CS678 Advanced Natural Language Processing

Introduction and Class Outline

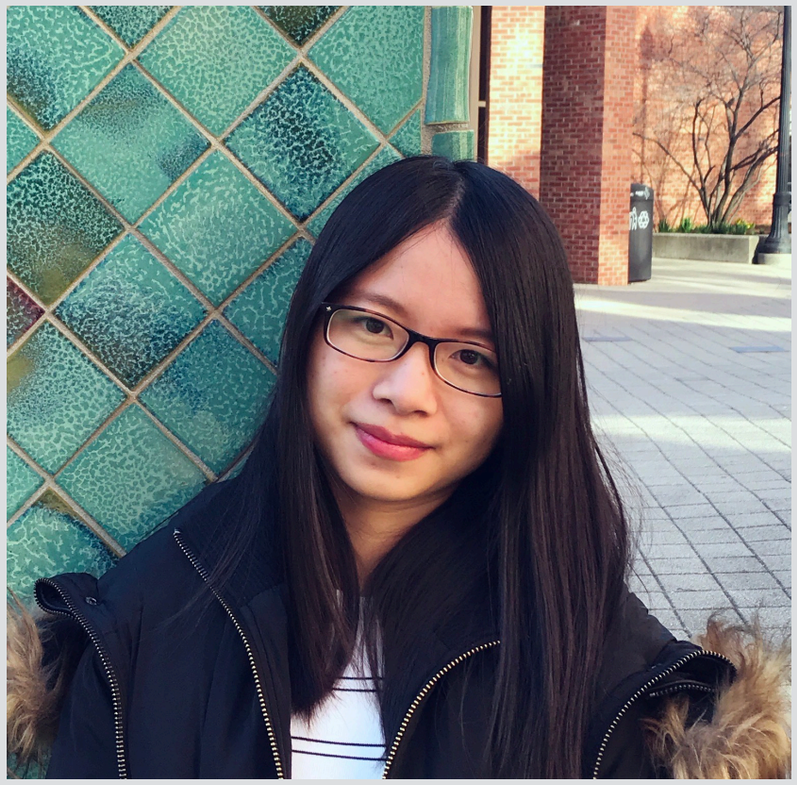
Ziyu Yao

https://nlp.cs.gmu.edu/course/cs678-fall22/

Acknowledgement: Some slides are adapted from Greg Durrett CS388@UT Austin and Antonios Anastasopoulos CS695@GMU

A bit about me & GMNLP

Hello, everyone!



Ziyu Yao, Asst. Prof.@CS

Ohio State University, CSE PhD

(ziyuyao@gmu.edu, ENGR4415)

- Research Interests: NLP and AI
  - Natural Language Interfaces (e.g., Question Answering, dialog systems)
  - Human-AI Interaction
  - Language and Code
  - Efficient Methods for NLP/AI

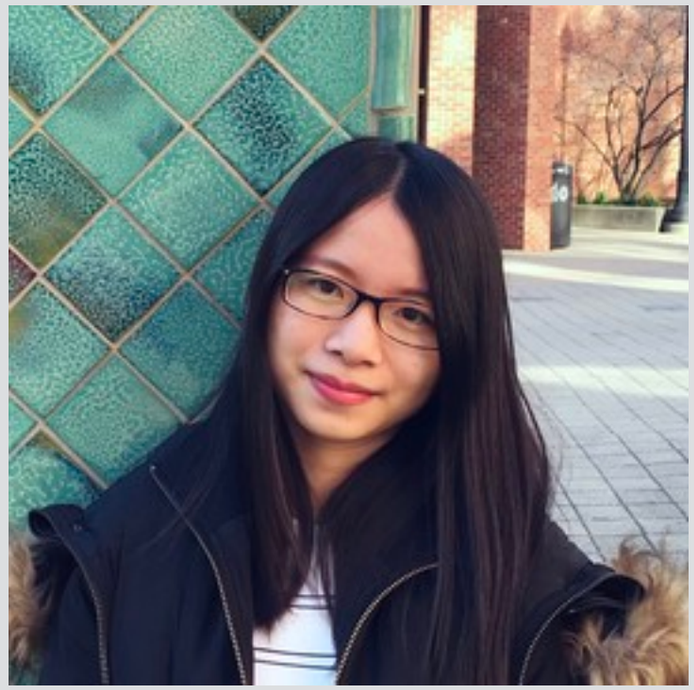
- Internship/Collaboration:

  Microsoft Research/Semantic Machines, Carnegie Mellon University, Facebook AI Research, Fujitsu Lab of America.

GMNLP

Antonios Anastasopoulos

Computational Linguistics, Machine Translation, Speech Recognition, NLP for Endangered Languages



Ziyu Yao

Question Answering/Generation, Human-AI Interaction, Language and Code, Efficient NLP/AI

https://nlp.cs.gmu.edu

We are growing!

Logistics

- Lectures: Thursday, 7:20pm - 10:00pm
- Location: Planetary Hall 206
- Course website: https://nlp.cs.gmu.edu/course/cs678-fall22 (syllabus, reading/assignment materials, links to Blackboard/Piazza)
- Teaching Assistant: Syeda Sabrina Akter (Sabrina)
- Instructor office hours: by appointment (ziyuyao@gmu.edu; virtual or in person at ENGR 4415)

Logistics

- We will use Blackboard for
  - Announcements
  - Distributing course materials before/after class
  - Distributing/submitting assignments
  - Sending grades
  - …
- We will also use Piazza for Q&A
  - Sign up immediately! (link on course website)

https://www.gmu.edu/safe-return-campus

Safe-Return-to-Campus Policy

Policy Status

Vaccinations and Boosters:	Strongly Encouraged; Available at Fenwick
Testing:	**Students:** Open to all; strongly encouraged for all weekly. All Residential Students: Pre-arrival via testing kit, week of Aug. 15, 2022. **Faculty and Staff:** Open to all; strongly encouraged for all weekly, especially the week of Aug. 15, 2022.
Reporting COVID Cases:	Students, faculty, and staff are required to report positive COVID cases via Mason COVID Health Check. However, we will no longer provide notification of potential positives to classes, events, etc.
Mason COVID Health Check:	Required when testing; available for use by event managers, instructors, etc.
Masks (Indoor):	**Mask Optional** **Exceptions:** Required when testing; Available for use by event managers, instructors, etc.

Note: Subject to change based on public health guidance

Course Requirements

- Prerequisites: CS 580 (Intro to AI) or CS 584 (Data Mining)

- Proficiency in (a) Algorithms and Data Structures and (b) Probability and Statistics (STAT 344) or equivalent.

- Proficiency in Python programming

  - All our materials will be based on Python

Textbooks

- Jurafsky and Martin, Speech and Language Processing, **3rd** edition (major)
- Jacob Eisenstein, Natural Language Processing (supplementary)
- Yoav Goldberg, Neural Network Methods in Natural Language Processing (supplementary)
- Free to download online (see course website)!

Class Format

- **Before class**: Read materials on the topic
  - Recommended but not required
- **During class:**
  - *Summary/Questions/Elaboration:* Instructor will summarize the materials, field questions, elaborate on details and talk about advanced topics
  - *Demo/Code Walk:* In some classes, we will walk through some demonstration code that implements a simple version of the main concepts presented in the reading materials
  - *Presentation*: (a) paper presentation; (b) project presentation
- **After class:**
  - *Assignments & Project:* We will talk about them in a minute
  - *Office hours:* with the instructor and the TA

Assignments

- There will be 2 assignments in total, each being an independent “small” coding project
  - HW1: Text Classification with Neural Nets
  - HW2: Implementing a small BERT model
- Students should work individually and independently

Project

- Team work of 2-4 students
- Highly recommend: pick from **suggested topics**
  - Could choose from both sections; priority given to the same section
- **Requirements & grading criteria: https://nlp.cs.gmu.edu/course/cs678-fall22/project/**
- Interest survey + 3 checkpoints: (1) proposal, (2) baseline implementation *from scratch*, (3) final project report and presentation
  - Note: all reports must follow the required format and include the required contents

Paper Presentation

- Second half of the semester (Oct 13 - Nov 17, totally 6 classes)
- Each class: 4 paper presentations, each in 15 minutes
  - Presenters: a group of 1-2 students, preparing slides
  - Other students: writing paper summaries as guided
- In total, 6 x 4 = 24 papers **(list will be shared)**
- Each student presents at least once in the semester

Grading Plan

- Two Assignments: 10% + 10%
- Paper Presentation and Summary: 15%
- Project: 65%
  - Proposal: 10%,
  - Baseline reproduction: 15%,
  - Final report and presentation: 40%

Note: Deadlines are all posted on course website!

Late Day Policy

- Late day policy: 3 late days over the two programming assignments without punishment.
  - Suggestion: save the late day tokens to the second assignment
  - Beyond: downgrade one half-grade per day late
  - Late submissions to project checkpoints are **not** allowed
- In case of serious illness or other excused absence (with necessary evidence, as defined by the university policies), students should inform and discuss with the instructor immediately

Outline for Today

- Class logistics

- Why NLP, and What is it?

- Scope of this course

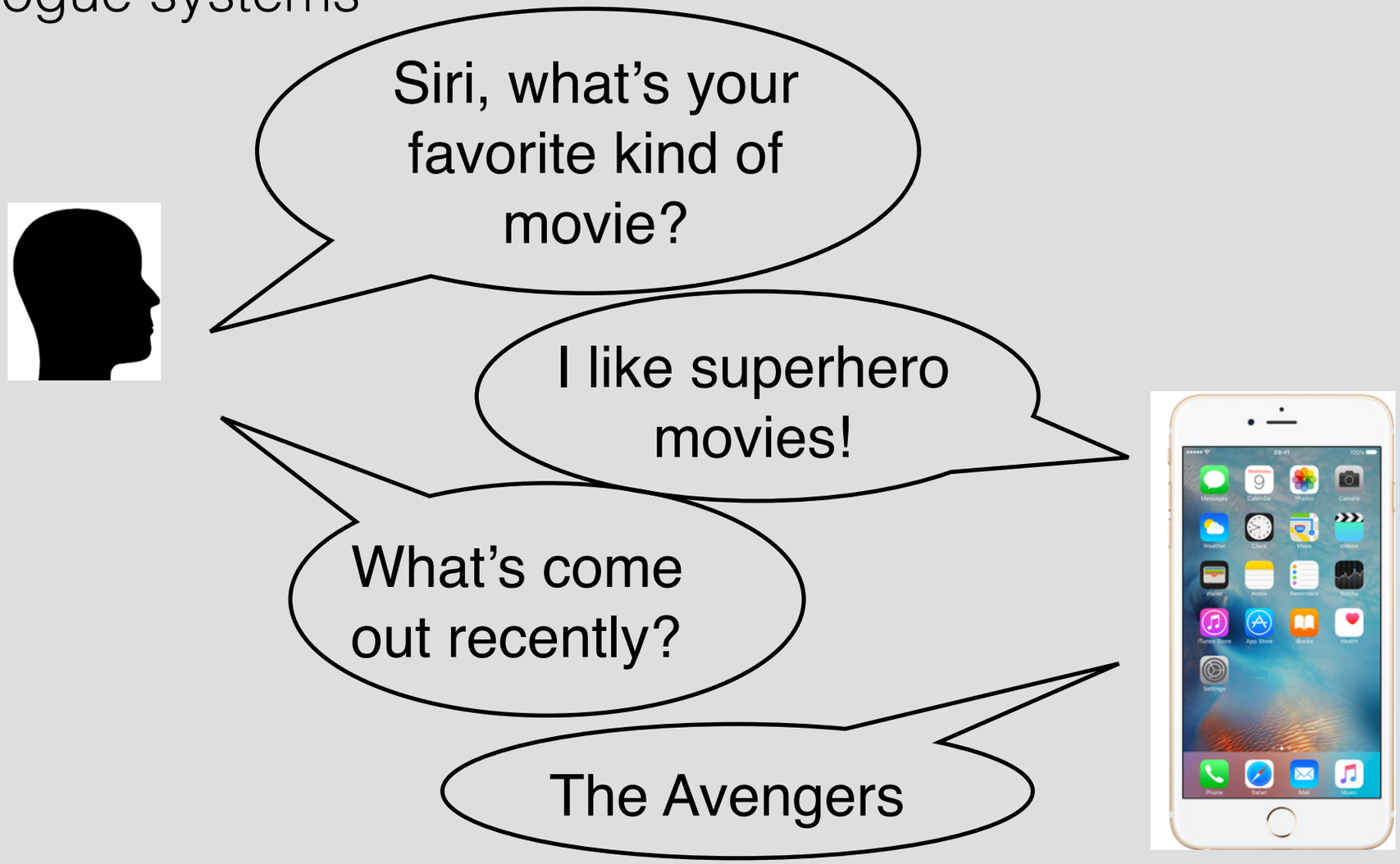
- Text Processing Basics

- Neural Network Basics

Why NLP?

What's the goal of NLP?

- Be able to solve problems that require **understanding** & **generation** of text
- Example: dialogue systems



Question Answering

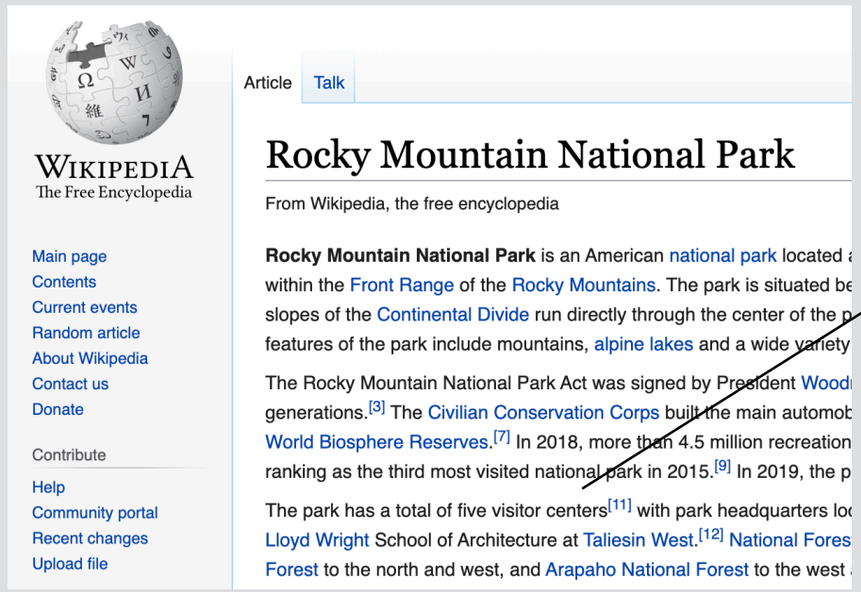
When was Abraham Lincoln born?

Name	Birthday
Lincoln, Abraham	2/12/1809
Washington, George	2/22/1732
Adams, John	10/30/1735

map to Birthday field

February 12, 1809

How many visitors centers are there in Rocky Mountain National Park?



