:24:02)
SET_BASE_NAV_GOAL returned status=0
[LOG] (07_08_2020 15:24:02)
SET_END_EFFECTOR_POSE
target_active X=1.42 Y=-0.23 Z=1.31
RX=0.0 RY=-1.57 RZ=0.0
[LOGERR] (07_08_2020 15:24:03)
SET_END_EFFECTOR_POSE returned
status=-2 [IK_COLL_SRV_MAX_TRIES]
fetch

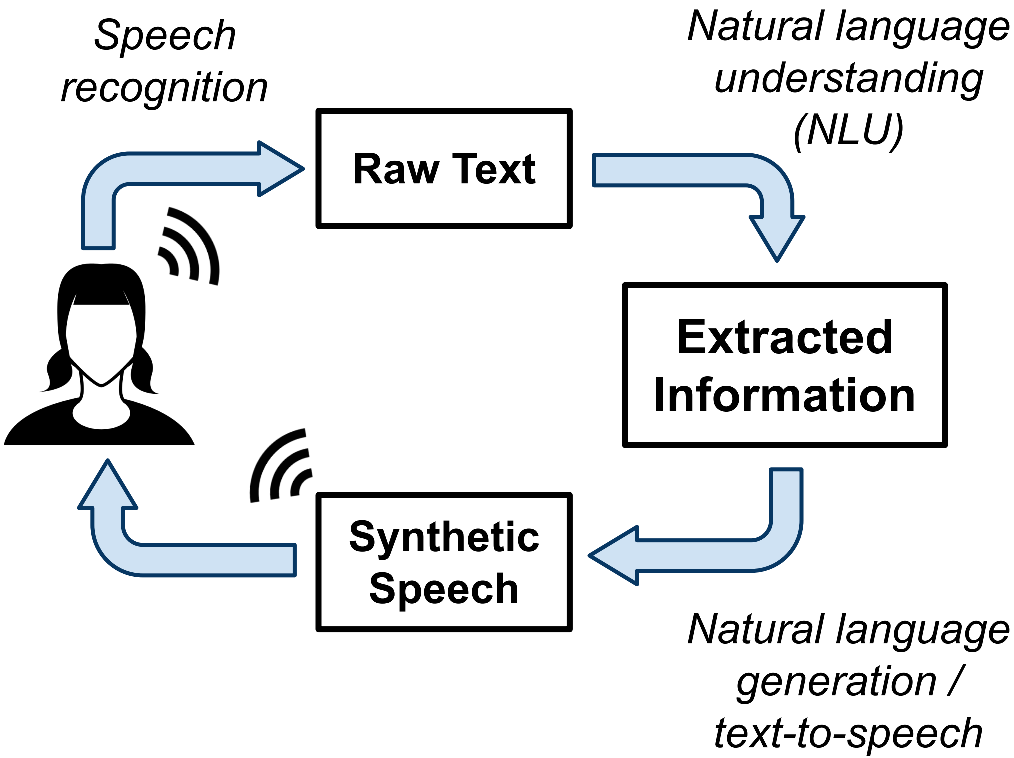

Hey robot, could you please fetch me an apple from the high shelf in the kitchen?
YES. I WILL GO TO THE KITCHEN AND TRY FIND AN APPLE IN THE UPPER SHELF.
I HAVE REACHED THE KITCHEN, BUT I'M SORRY. THE APPLE IS TOO HIGH FOR ME TO REACH.
WOULD YOU LIKE TO GET IT YOURSELF? OR I CAN BRING YOU A BANANA INSTEAD.
fetch

What is NLP?

A branch of **artificial intelligence**

dealing with **communication between humans and machines**

in the **natural language** of the human (text, speech, etc.)



Applications of NLP

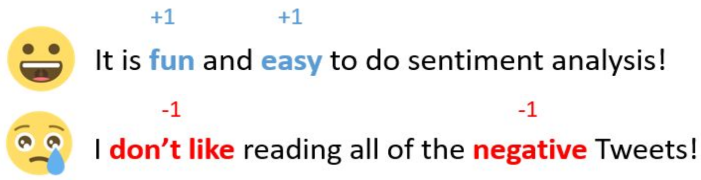
Classification: Sentiment analysis, topic/intent detection, part-of-speech tagging, etc.

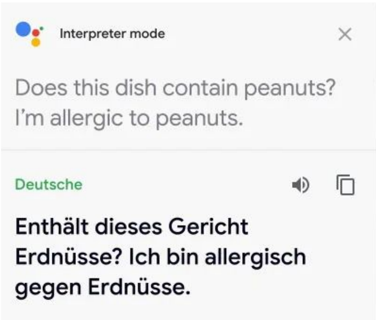
Generation: Translation, image captioning, text summarization, speech synthesis, etc.

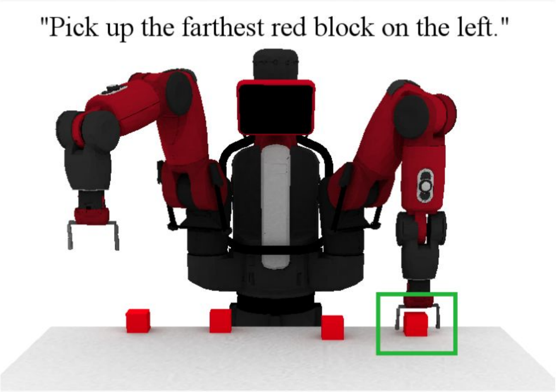
Discourse: Question answering, chatbots, etc.

Grounding: Associating language with entities in the real world (objects, actions, concepts, etc.)

… and more!



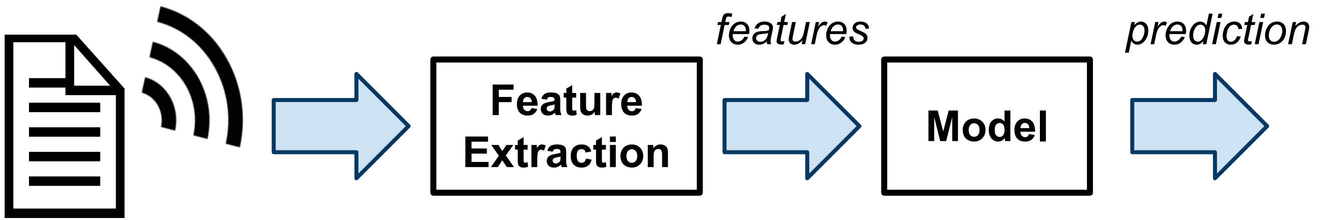




Rule-Based vs. Statistical NLP

Rule-Based Hard-coded rules (grammars, heuristics, patterns, etc.)

Statistical Automatically learning rules from large bodies* of data using statistical techniques such as *machine learning*.



** corpus (plural: corpora) = "body" in Latin*

Rule-Based NLP

Rule-Based NLP

Can perform well in simple, specific cases but does not generalize well.