"The park has a total of five visitor centers"

five

Machine Translation

The Political Bureau of the CPC Central Committee July 30 hold a meeting

中共中央政治局7月30日召开会议，会议分析研究当前经济形势，部署下半年经济工作。

People's Daily, August 10, 2020

The Political Bureau of the CPC Central Committee held a meeting on July 30 to analyze and study the current economic situation and plan economic work in the second half of the year.

Text Summarization

POLITICS

Google Critic Ousted From Think Tank Funded by the Tech Giant

WASHINGTON — In the hours after European antitrust regulators levied a record $2.7 billion fine against Google in late June, an influential Washington think tank learned what can happen when a tech giant that shapes public policy debates with its enormous wealth is criticized.

• • •

But not long after one of New America's scholars posted a statement on the think tank's website praising the European Union's penalty against Google, Mr. Schmidt, who had been chairman of New America until 2016, communicated his displeasure with the statement to the group's president, Anne-Marie Slaughter, according to the scholar.

• • •

Ms. Slaughter told Mr. Lynn that "the time has come for Open Markets and New America to part ways," according to an email from Ms. Slaughter to Mr. Lynn. The email suggested that the entire Open Markets team — nearly 10 full-time employees and unpaid fellows — would be exiled from New America.

compress text

provide missing context

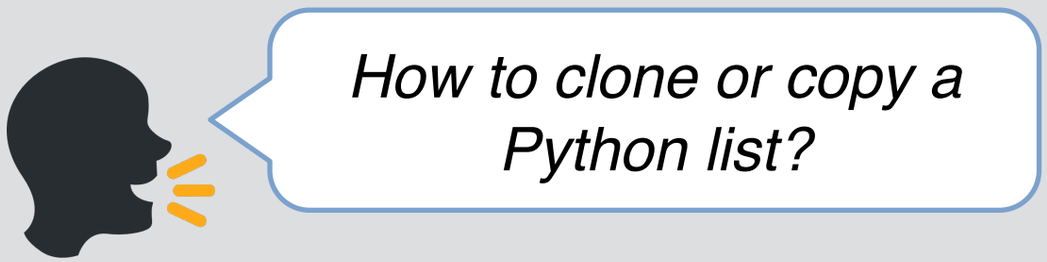
One of New America's writers posted a statement critical of Google. Eric Schmidt, Google's CEO, was displeased.

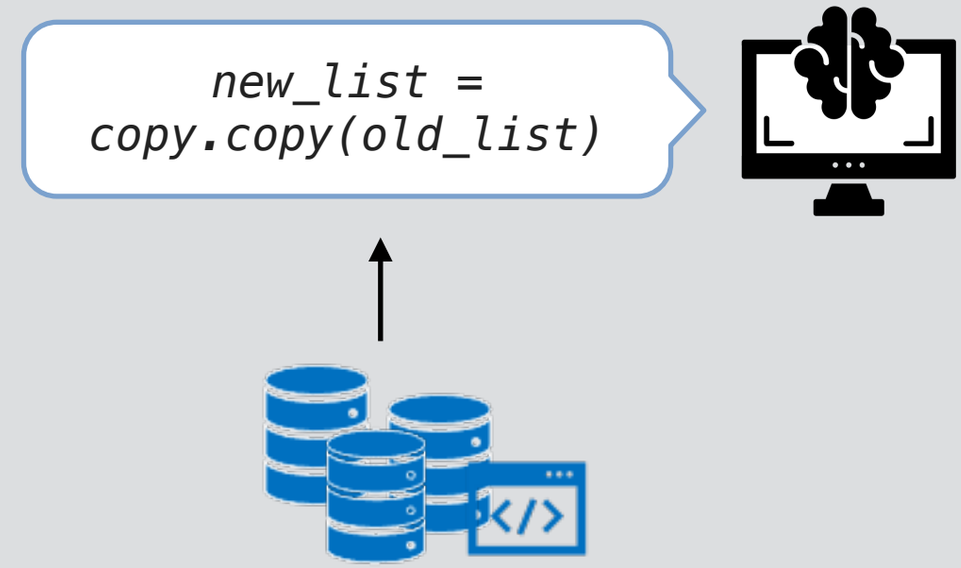
The writer and his team were dismissed.

paraphrase to provide clarity

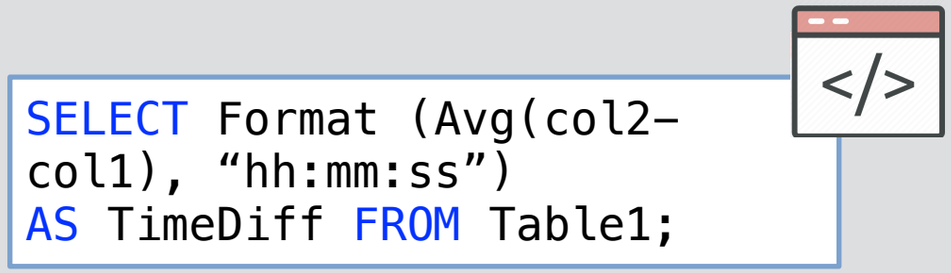
NLP for Programming

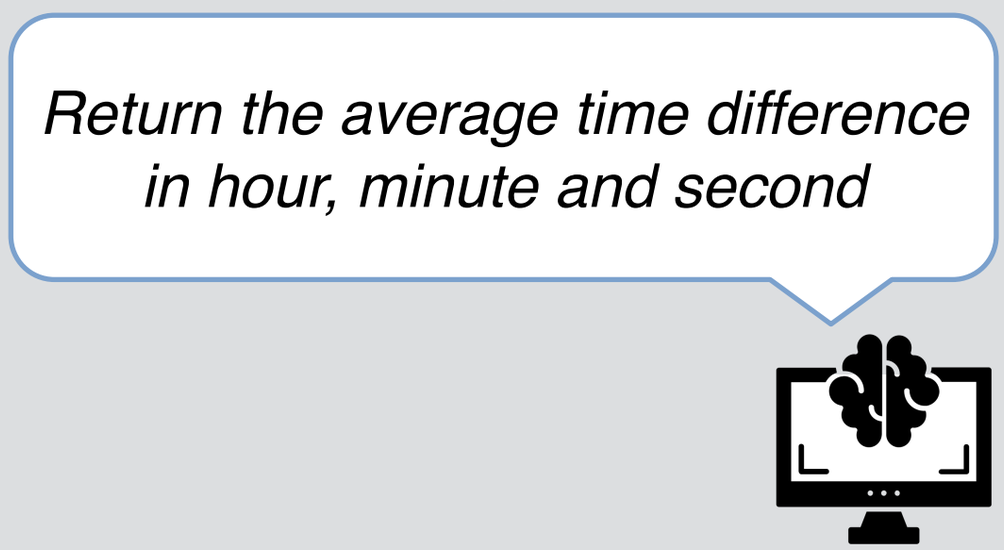
Code Retrieval





Code Annotation



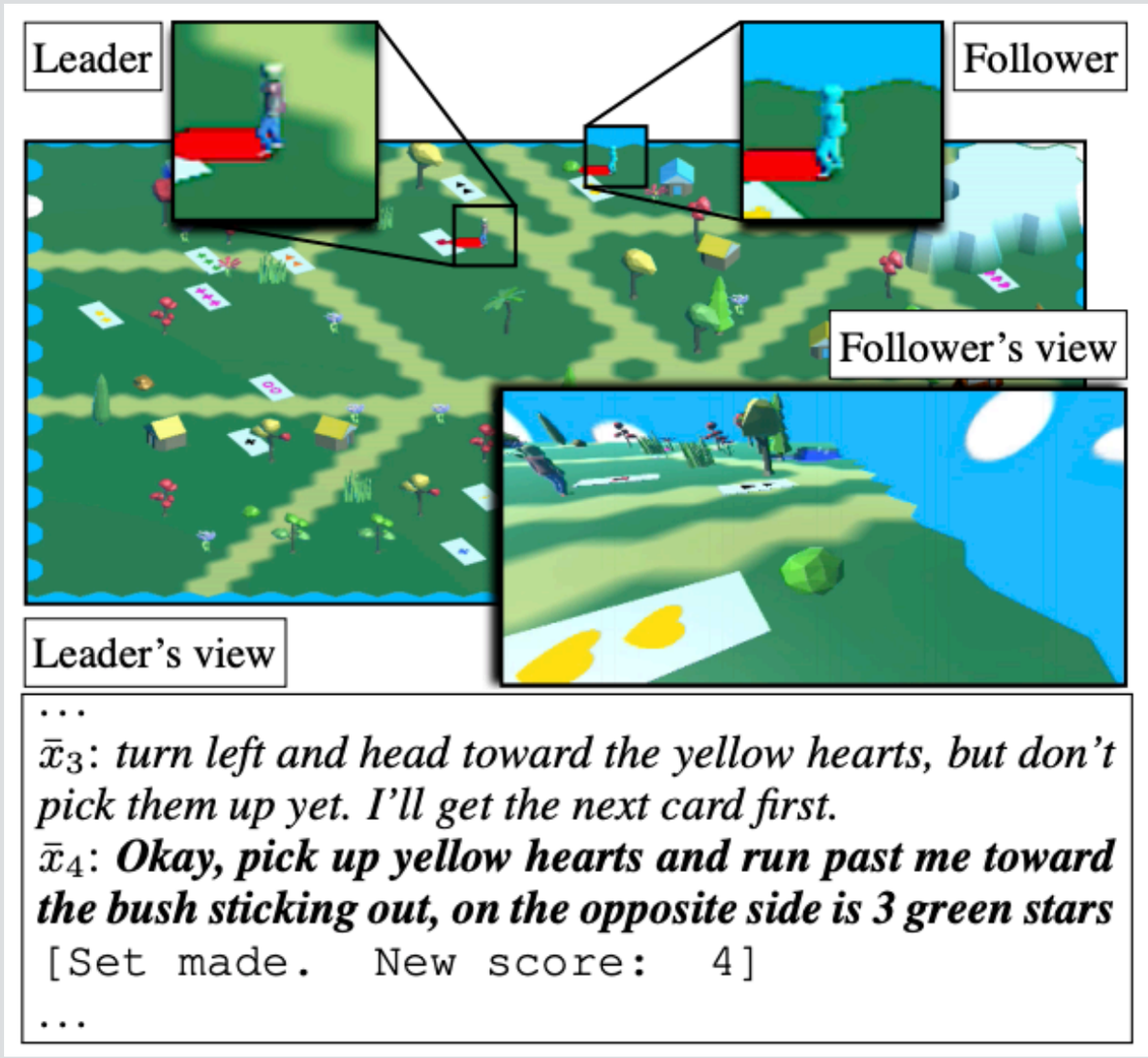


Yao et al., “StaQC: A Systematically Mined Question-Code Dataset from Stack Overflow”, WWW 2018

Allamanis, Miltiadis, et al. "A survey of machine learning for big code and naturalness." ACM Computing Surveys (CSUR) 51.4 (2018): 1-37.

Vision + Language

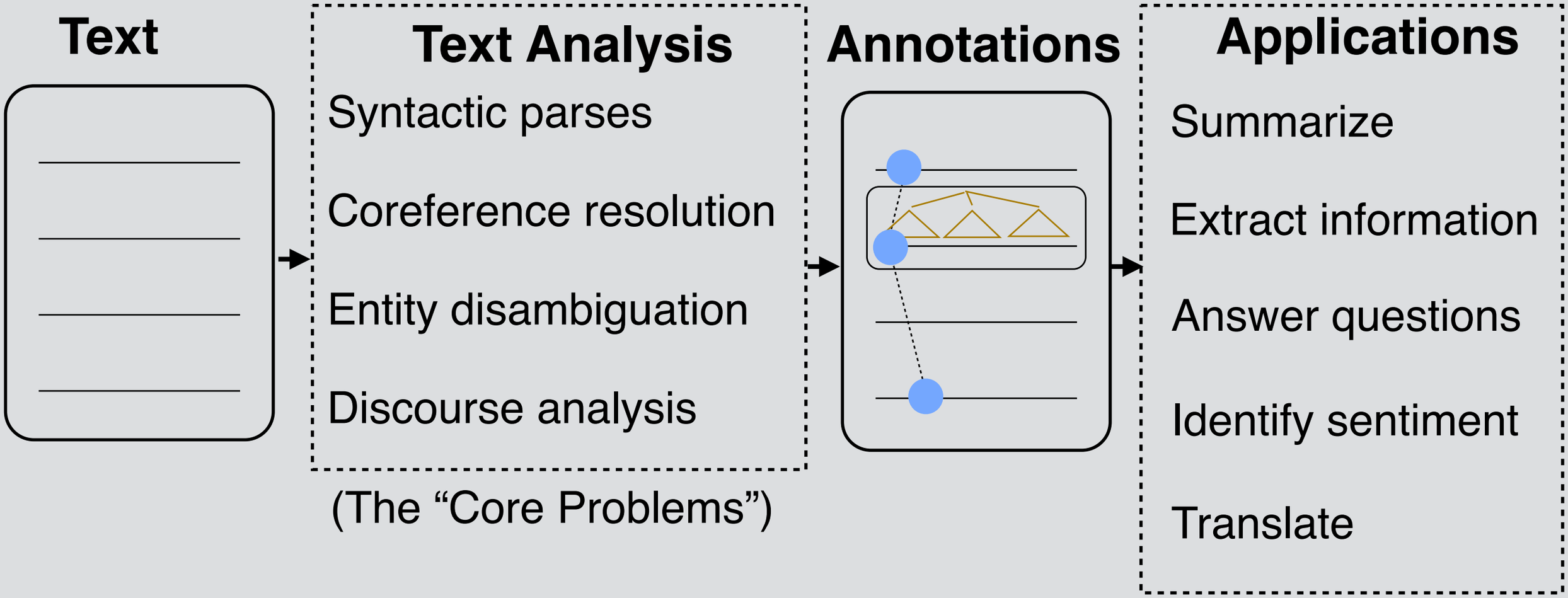
“Place a wet soap in the trash can”



Shridhar, Mohit, et al. "ALFRED: A Benchmark for Interpreting Grounded Instructions for Everyday Tasks." *CVPR*, 2020.
Suhr, Alane, et al. "Executing Instructions in Situated Collaborative Interactions." *Proceedings of EMNLP-IJCNLP,* 2019.

What exactly is NLP doing?

NLP Analysis Pipeline



- NLP is about building these pieces!
- All of these components are modeled with statistical approaches trained with machine learning

How do we represent language?