Examples

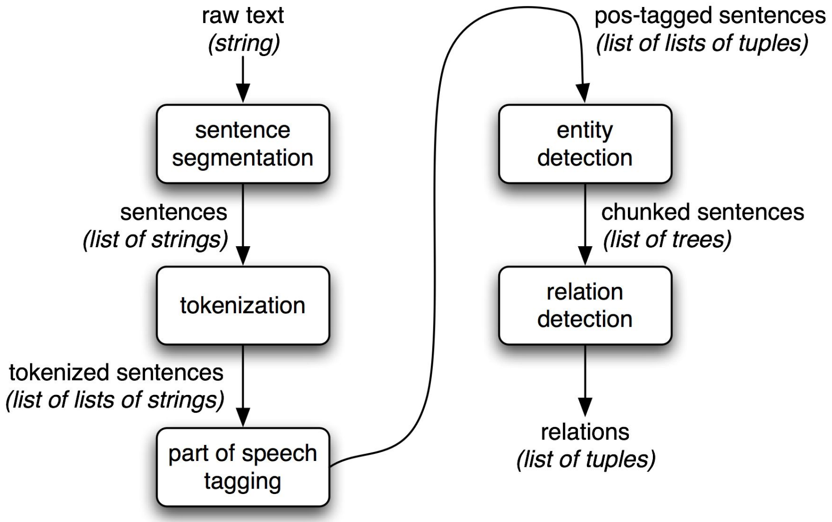
- Preprocessing text
- Searching for keywords, templates, patterns, etc. from a knowledge base
- Parsing and analysis using linguistic rules (*grammars*)

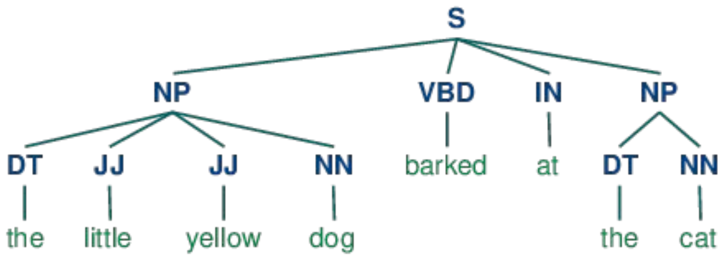
Typically, rule-based NLP supplements machine learning approaches.

Structure of Rule-Based NLP

A typical rule-based pipeline consists of:

1. Preprocessing text
   (e.g. sentence segmentation, tokenization)
2. Tagging parts of speech
   (usually involves a learned model)
3. Parsing the speech using a predefined grammar
4. Extracting key information
   (e.g. named entities, relations, coreferences)





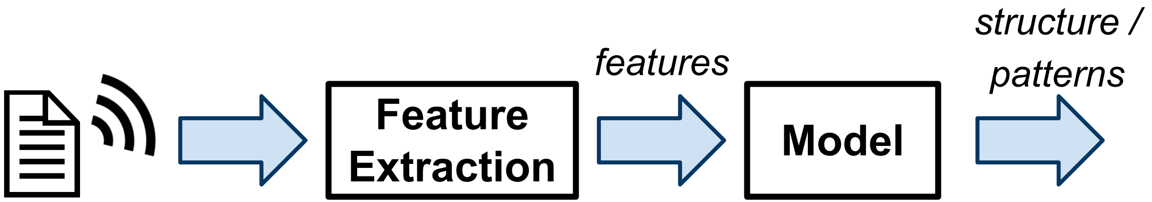
Source: https://www.nltk.org/book/ch07.html

Statistical NLP

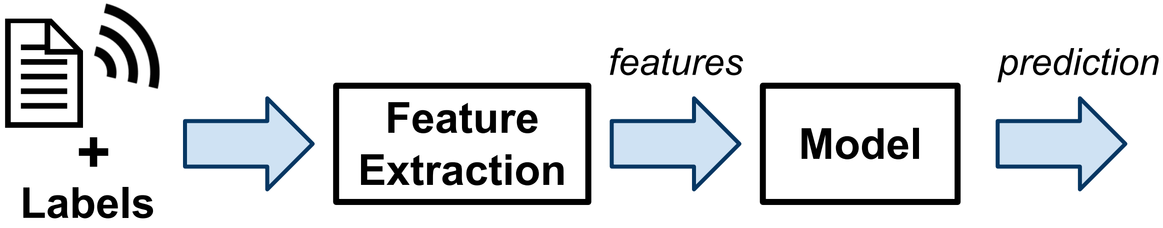
(B.D.L.: before deep learning)

Statistical NLP

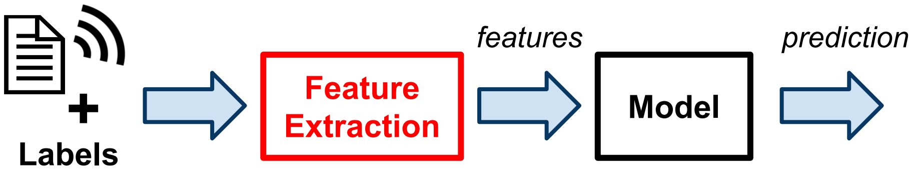
Unsupervised learning:
Learning structure from unlabeled data
(e.g. clustering, topic modeling)



Supervised learning:
Learning to predict from labeled data (e.g. regression, classification, generation, etc.)



Extracting Features from Text



Manual features:
Hand-coded information that may be related to the modeling goals

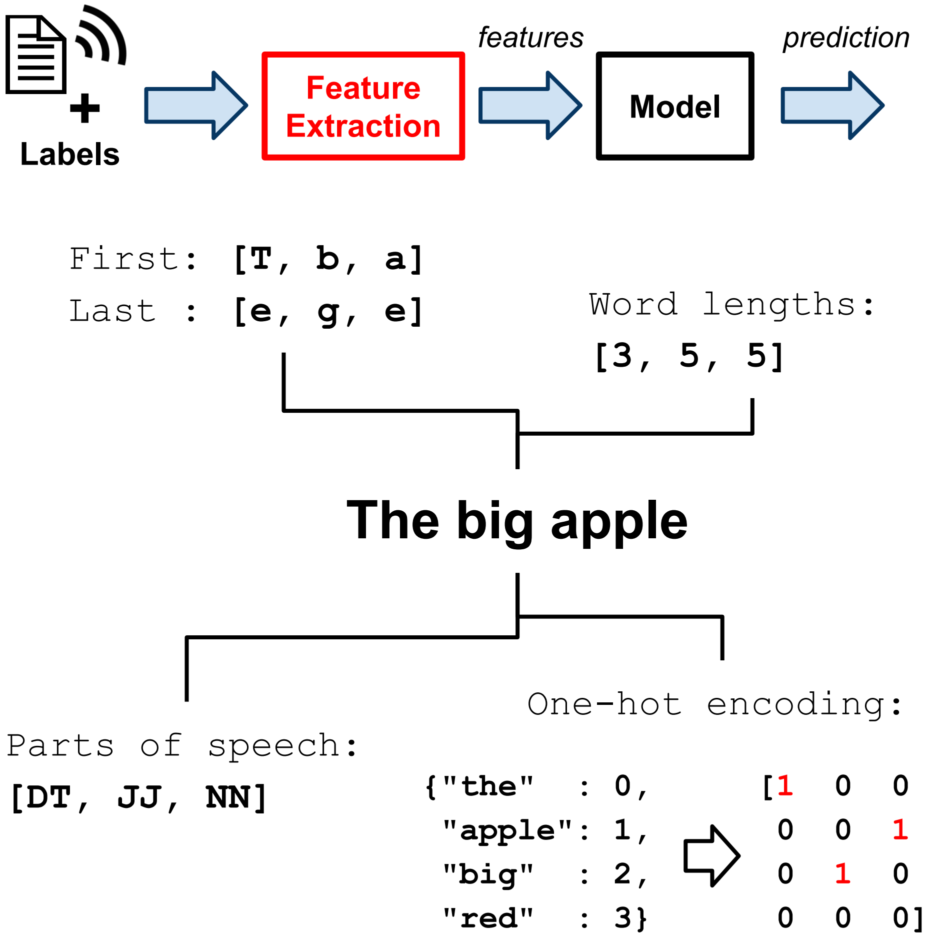
Bag of Words features:
Counting occurrences of text in documents
tf-idf for frequency weighting

n-grams: dealing with ordered sequences

Unsupervised learning:
Learning features from data and using them for a supervised model (e.g. word embeddings)

Extracting Features from Text

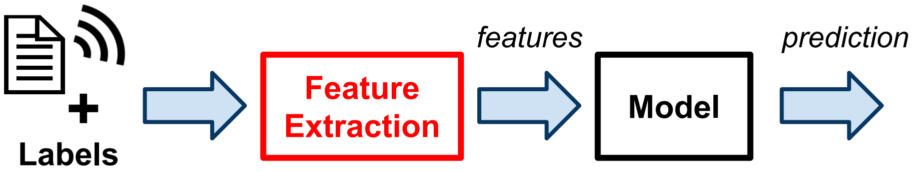
- **Manual features:**
  Hand-coded information that may be related to the modeling goals



Extracting Features from Text

Manual features:
Hand-coded information that may be related to the modeling goals

Bag of Words features:
Counting occurrences of text in documents



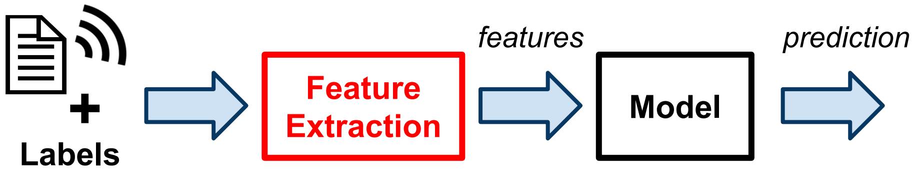

```
Doc 1: I love dogs.
Doc 2: I hate dogs and knitting.
Doc 3: Knitting is my hobby and my passion.
```

	I	love	dogs	hate	and	knitting	is	my	hobby	passion
Doc 1	1	1	1							
Doc 2	1		1	1	1	1				
Doc 3					1	1	1	2	1	1

Bag of Words
Document-Term Matrix (DTM)

Source: https://www.datacamp.com/community/tutorials/text-analytics-beginners-nltk

Extracting Features from Text



Manual features:

Hand-coded information that may be related to the modeling goals

Bag of Words features:

Counting occurrences of text in documents

tf-idf for frequency weighting

tf = Term Frequency

$$tf(t) = \frac{\text{\# t in document}}{\text{Total \# terms in document}}$$

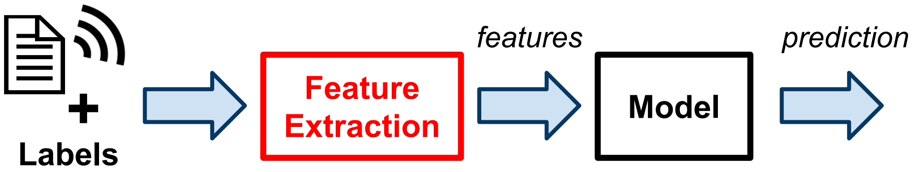
idf = Inverse Document Frequency

$$idf(t) = \frac{\text{Total \# documents}}{\text{\# documents with term t}}$$

	I	love	dogs	hate	and	knitting	is	my	hobby	passion
Doc 1	0.18	**0.48**	0.18							
Doc 2	0.18		0.18	**0.48**	0.18	0.18				
Doc 3					0.18	0.18	**0.48**	**0.95**	**0.48**	**0.48**

Source: https://www.datacamp.com/community/tutorials/text-analytics-beginners-nltk

Extracting Features from Text



Manual features:
Hand-coded information that may be related to the modeling goals

Bag of Words features:
Counting occurrences of text in documents
tf-idf for frequency weighting

n-grams: dealing with ordered sequences

Bag of words methods only count words, but not the *context* in which they occur, i.e., neighboring words. *n-grams* can help solve this.

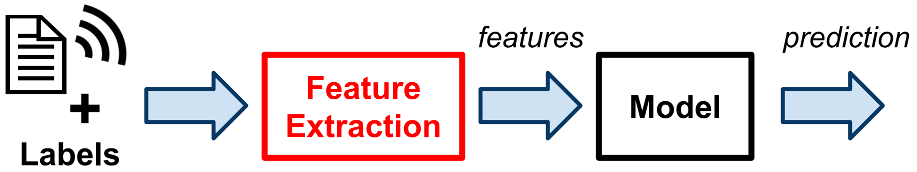
The quick brown fox jumps over the lazy dog

```
1-grams (words):     ["the", "quick", "brown", ...]

2-grams (bigrams):   ["the quick", "quick brown",
                      "brown fox", ...]

3-grams (trigrams):  ["the quick brown",
                      "quick brown fox",
                      "brown fox jumps", ...]
```