Text

Labels

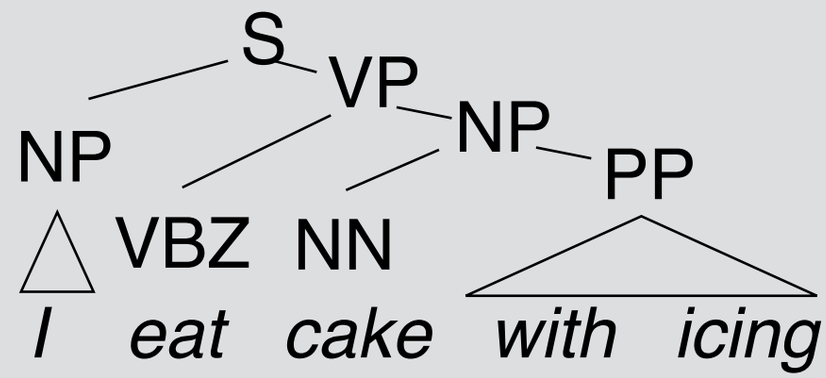
the movie was good +

Beyoncé had one of the best videos of all time Subjective

Sequences/tags

PERSON *Tom Cruise* *stars in the new* **WORK_OF_ART** *Mission Impossible* *film*

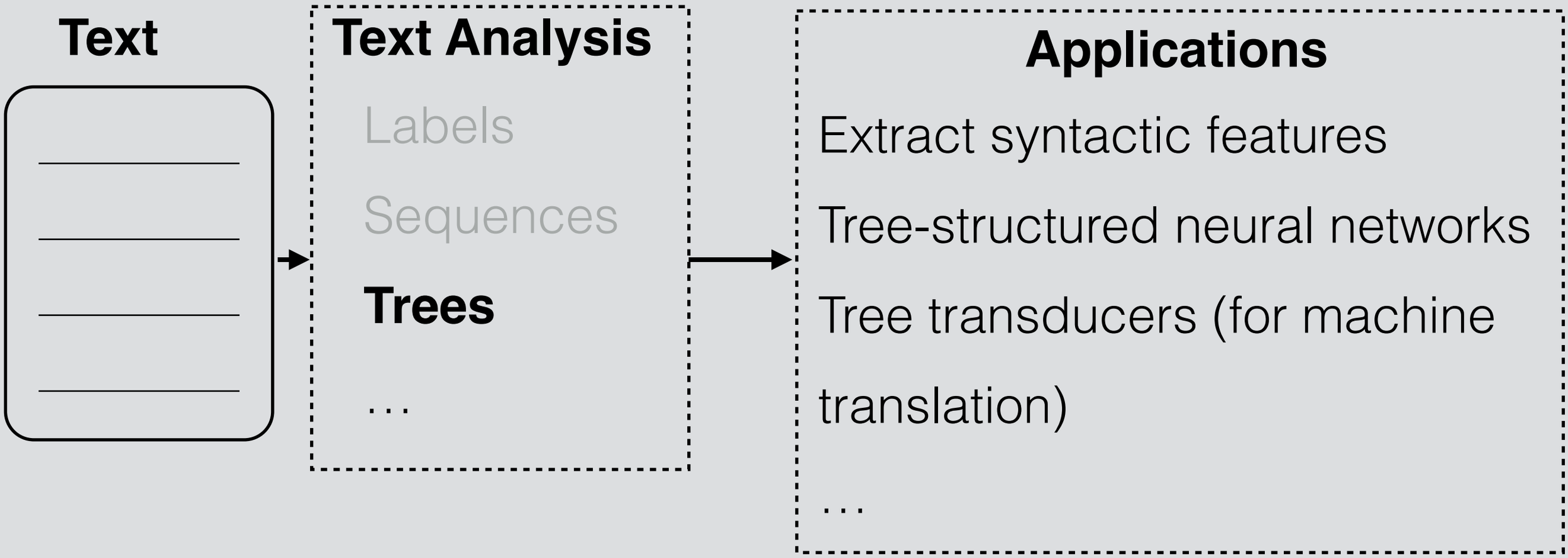
Trees



λx. flight(x) ∧ dest(x)=Miami

flights to Miami

How do we use these representations?



- Main question: What representations do we need for language? What do we want to know about it?
- Boils down to: what ambiguities do we need to resolve?

Language is Ambiguous!

Language is Ambiguous!

- Hector Levesque (2011): “Winograd schema challenge” (named after Terry Winograd, the creator of SHRDLU)

The city council refused the demonstrators a permit because they advocated violence

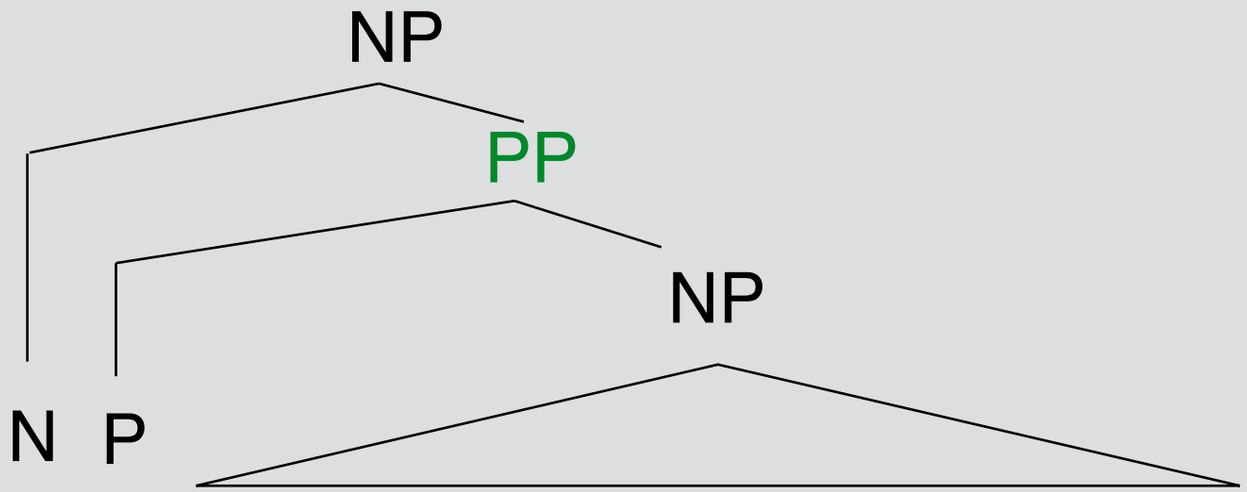
The city council refused the demonstrators a permit because they feared violence

The city council refused the demonstrators a permit because they _____ violence

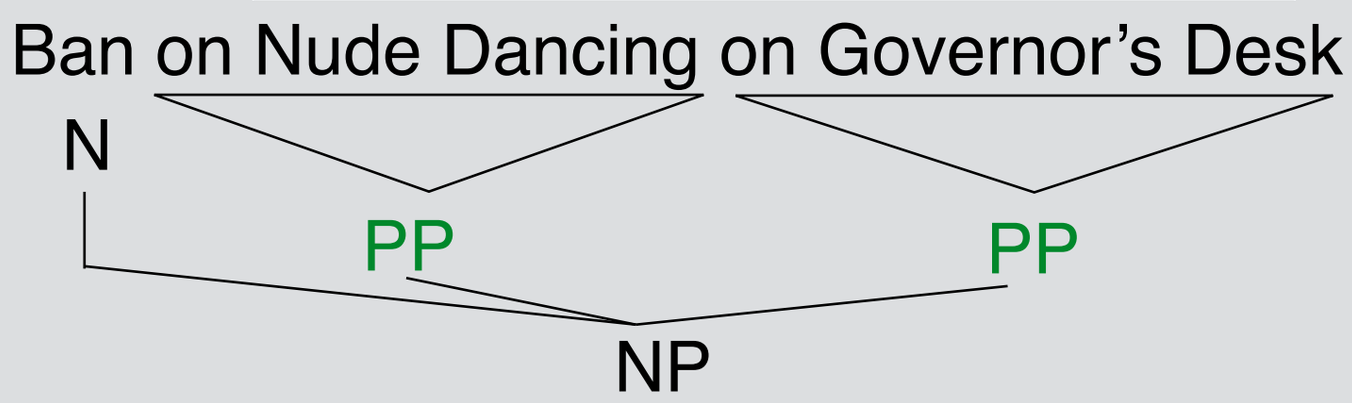
- >5 datasets in the last few years examining this problem and commonsense reasoning
- Referential ambiguity

Language is Ambiguous!

N N V N
N V ADJ N
Teacher Strikes Idle Kids



body/ position body/ weapon
Iraqi Head Seeks Arms



- Syntactic and semantic ambiguities: parsing needed to resolve these, but need context to figure out which parse is correct

Example credit: Dan Klein

Language is Ambiguous!

- Pragmatic ambiguity

Turn on the AC before I return home.

OK!

"If-Then" program:

- **Trigger**
  `Location.current_area(non-home)`
- **Action**
  `AirConditioner.switch_on()`

Healthcare, medicine, politics, business, etc.

What Do We Need to Understand Language?

Approaches to NLP

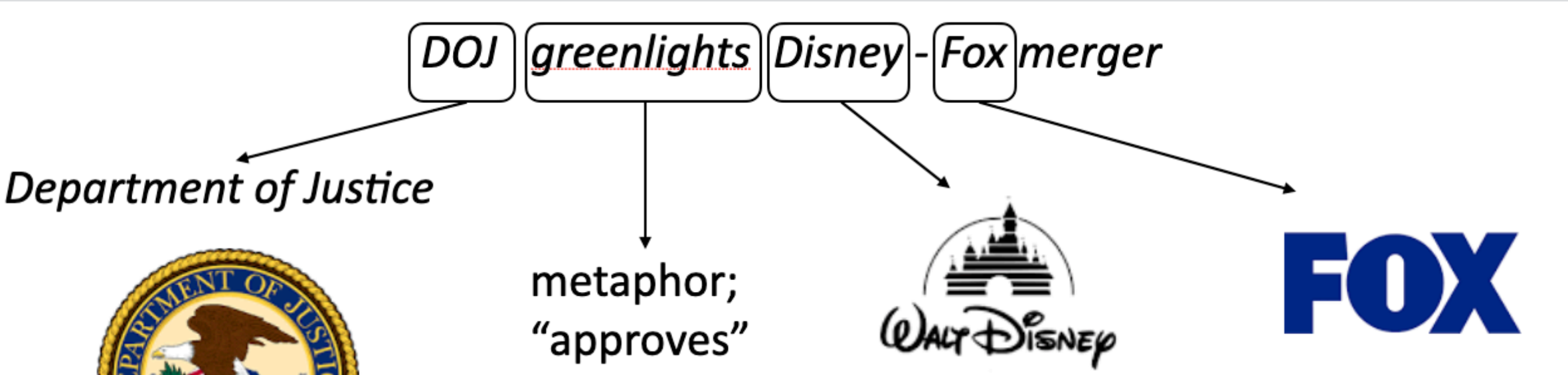
- Linguistics
  - Morphology, syntax, semantics, discourse, pragmatics

Approaches to NLP

- Linguistics
  - Morphology, syntax, semantics, discourse, pragmatics
- Learning
  - Statistical modeling -> machine learning -> deep neural nets (supervised, architecture search, etc.) -> large language models transfer learning (BERT, GPT-2/3, etc.)
  - Integration of world knowledge

World Knowledge

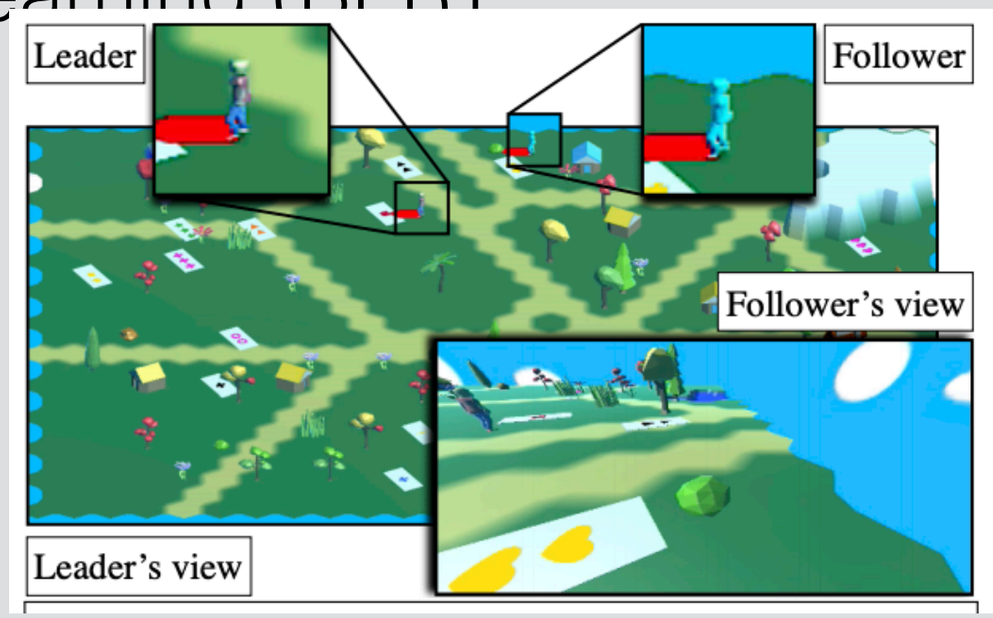
Beyond training data



- What is a green light? How do we understand what "green lighting" does?
- Need commonsense knowledge

Approaches to NLP

- Linguistics
  - Morphology, syntax, semantics, discourse, pragmatics
- Learning
  - Statistical modeling -> machine learning -> deep neural nets (supervised, architecture search, etc.) -> large language models transfer learning (BERT, GPT-2/3, etc.)
  - Integration of world knowledge
  - Grounding in the environment



Approaches to NLP

- Linguistics
  - Morphology, syntax, semantics, discourse, pragmatics
- Learning
  - Statistical modeling -> machine learning -> deep neural nets (supervised, architecture search, etc.) -> large language models transfer learning (BERT, GPT-2/3, etc.)
  - Integration of world knowledge
  - Grounding in the environment
  - …
- Linguistics + Learning

Interpreting NLP Models

The movie is mediocre, maybe even bad.	**Negative** 99.8%
The movie is mediocre, maybe even ~~bad~~.	**Negative** 98.0%
The movie is ~~mediocre~~, maybe even bad.	**Negative** 98.7%
The movie is ~~mediocre~~, maybe even ~~bad~~.	**Positive** 63.4%
The movie is ~~mediocre~~, ~~maybe~~ even ~~bad~~.	**Positive** 74.5%
The ~~movie~~ is mediocre, maybe even ~~bad~~.	**Negative** 97.9%

The movie is mediocre, maybe even bad.

- Computational Social Science and Cultural Analytics
- Dialogue and Interactive Systems
- Discourse and Pragmatics
- Ethics and NLP
- Information Extraction
- Information Retrieval and Text Mining
- Interpretability and Analysis of Models for NLP
- Language Grounding to Vision, Robotics and Beyond
- Linguistic theories, Cognitive Modeling and Psycholinguistics
- Machine Learning for NLP
- Machine Translation and Multilinguality
- NLP Applications
- Phonology, Morphology and Word Segmentation
- Question Answering
- Resources and Evaluation
- Semantics: Lexical
- Semantics: Sentence-level Semantics, Textual Inference and Other areas
- Sentiment Analysis, Stylistic Analysis, and Argument Mining
- Speech and Multimodality
- Summarization
- Syntax: Tagging, Chunking and Parsing

Any *ACL conferences, e.g., https://2021.aclweb.org/calls/papers/

Outline for Today

- Class logistics
- Why NLP, and What is it?
- Scope of this course
- Text Processing Basics
- Neural Network Basics

Class Plan

Neural Nets Basics

Neural Nets Architectures and Representations

Principles in NLP Model Development

Structured Predictions

Date	Topic
08/25	Introduction and Class Outline; Projects and Paper Presentations; Neural Network Basics
09/01	Word Embeddings; Binary/Multiclass Classification; NN Basics (continued: FFNN)
09/08	Language Modeling: n-gram models and Recurrent Neural Networks
09/15	Distributional Semantics, and Contextual Representations (ELMo, BERT)
09/22	Lang Generation; NN Architectures: Encoder-Decoder, Attention, and Transformers
09/29	Experimental Design; Interpreting and debug-ging NLP models
10/06	Sequence Labeling: HMM & CRF, Syntactic Parsing 1
10/13	Syntactic Parsing 2; Semantic Parsing

10/20	Machine Translation
10/27	Pre-training & Prompting
11/03	Question Answering
11/10	Language Generation (Dialog, Summarization, etc.)
11/17	Human-centered NLP: Multilinguality, Ethics, and Interactivity
11/24	**NO CLASS**
12/01	Final Project Presentations

Advanced NLP Topics

Final Presentation!