Extracting Features from Text

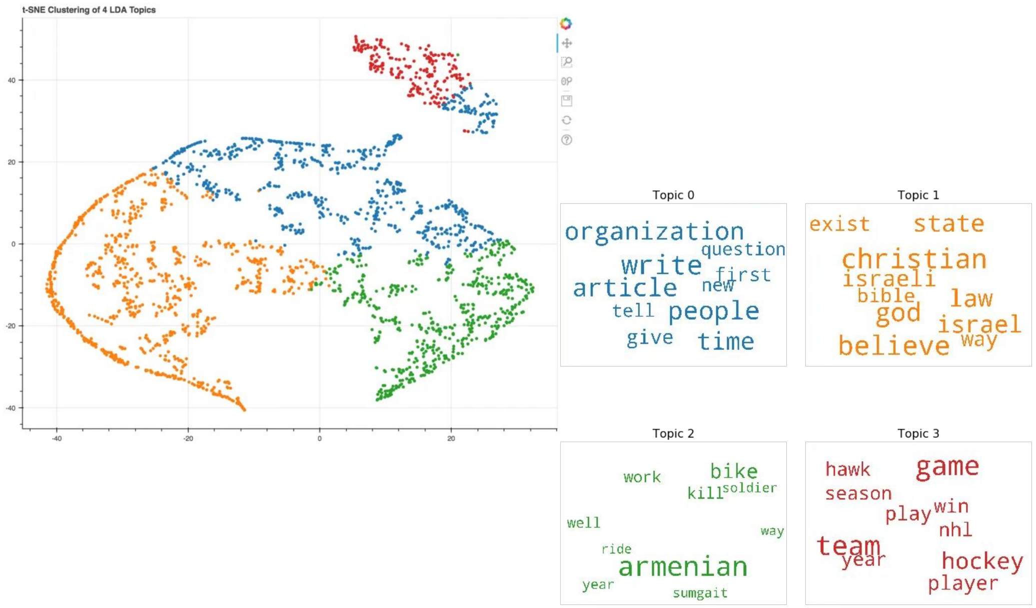


Manual features:
Hand-coded information that may be related to the modeling goals

Bag of Words features:
Counting occurrences of text in documents
tf-idf for frequency weighting

n-grams: dealing with ordered sequences

Unsupervised learning:
Learning features from data and using them for a supervised model (e.g. word embeddings)



Source:
https://www.machinelearningplus.com/nlp/topic-modeling-visualization-how-to-present-results-lda-models/

So Many Words, So Little Memory!
Keeping Feature Vector Sizes in Check

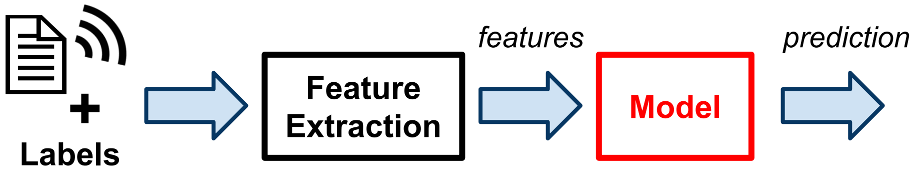
Removing stop words: e.g. `"the", "a", "that", "which"`

Limiting vocabulary size: Replace out-of-vocabulary words with `UNK` token

Using root terms of words: e.g. `“walking”, “walks”, “walked” → “walk”`
Techniques: *stemming*, *lemmatization*, *canonicalization*

Using sub-word features: English: 170000 words, 44 phonemes, 26 characters
Leads to much lower feature dimensions, but more difficult to keep the language “natural”

Types of Models



Once you have extracted features from text, various types of machine learning models can be used for classification, regression, text generation, etc.

Unsupervised

- **Clustering**
  e.g. k-means,
  Expectation Maximization (EM)
- Topic modeling
  e.g. Latent Dirichlet Allocation (LDA),
  Latent Semantic Analysis (LSA)
- **Word embeddings**
  (or other sub-word embeddings)

Supervised

- **Decision trees**
- **Bayesian algorithms** (e.g. Naive Bayes)
- **Regression algorithms** (e.g. linear / logistic)
- **Instance-based algorithms**
  e.g. k-Nearest Neighbors,
  Support Vector Machines (SVM)
- **Neural networks**

Resource: https://machinelearningmastery.com/a-tour-of-machine-learning-algorithms/

Statistical NLP

(A.D.L.: After deep learning)

Where Does Deep Learning Come in?

Issues with traditional statistical NLP approaches:

- **Hand-engineered features** are inefficient
  - Need one feature dimension for each word / n-gram in the vocabulary
  - Feature vectors are sparse -- a typical sentence will have few nonzero elements
  - Each word / n-gram is treated independently -- no good representation for similar words (e.g. “the big dog” vs. “the large hound”)
- **Representational capacity** of “shallow” models is limited
- **Long and/or variable-sized sequences** are challenging
  - n-grams are impractical beyond 4- or 5-grams
  - Many traditional models accept fixed-size data

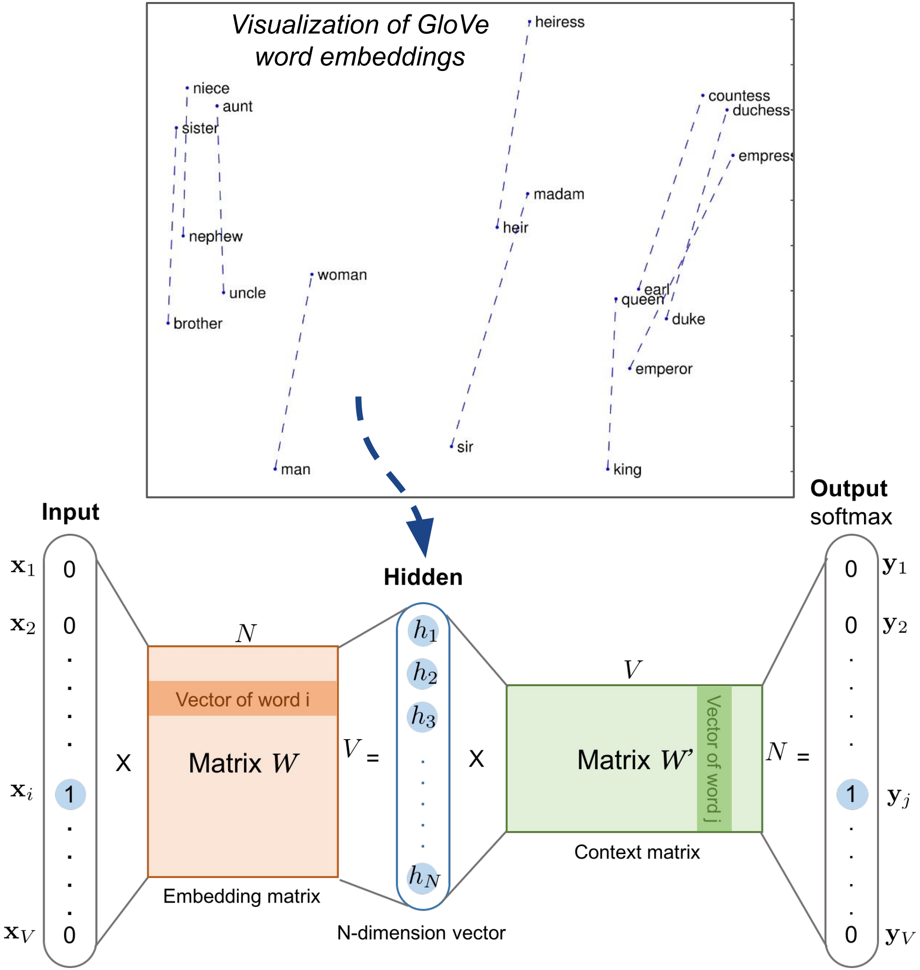
Learning Better Features

Can learn lower-dimensional **word embeddings** from large datasets.