Suggested Project Topics: https://nlp.cs.gmu.edu/course/cs678-fall22/projectideas/

Paper List for Presentation: TBD

Outline for Today

- Class logistics

- Why NLP, and What is it?

- Scope of this course

- Text Processing Basics

- Neural Network Basics

BREAK

Text Processing Basics

Some Concepts

- Text Processing
  - Word tokenization
  - Word normalization: Lemmatization, Stemming
  - Vocabulary construction
  - Word frequency, unknown words

Word Tokenization

- Tokenization: segmenting running text into words

I'm very excited to learning NLP in this semester!

HOW?

Word Tokenization

- Tokenization: segmenting running text into words

I'm very excited to learning NLP in this semester!

Whitespace split

["I'm", 'very', 'excited', 'to', 'learn', 'NLP', 'in', 'this', 'semester!']

contraction

punctuation

Does it look right?

Word Tokenization

- Tokenization: segmenting running text into words
- How about:
  - Abbreviations, e.g., D.C.?
  - Names consisting of multiple words, e.g., New York?
  - Hyphen connected names, e.g., co-occur?

Word Tokenization

- Tokenization: segmenting running text into words

I'm very excited to learning NLP in this semester!

Penn Treebank
tokenization

['I', "'m", 'very', 'excited', 'to', 'learn', 'NLP', 'in', 'this', 'semester', '!']

Word Tokenization

- Tokenization: segmenting running text into words

I went to D.C. last month *['I', 'went', 'to', 'D.C.', 'last', 'month']*

New York City, or NYC for short...

['New', 'York', 'City', ',', 'or', 'NYC', 'for', 'short', '...']

Can the two words co-occur?

['Can', 'the', 'two', 'words', 'co-occur', '?']

Word Tokenization

- Tokenization: segmenting running text into words
- Commonly used tool: Python NLTK package
  - https://www.nltk.org/api/nltk.tokenize.html

```
>>> from nltk.tokenize import word_tokenize
>>> s = '''Good muffins cost $3.88\nin New York.  Please buy me
... two of them.\n\nThanks.'''
>>> word_tokenize(s)
['Good', 'muffins', 'cost', '$', '3.88', 'in', 'New', 'York', '.',
'Please', 'buy', 'me', 'two', 'of', 'them', '.', 'Thanks', '.']
```

Word Normalization

- Normalization: converting words to a standard form
- Why? e.g., if a search engine has learned *"cat"*, then it should also answer questions about *"cats"*
- Two methods:
  - Lemmatization: return the word "lemma", i.e., the "root" of a word, despite its surface form
  - Stemming: return the word "stem", which chops off the word affixes
    - A simpler version of lemmatization

Lemmatization

- Words could share the same “root”, despite their surface differences

- Examples:
  - am, are, is, was, were —> be
  - cat, cats —> cat
  - stories —> story
  - reading —> read
  - sing, sang, sung —> sing
  - people, person —> person

Lemmatization

- Words could share the same “root”, despite their surface differences

- Examples:

- Morphological parsing

  - Morphology: study how words are built up from smaller meaning-bearing units called “morphemes”

  - cats = cat (“stem”) + s (“affix”)

Stemming

- Return the “stem” of a word
  - Note that a stemmer may not return a valid word
  - One widely used algorithm: Porter Stemming (Porter, 1980)

- Examples:
  - stories —> stori
  - accurate —> accuri
  - Reading —> read
  - am, are, is, was, were —> am, are, is, wa, were
  - single, people —> singl, peopl

Lemmatization/Stemming

- Tools (in Python):
  - NLTK's Porter/Snowball Stemmer
  - NLTK's WordNetLemmatizer
  - SpaCy's word lemmatize function
- A demo website: https://textanalysisonline.com/

Word Normalization

- Normalization: converting words to a standard form
- Why? e.g., if a search engine has learned *“cat”*, then it should also answer questions about *“cats”*
- Two methods: Lemmatization and Stemming
- Others: case folding (i.e., mapping every word to its lowercase from), or application-specific normalization (e.g., normalizing #hashtags for tweets)
- Note: nowadays for most NLP tasks we don’t use them

Vocabulary

- Word type: distinct words in a corpus
- Word token: running occurrence of a word type
- Vocabulary: a set of word types (i.e., *distinct* word tokens)
  - Vocabulary size: the number of word types
- Word (type) frequency: the number of occurrences (tokens) of that type in a corpus

Construct Vocabulary

A very large number of published documents contain text only. They often look boring, and they are often written in obscure language, using mile-long sentences and cryptic technical terms, using one font only, perhaps even without headings. Such style, or lack of style, might be the one you are strongly expected to follow when writing eg scientific or technical reports, legal documents, or administrative papers. It is natural to think that such documents would benefit from a few illustrative images. (However, just adding illustration might be rather useless, if the text remains obscure and unstructured.)

Construct Vocabulary

a very large number of published documents contain text only . they often look boring , and they are often written in obscure language , using mile-long sentences and cryptic technical terms , using one font only , perhaps even without headings . such style, or lack of style, might be the one you are strongly expected to follow when writing eg scientific or technical reports , legal documents , or administrative papers . it is natural to think that such documents would benefit from a few illustrative images . (however , just adding illustration might be rather useless , if the text remains obscure and unstructured .)

truecase + tokenize

Construct Vocabulary

a very large number of published documents contain text only . they often look boring , and they are often written in obscure language , using mile-long sentences and cryptic technical terms , using one font only , perhaps even without headings . such style, or lack of style, might be the one you are strongly expected to follow when writing eg scientific or technical reports , legal documents , or administrative papers . it is natural to think that such documents would benefit from a few illustrative images . (however , just adding illustration might be rather useless , if the text remains obscure and unstructured .)

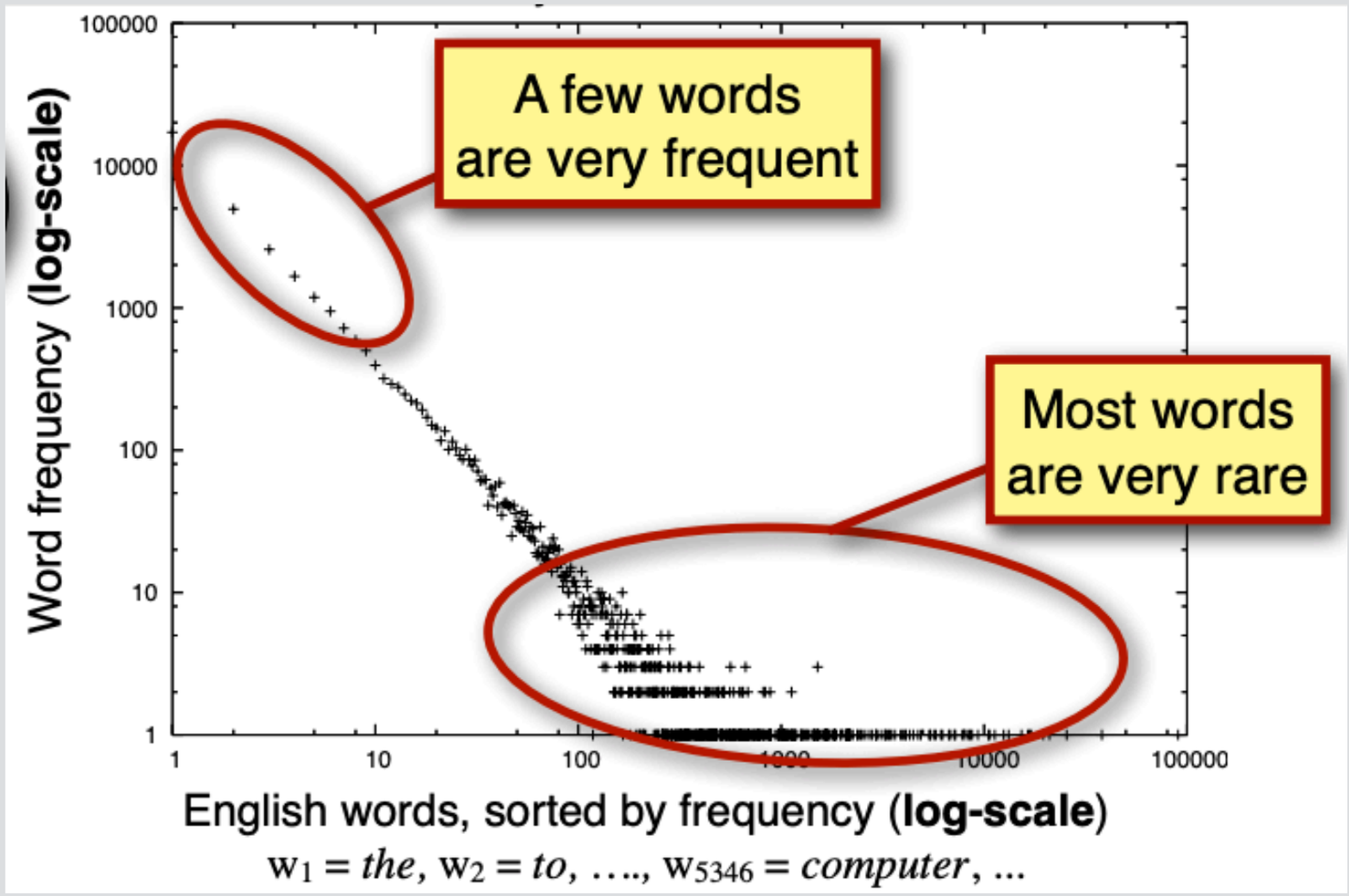
count the word type, and sort by frequency

vocabulary (typically after merging rare words)

,	10
.	5
documents	3
and	3
or	3
a	2
of	2
text	2
only	2
...	

Zipf's Law

- How many words occur more/less frequently?



https://en.wikipedia.org/wiki/Zipf%27s_law;
Figure from https://courses.engr.illinois.edu/cs447/fa2020/Slides/Lecture02.pdf

Dealing with Unknown Words

- What if the text analyzer (e.g., a text classifier) meets a word which it has never seen before? e.g.,
  - Distribution shift: words in the deployment time (the **test** corpus) did not occur when building the analyzer (the **training** corpus)
  - Limited capacity: when built with a limited vocabulary size due to restrictions on compute power
- Solutions:
  - Replace all rare words in the training corpus with an UNK token, and treat UNK the same as other regular tokens
  - Subword/character tokenization, e.g., Byte Pair Encoding
    - Often used in neural models

Outline for Today

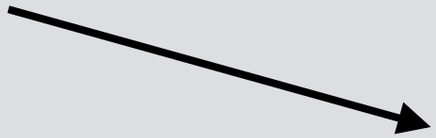
- Class logistics
- Why NLP, and What is it?
- Scope of this course
- Text Processing Basics
- Neural Network Basics

Neural Networks:

A Tool for Doing Hard Things

An Example Prediction Problem: Sentence Classification

I hate this movie

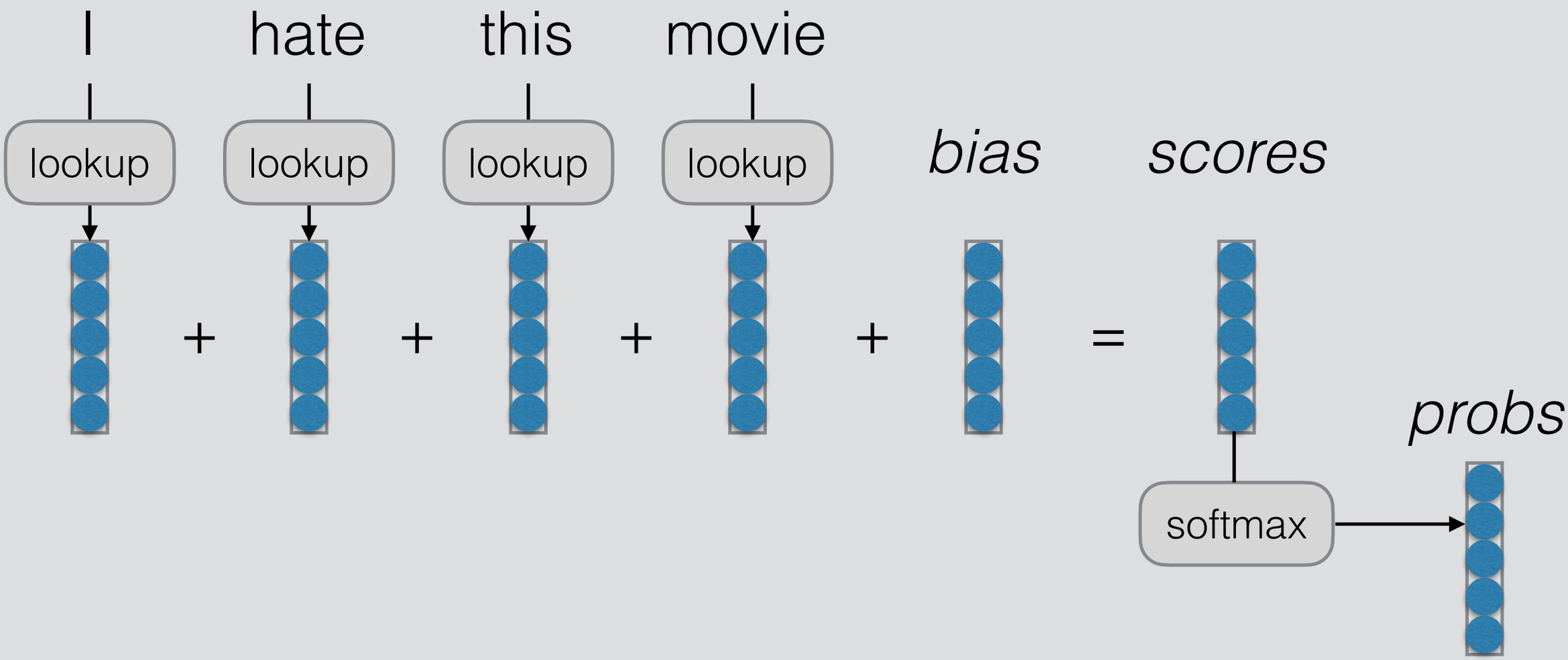


very good
good
neutral
bad
very bad

I love this movie

very good
good
neutral
bad
very bad

A First Try:
Bag of Words (BOW)



What do Our Vectors Represent?