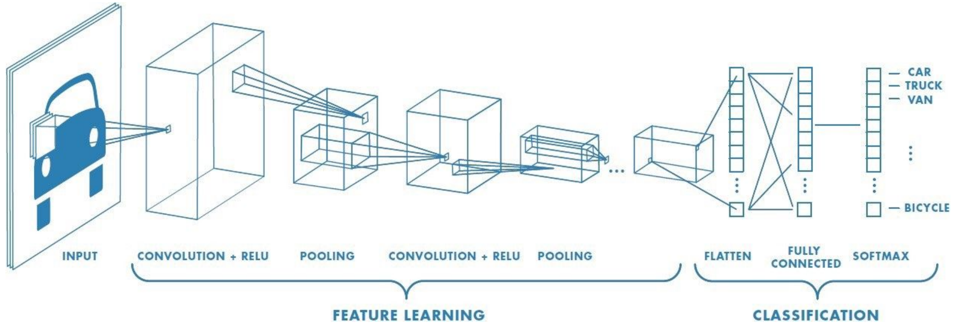
There are many common algorithms and pretrained embedding models.

- Context independent (words): Word2Vec, GloVe, FastText
- Context dependent (sentences): InferSent, ELMo

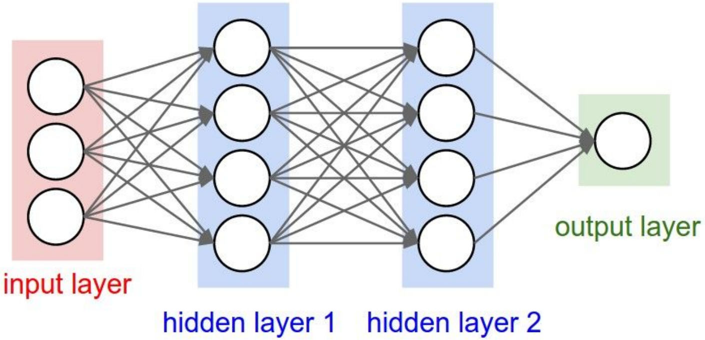
https://lilianweng.github.io/lil-log/2017/10/15/learning-word-embedding.html



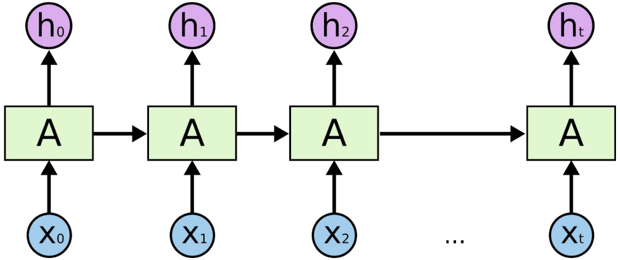
Types of Neural Networks



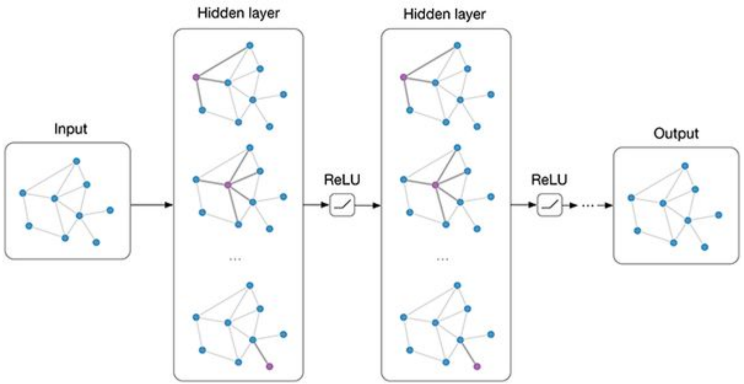
Convolutional Neural Network (CNN)



Fully Connected Network (FCN)
Multi-layer Perceptron (MLP)



Recurrent Neural Network (RNN)

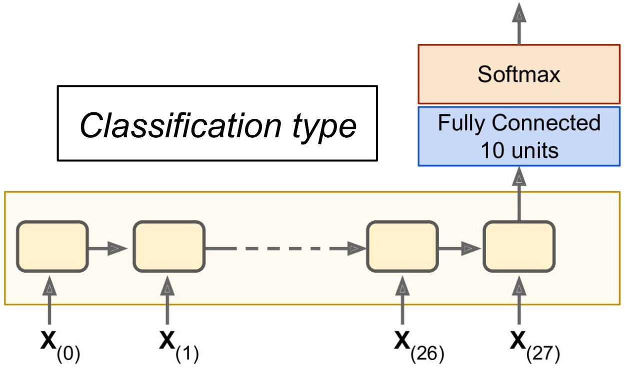


Graph Neural Network (GNN)

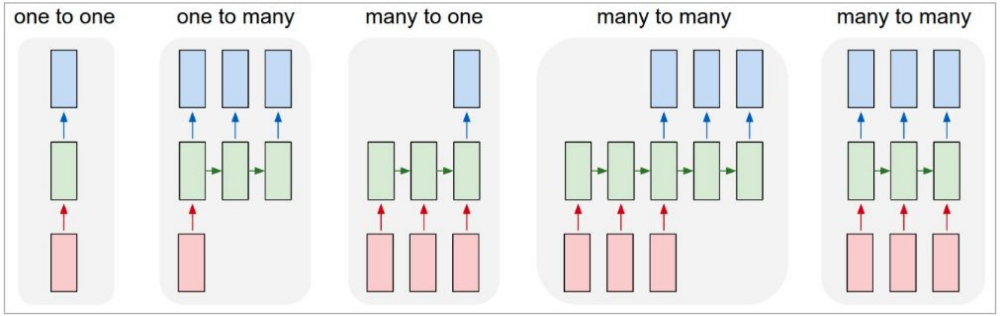
Recurrent Neural Networks (RNNs)

Handle variable-length sequences: recurrent units share the same weights so they can be chained to any length.