- Each word has its own 5 elements corresponding to [very good, good, neutral, bad, very bad]

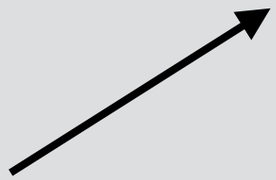
- “hate” will have a high value for “very bad”, etc.

Build It, Break It

I don’t love this movie

very good
good
neutral
bad
very bad

There’s nothing I don’t love about this movie



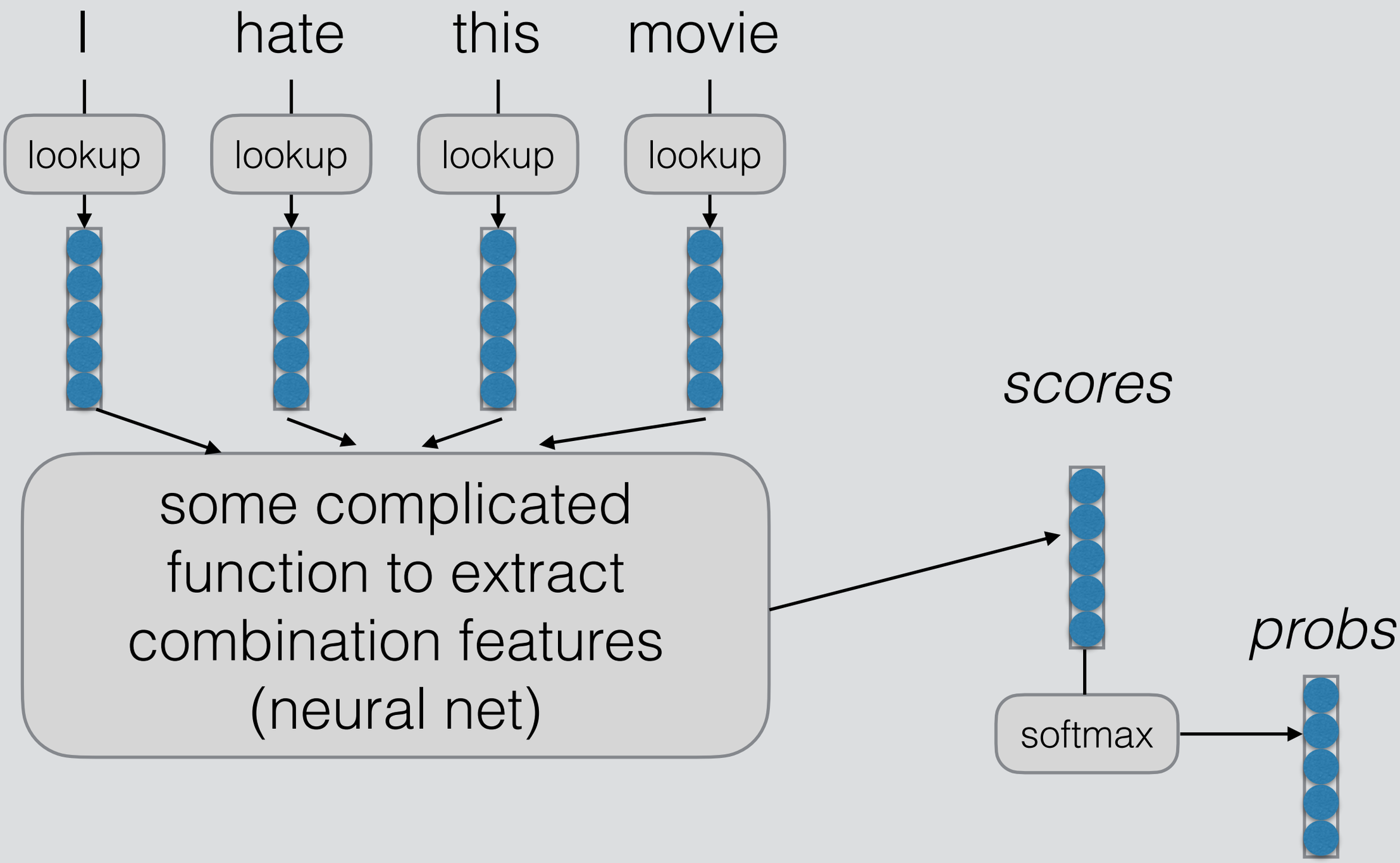
very good
good
neutral
bad
very bad

Combination Features

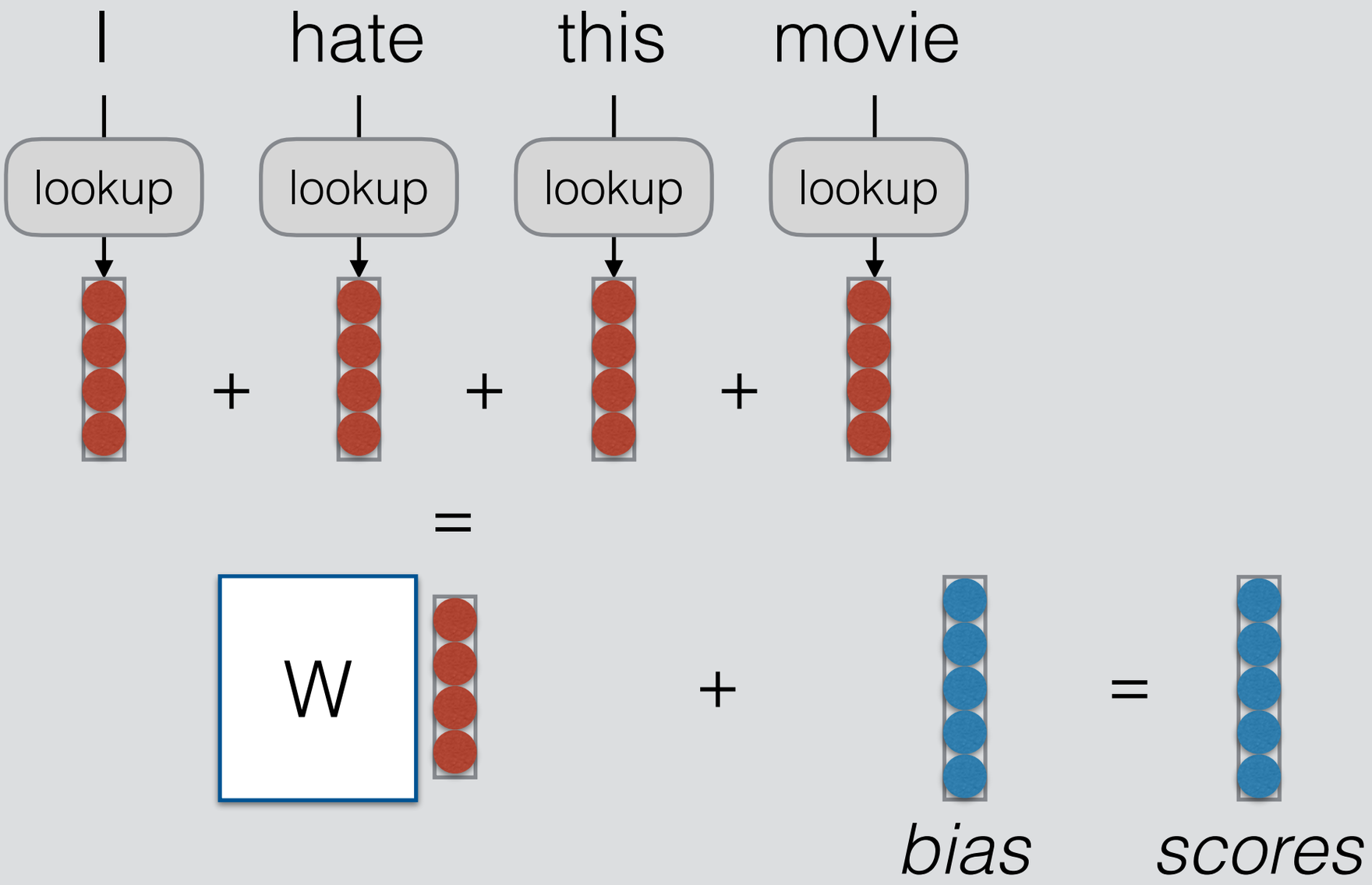
- Does it contain “don’t” and “love”?

- Does it contain “don’t”, “i”, “love”, and “nothing”?

Basic Idea of Neural Networks (for NLP Prediction Tasks)



Continuous Bag of Words (CBOW)



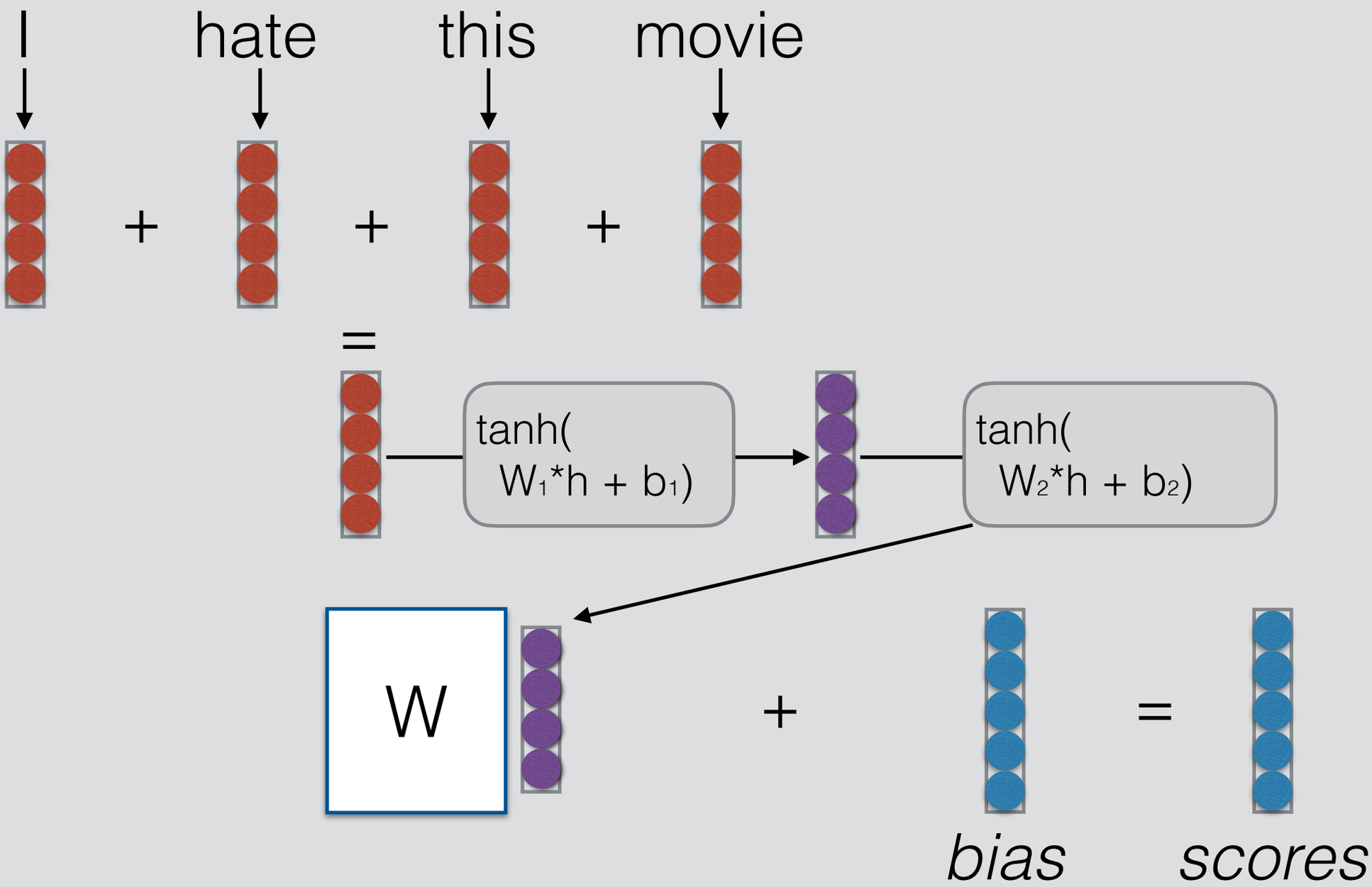
What do Our Vectors Represent?

- Each vector has “features” (e.g. is this an animate object? is this a positive word, etc.)

- We sum these features, then use these to make predictions

- Still no combination features: only the expressive power of a linear model, but dimension reduced

Deep CBOW



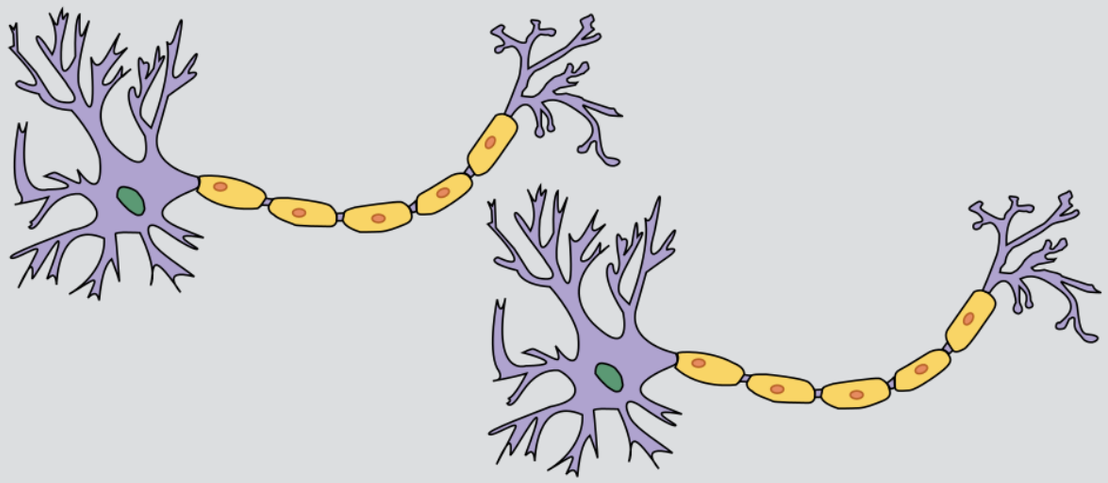
What do Our Vectors Represent?