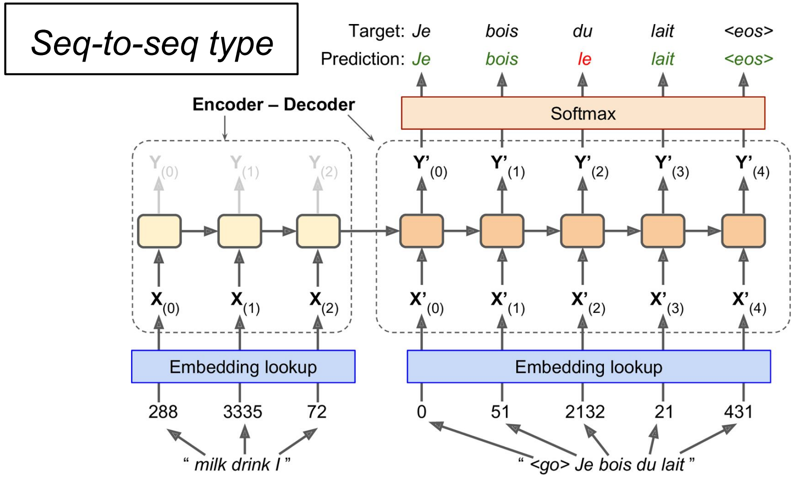
Encoder / decoder can use one-hot encoding, word embeddings, or sub-word embeddings.



https://www.oreilly.com/library/view/neural-networks-and/9781492037354/ch04.html



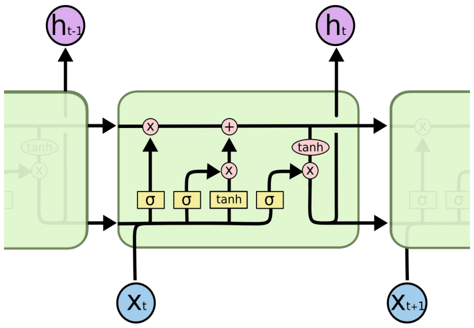
https://blog.floydhub.com/a-beginners-guide-on-recurrent-neural-networks-with-pytorch/



Improvements to RNNs

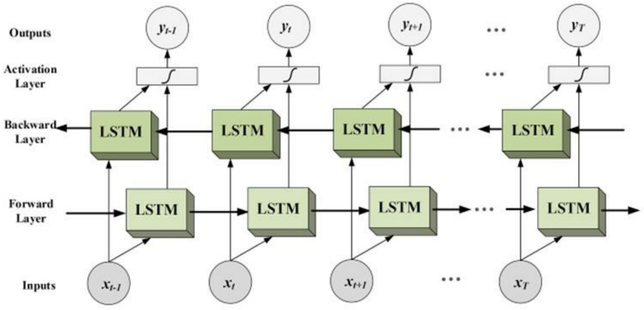
Long-Short Term Memory (LSTM) Units:
Handles issues with vanishing and exploding gradients, especially in longer sequences.

Other variations such as Gated Recurrent Unit (GRU)



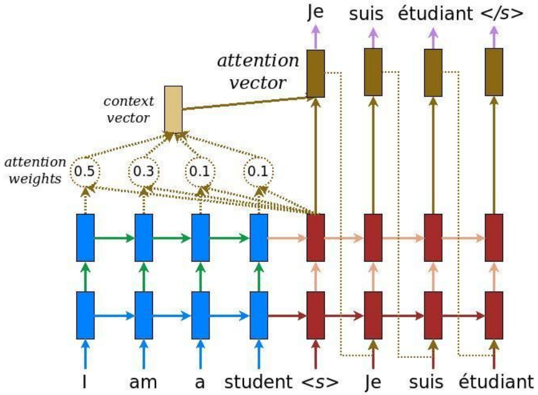
Bidirectional RNNs:

Can help to learn context in both forward and backward directions, if available.



Attention Mechanism:
Learn additional weights that operate on the entire sequence and "attend" to important parts.

Helps with longer sequences and input-output sequence reordering.



Modern NLP Models:
“Attention is All You Need”

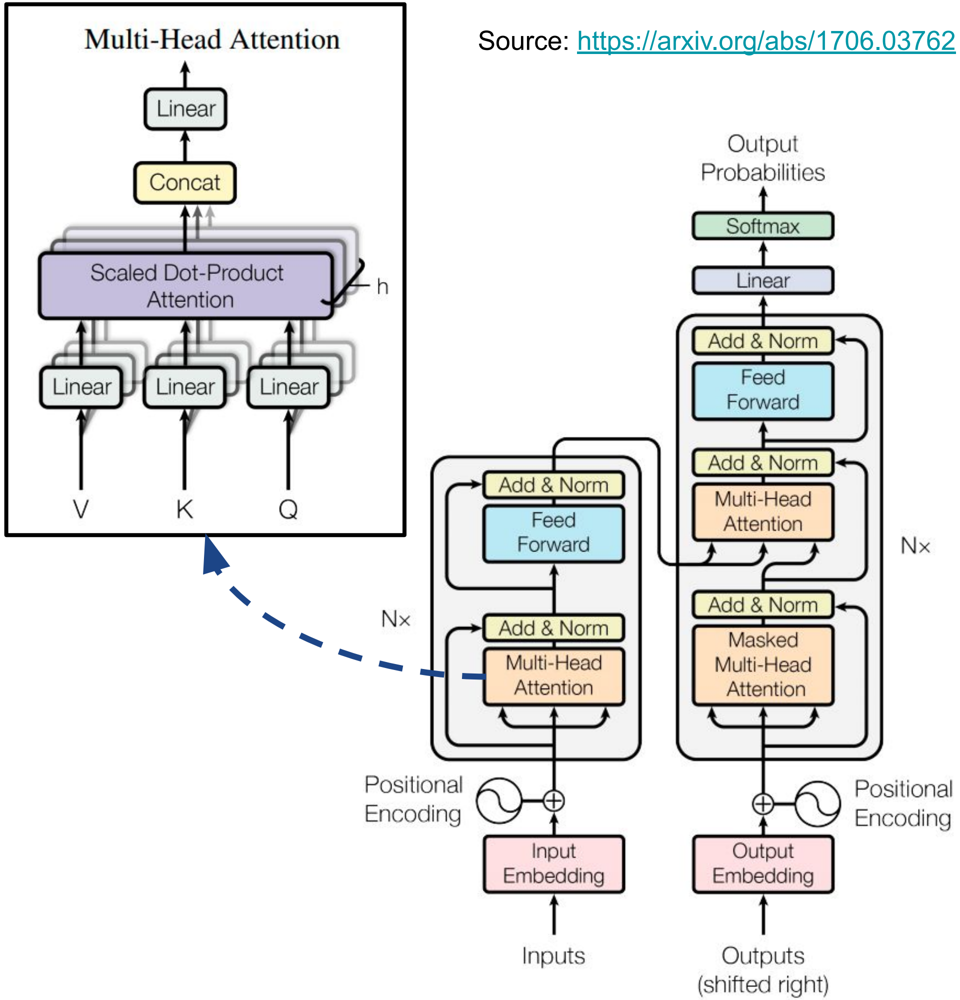
LSTMs are difficult to parallelize and have challenges for longer sequences.

Transformer networks use only attention mechanisms, with some positional encoding information.

… however, they are often huge models with lots of weights.

http://jalammar.github.io/illustrated-transformer/

Source: https://arxiv.org/abs/1706.03762



NLP Is More Than Text

... especially for robotics

Multimodal NLP:
“Humans have many senses, so why not robots?”

Example: Audio + Text

https://github.com/david-yoon/multimodal-speech-emotion
https://arxiv.org/abs/1810.04635

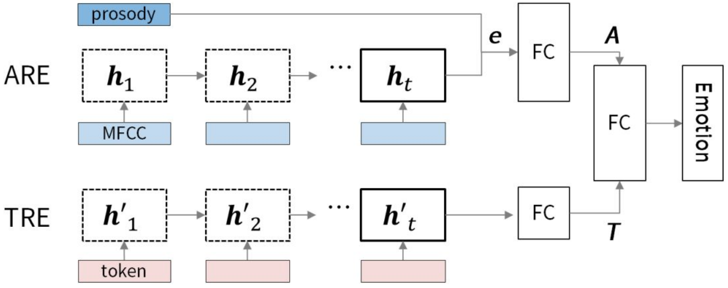
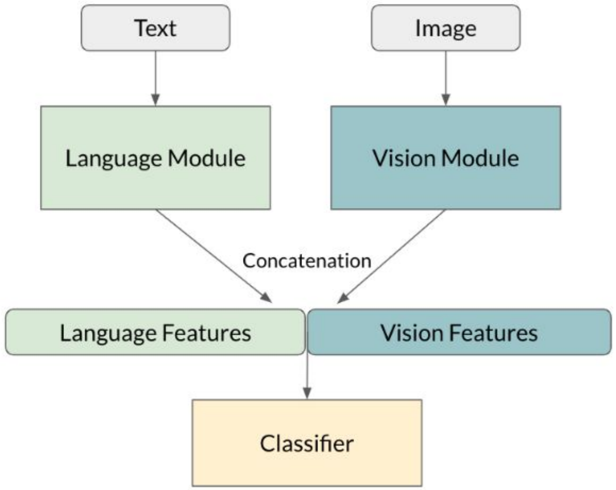


Fig. 1. Multimodal dual recurrent encoder. The upper part shows the ARE, which encodes audio signals, and the lower part shows the TRE, which encodes textual information.

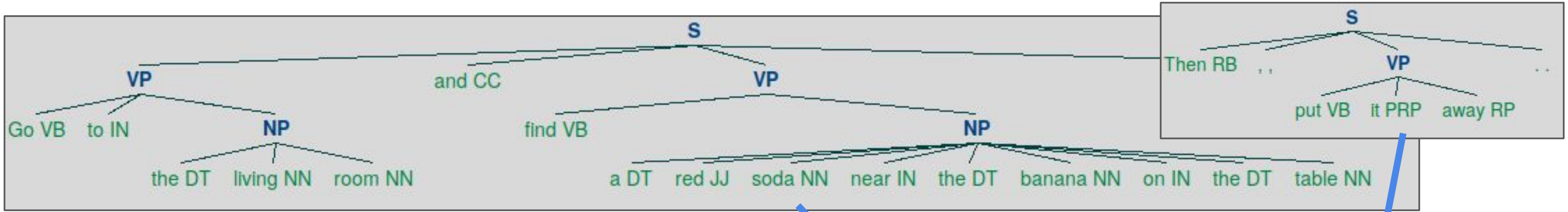
Example: Vision + Text

https://www.drivendata.co/blog/hateful-memes-benchmark/



Robotics Case Study: MIT CSAIL, 2020

“Go to the living room and find a red soda near the banana on the table. Then, put it away.”



Resolves to simple steps:

1. Go to the living room
2. Find a soda (using visual grounding ⟶)
3. Put a soda away
   (should be the one you just found… right?)

What if there is no red soda?
Want our visual grounding to detect absence of objects -- not just the most likely in the scene.

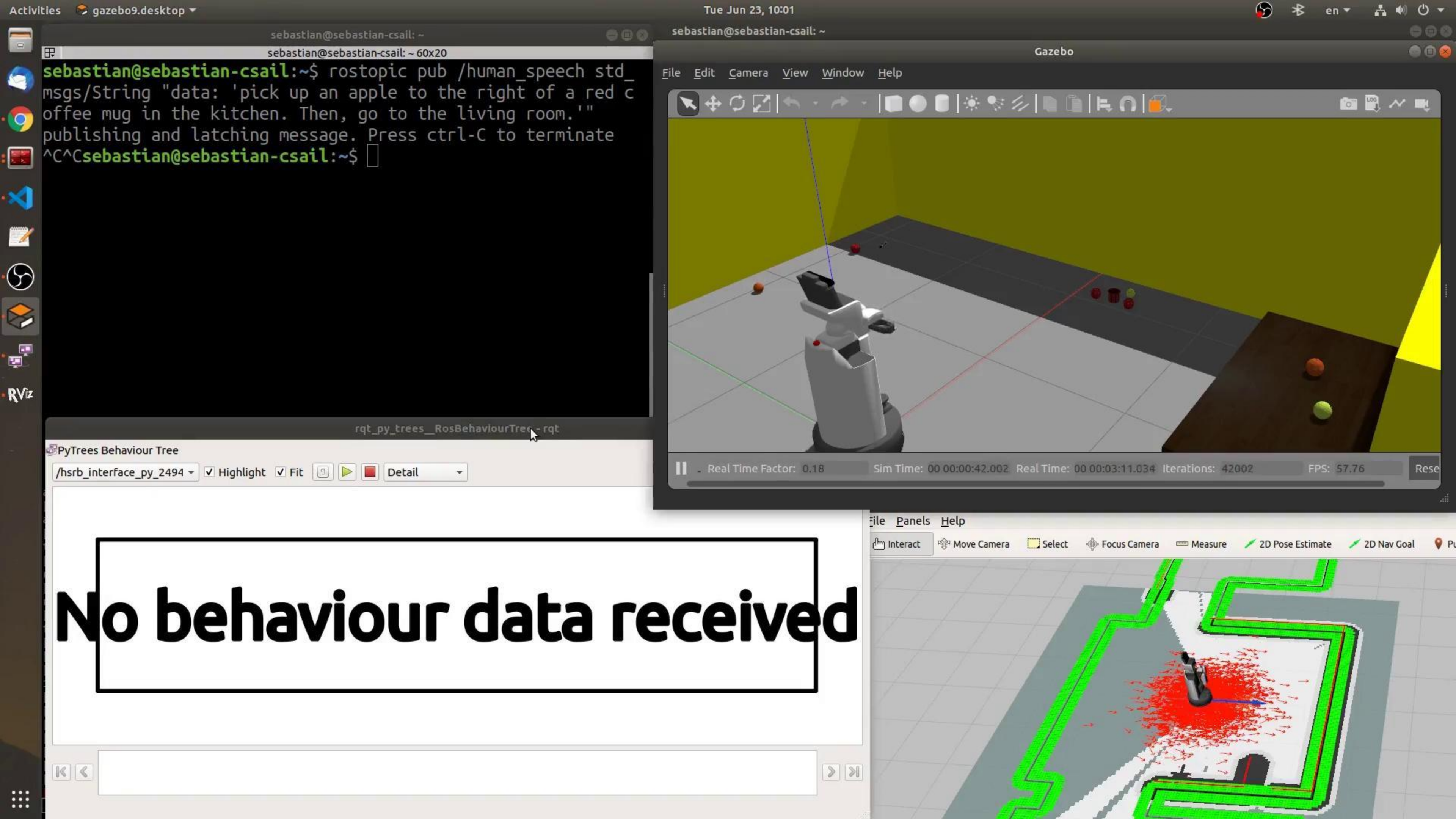
“it”

…

*“a red **soda** near the banana on the table”*

…

*“a **soda**”*

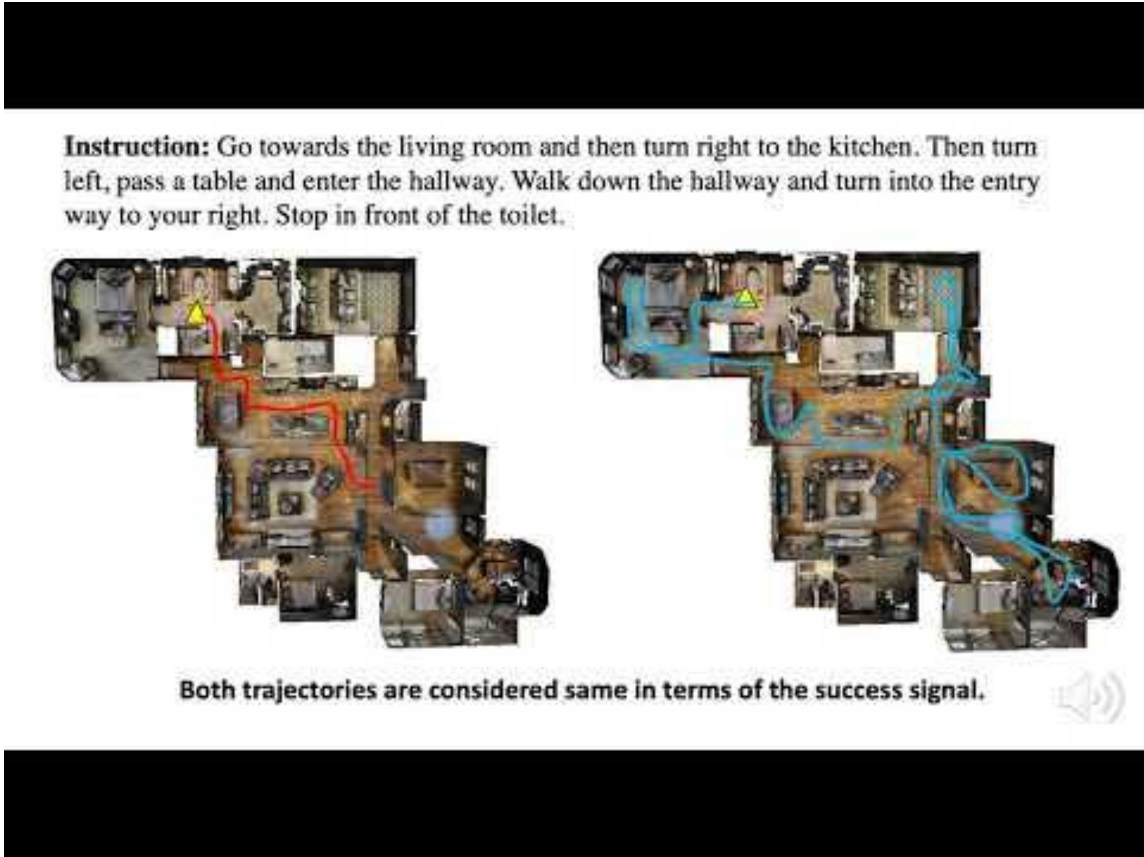

Activities
gazebo9.desktop
Tue Jun 23, 10:01
en
sebastian@sebastian-csail: ~
sebastian@sebastian-csail: ~ 60x20
sebastian@sebastian-csail:~$ rostopic pub /human_speech std_msgs/String "data: 'pick up an apple to the right of a red coffee mug in the kitchen. Then, go to the living room.'"
publishing and latching message. Press ctrl-C to terminate
^C^Csebastian@sebastian-csail:~$
sebastian@sebastian-csail: ~
Gazebo
File Edit Camera View Window Help
Real Time Factor: 0.18
Sim Time: 00 00:00:42.002
Real Time: 00 00:03:11.034
Iterations: 42002
FPS: 57.76
Rese
rqt_py_trees__RosBehaviourTree - rqt
PyTrees Behaviour Tree
/hsrb_interface_py_2494
Highlight
Fit
Detail
No behaviour data received
File Panels Help
Interact
Move Camera
Select
Focus Camera
Measure
2D Pose Estimate
2D Nav Goal
RViz

Robotics Case Study: Microsoft Research, 2019

Vision-Language Navigation (VLN)

Reinforcement Learning (RL) in photorealistic simulation environments

https://www.microsoft.com/en-us/research/blog/see-what-we-mean-visually-grounded-natural-language-navigation-is-going-places



Wrap-Up

Summary: Recap

- **NLP** = human-machine interaction in human language (speech, text, etc.)
- **Applications:**

Classification	Sentiment analysis, part-of-speech tagging, etc.
Generation	Translation, image captioning, text summarization, etc.
Discourse	Question answering, chatbots, etc.
Grounding	Associating language with entities in the real world
… and more!	

- **Rule-based vs. statistical methods:**
  Deep learning based methods have dominated NLP in the last few years
- NLP can be **multimodal**: active research area, especially in robotics

Summary: Popular NLP Tools and Resources

NLP tools:

- NLTK
- spaCy
- Stanza
- Gensim
- Pattern
- TextBlob

Machine Learning core:

- scikit-learn
- PyTorch
- TensorFlow

NLP specific ML libraries:

- 🤗 Transformers
- AllenNLP

NLP with audio:

- The Ultimate Guide to Speech Recognition with Python
- How to Convert Text to Speech in Python
- Audio Data Analysis Using Deep Learning with Python [Part 1] [Part 2]

Thank You!

roboticseabass.wordpress.com

github.com/sea-bass

Get the code and slides at github.com/sea-bass/**intro-nlp**