- Now things are more interesting!
- We can learn feature combinations (a node in the second layer might be “feature 1 AND feature 5 are active”)
- e.g. capture things such as “not” AND “hate”

What is a Neural Net?: Computation Graphs

“Neural” Nets

Original Motivation: Neurons in the Brain



Current Conception: Computation Graphs

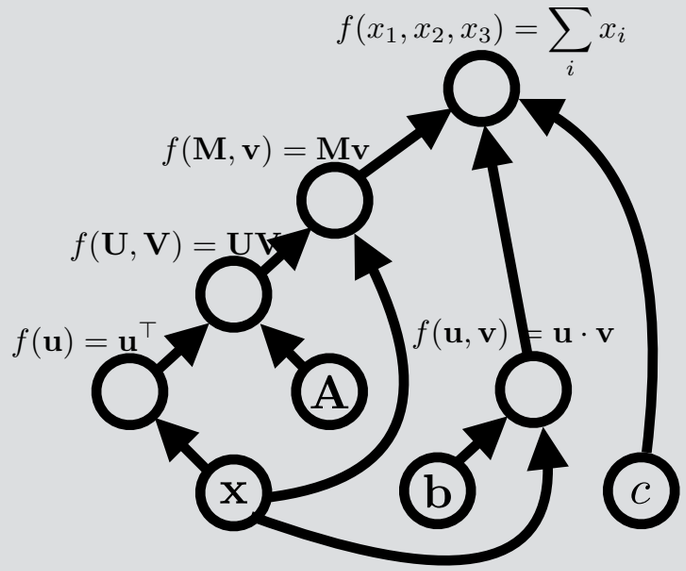


Image credit: Wikipedia

expression:

$$\mathbf{x}$$

graph:

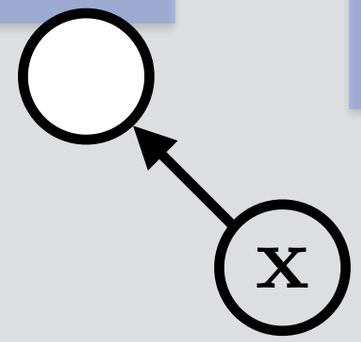
A **node** is a {tensor, matrix, vector, scalar} value

An **edge** represents a function argument (and also an data dependency). They are just pointers to nodes.

A **node** with an incoming **edge** is a **function** of that edge's tail node.

A **node** knows how to compute its value and the *value of its derivative w.r.t each argument (edge) times a derivative of an arbitrary input* $\frac{\partial \mathcal{F}}{\partial f(\mathbf{u})}$.

$f(\mathbf{u}) = \mathbf{u}^\top$



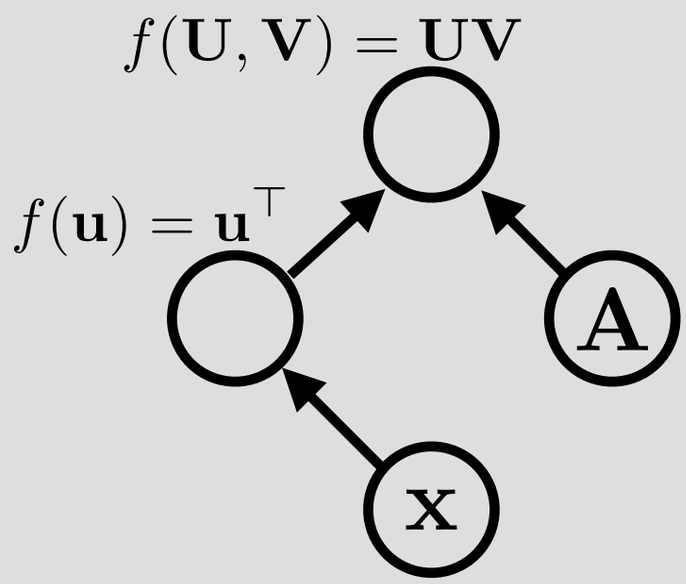
$$\frac{\partial f(\mathbf{u})}{\partial \mathbf{u}} \frac{\partial \mathcal{F}}{\partial f(\mathbf{u})} = \left(\frac{\partial \mathcal{F}}{\partial f(\mathbf{u})} \right)^\top$$

expression:

$$\mathbf{x}^\top \mathbf{A}$$

graph:

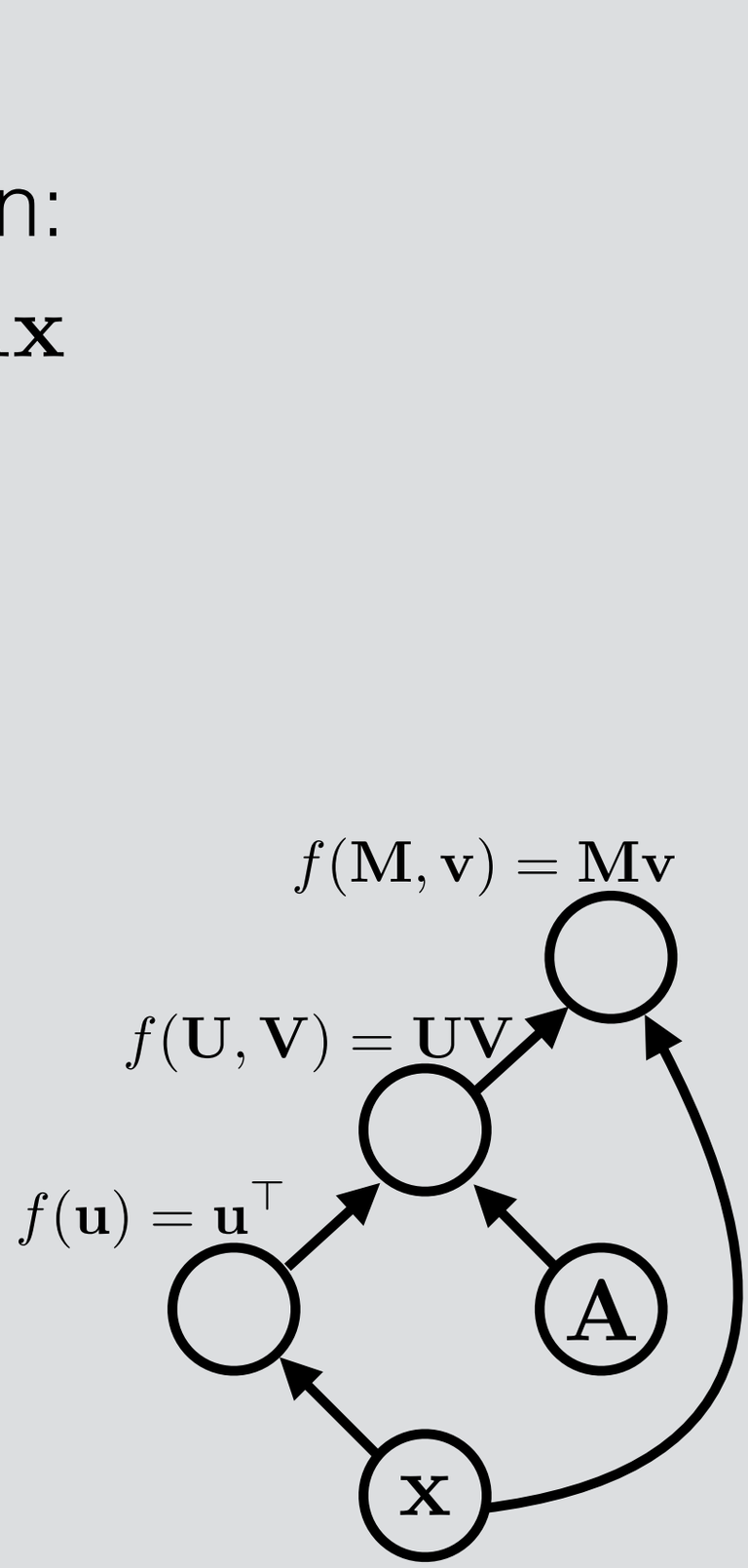
Functions can be nullary, unary,
binary, … *n*-ary. Often they are unary or binary.



expression:

$$\mathbf{x}^\top \mathbf{A} \mathbf{x}$$

graph:

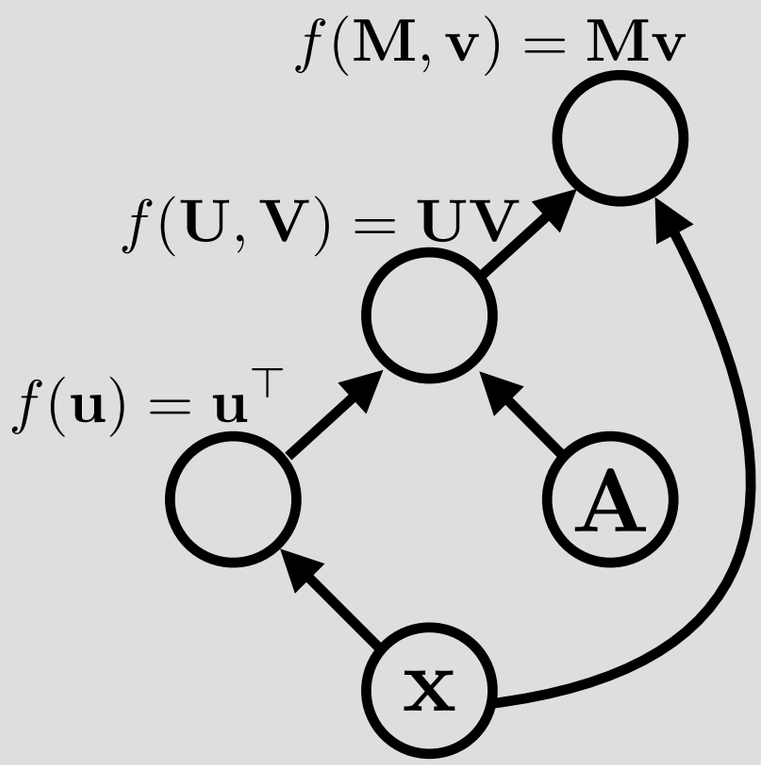


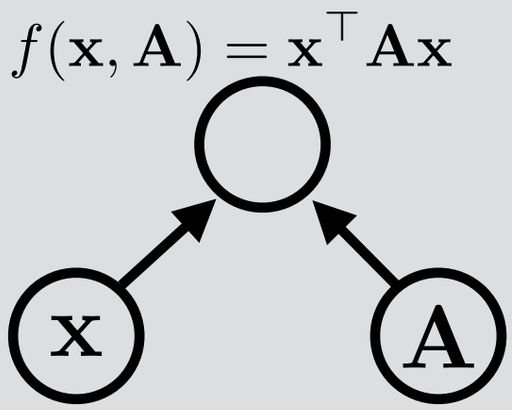
Computation graphs are directed and acyclic (in DyNet)

expression:

$$\mathbf{x}^\top \mathbf{A}\mathbf{x}$$

graph:





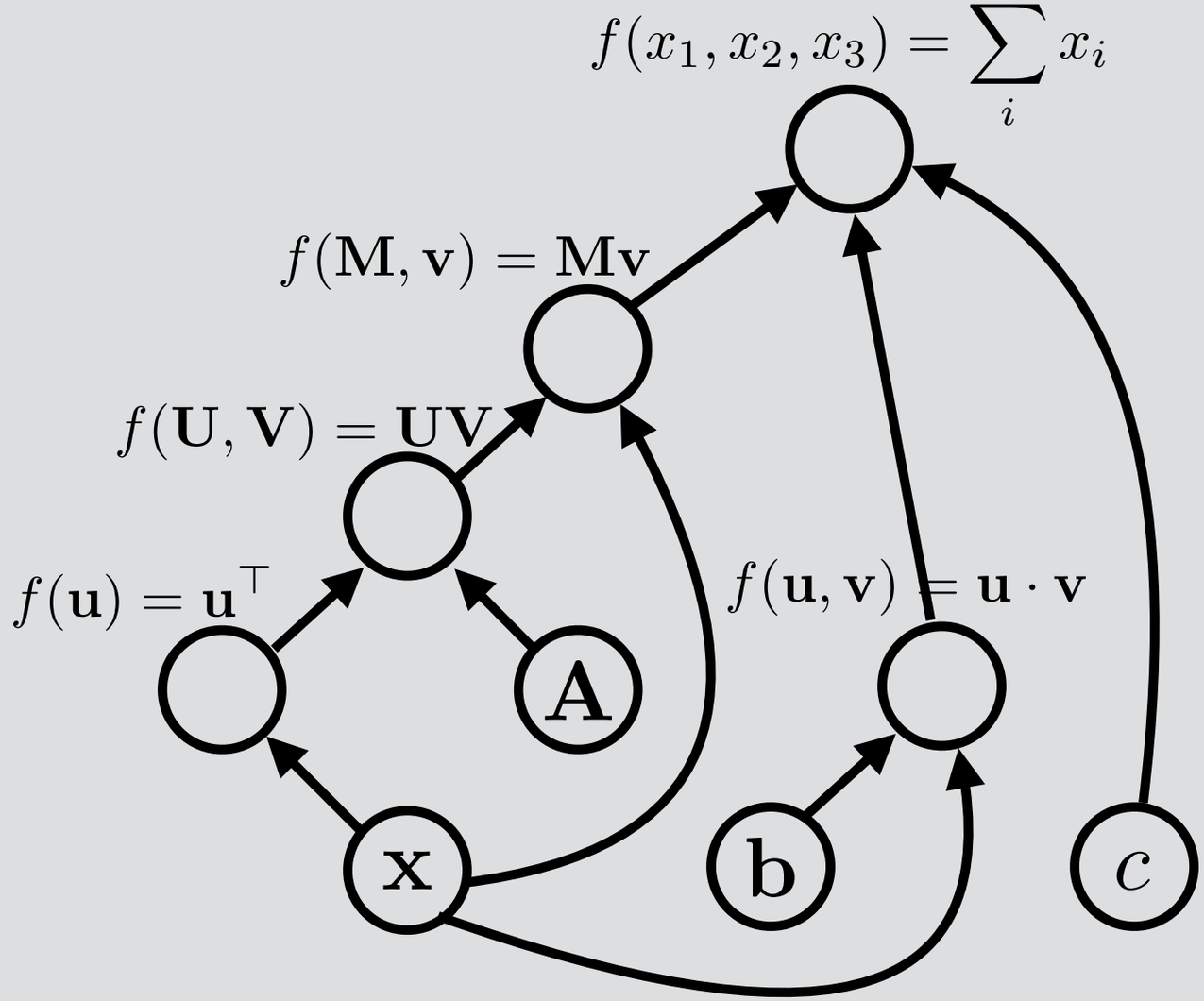
$$\frac{\partial f(\mathbf{x}, \mathbf{A})}{\partial \mathbf{x}} = (\mathbf{A}^\top + \mathbf{A})\mathbf{x}$$

$$\frac{\partial f(\mathbf{x}, \mathbf{A})}{\partial \mathbf{A}} = \mathbf{x}\mathbf{x}^\top$$

expression:

$$\mathbf{x}^\top \mathbf{A}\mathbf{x} + \mathbf{b} \cdot \mathbf{x} + c$$

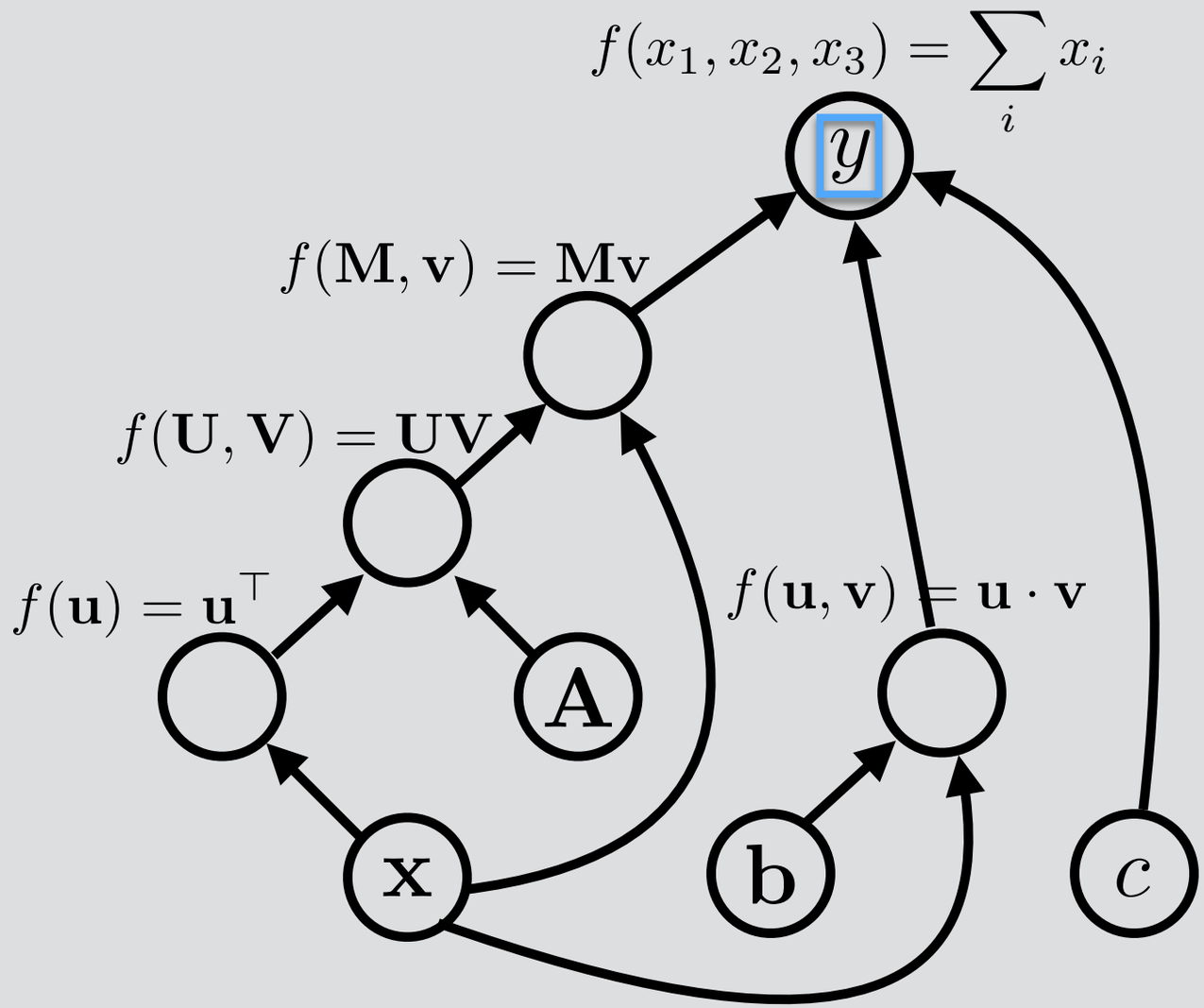
graph:



expression:

$$\boxed{y =} \mathbf{x}^\top \mathbf{A}\mathbf{x} + \mathbf{b} \cdot \mathbf{x} + c$$

graph:



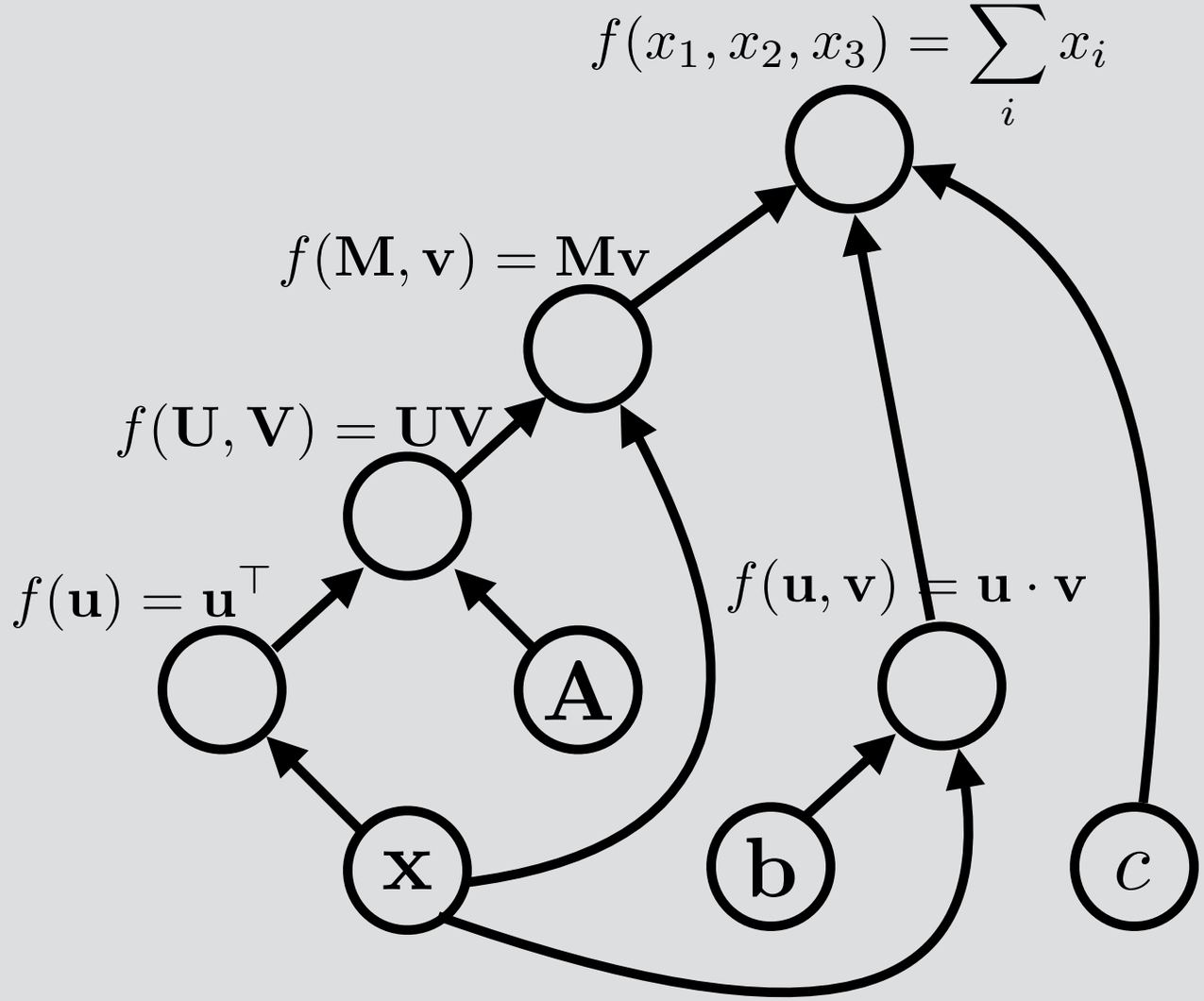
variable names are just labelings of nodes.

Algorithms (1)

- **Graph construction**
- **Forward propagation**
  - In topological order, compute the **value** of the node given its inputs

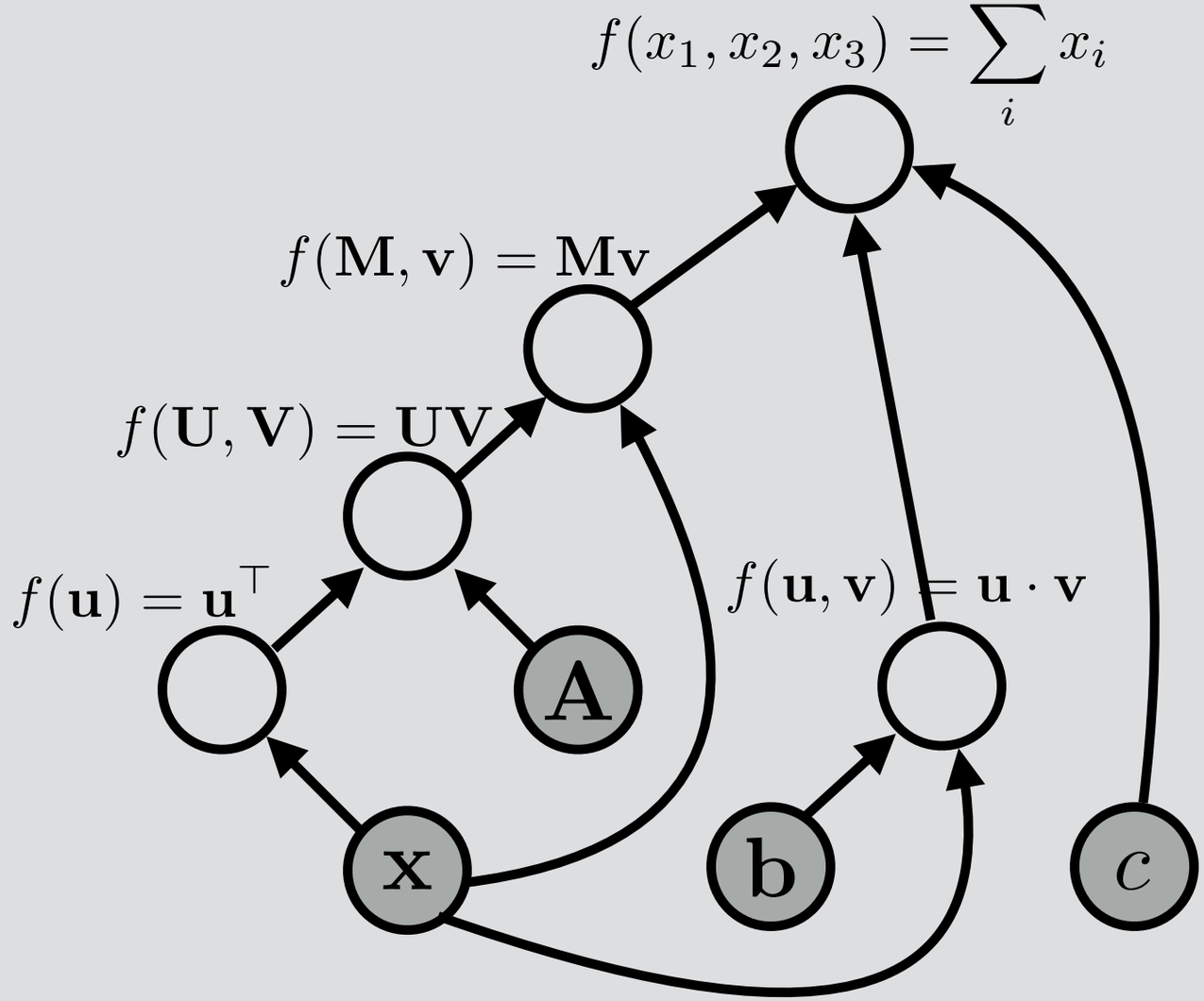
Forward Propagation

graph:



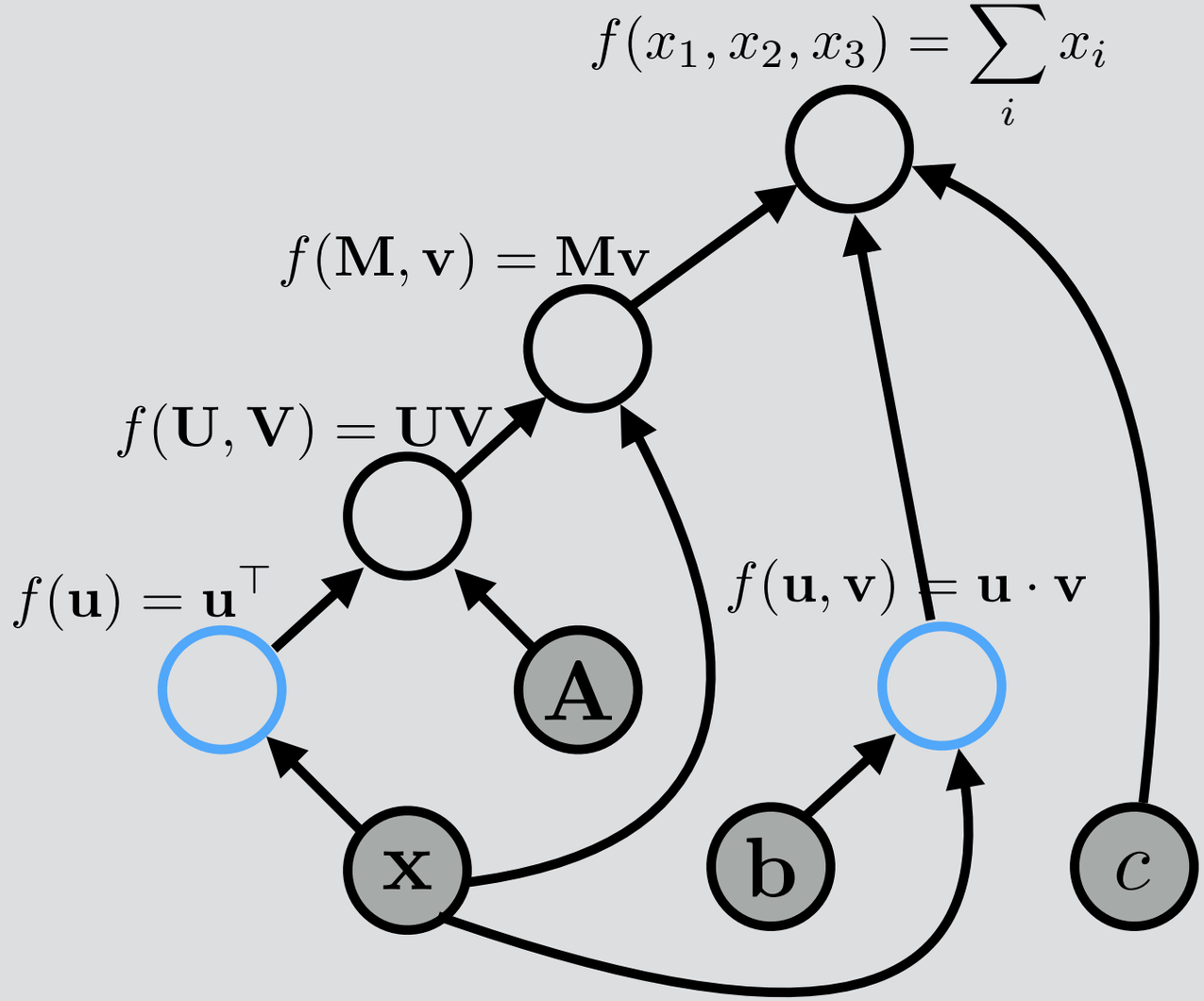
Forward Propagation

graph:



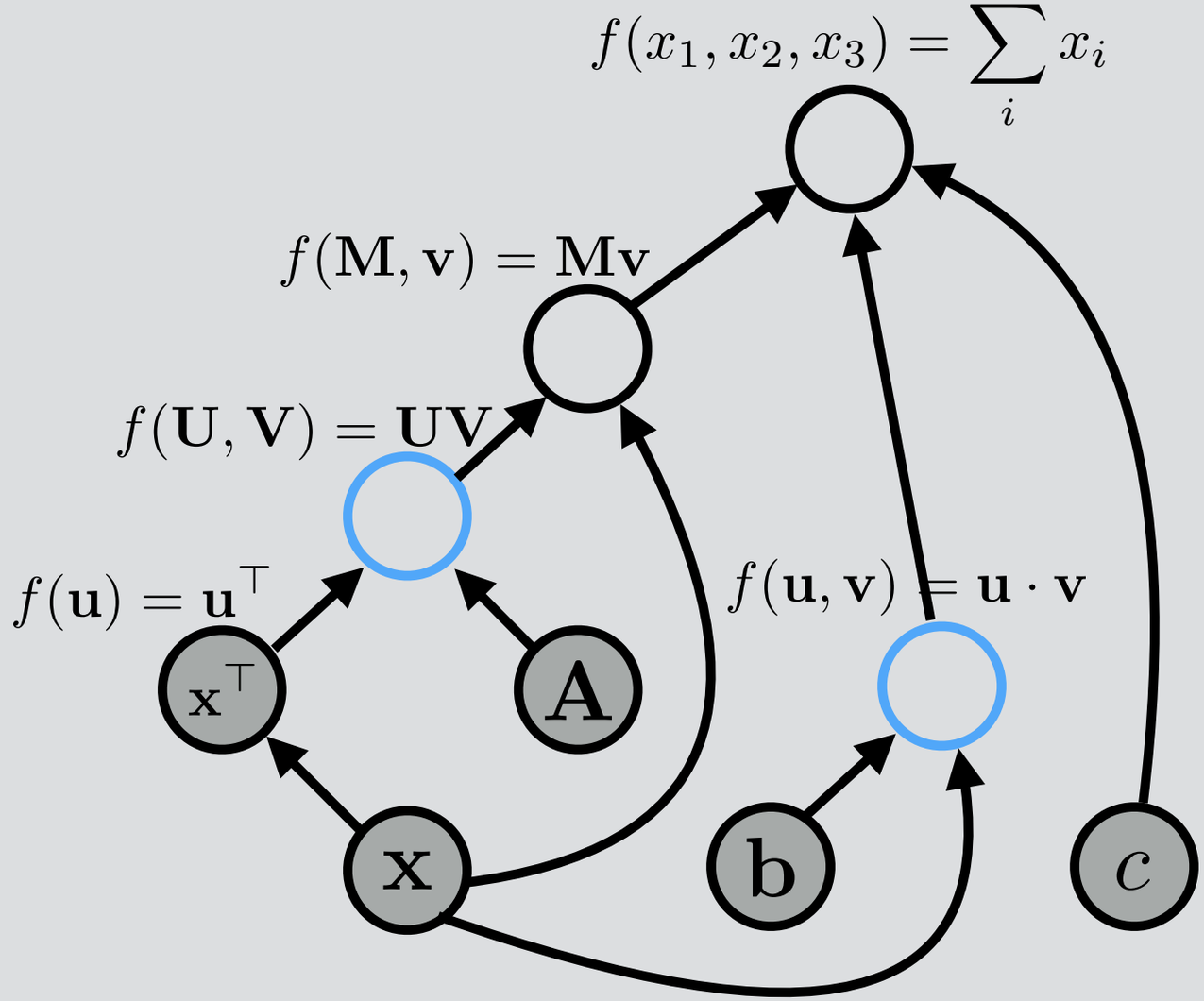
Forward Propagation

graph:



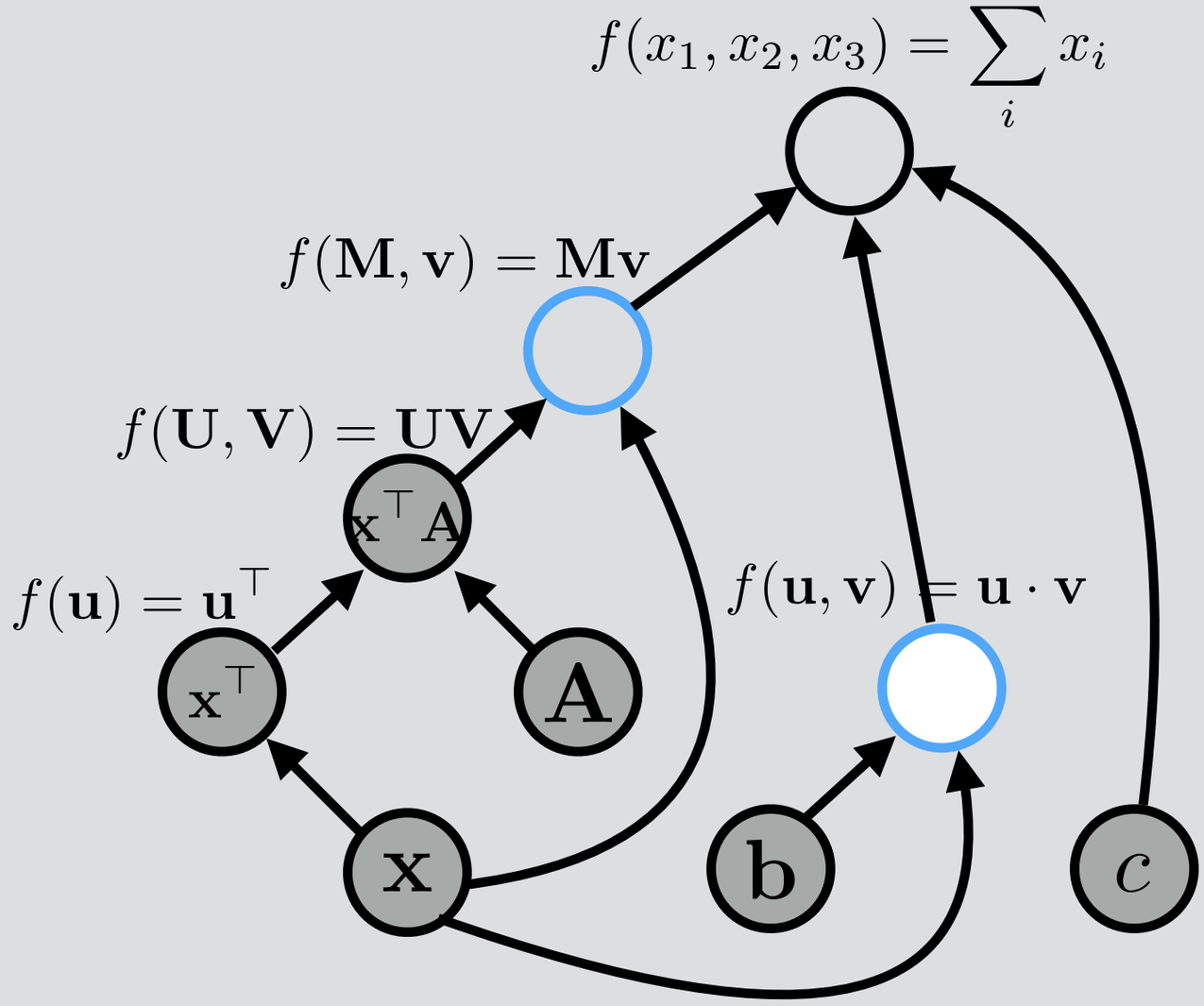
Forward Propagation

graph:



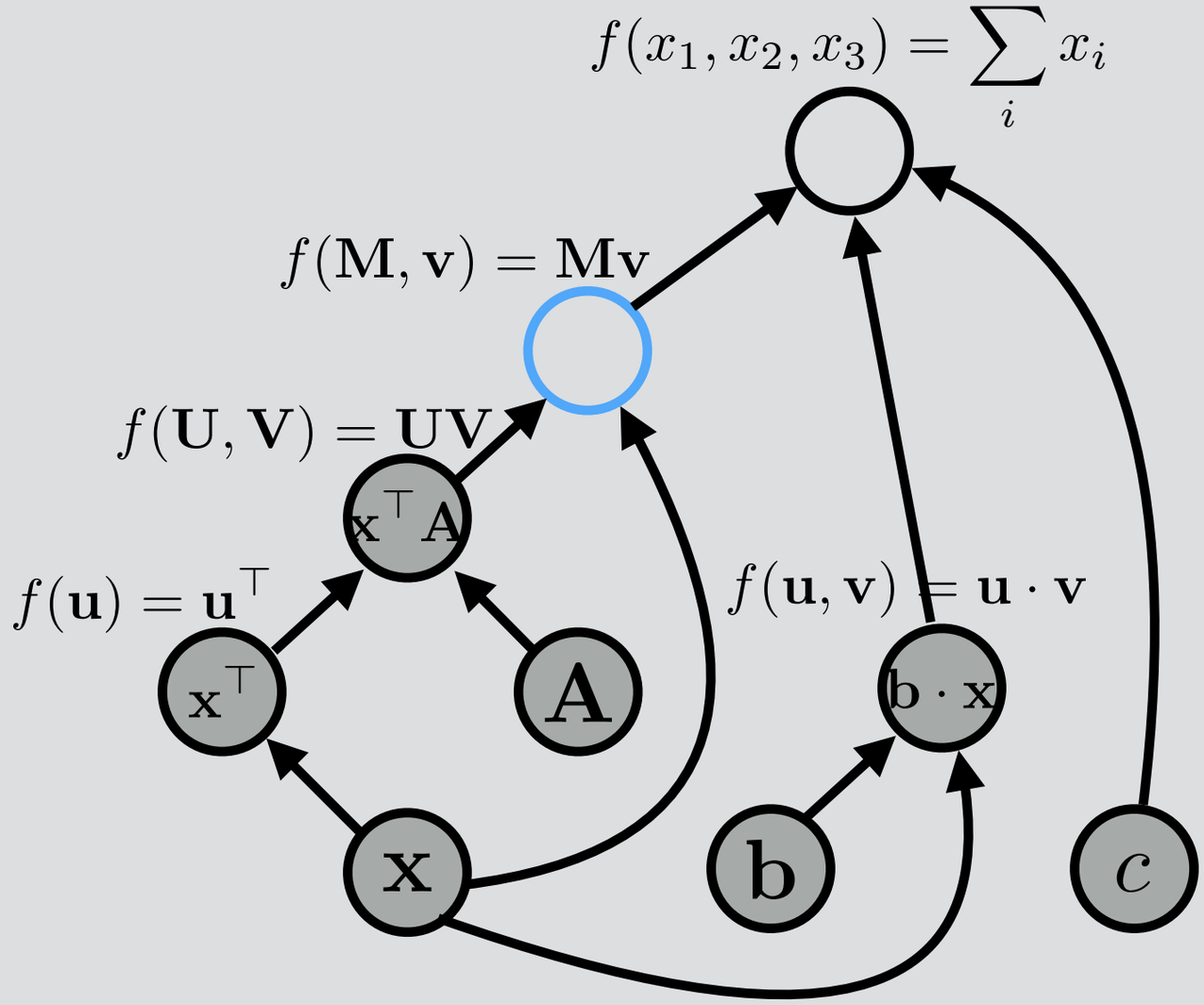
Forward Propagation

graph:



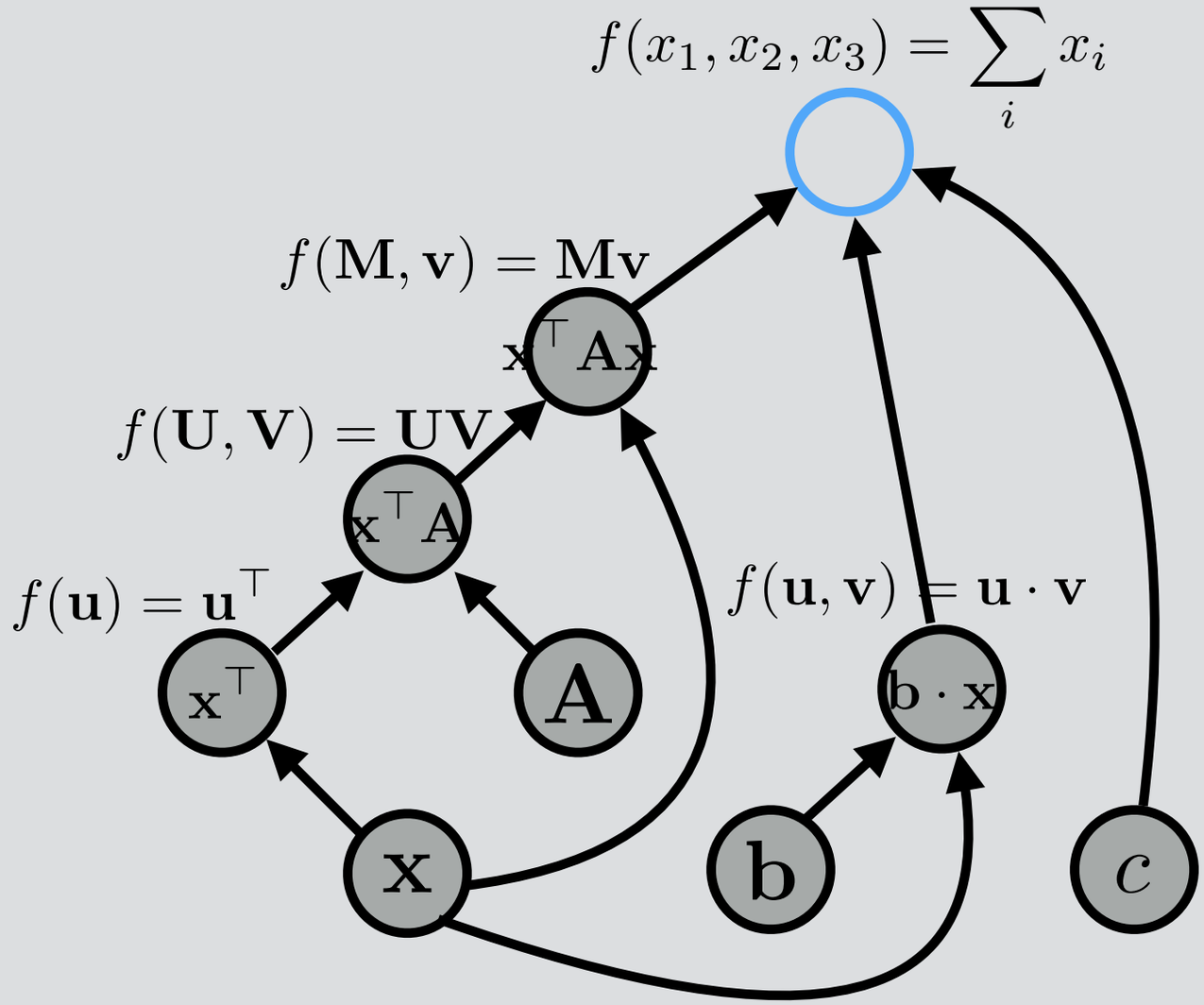
Forward Propagation

graph:



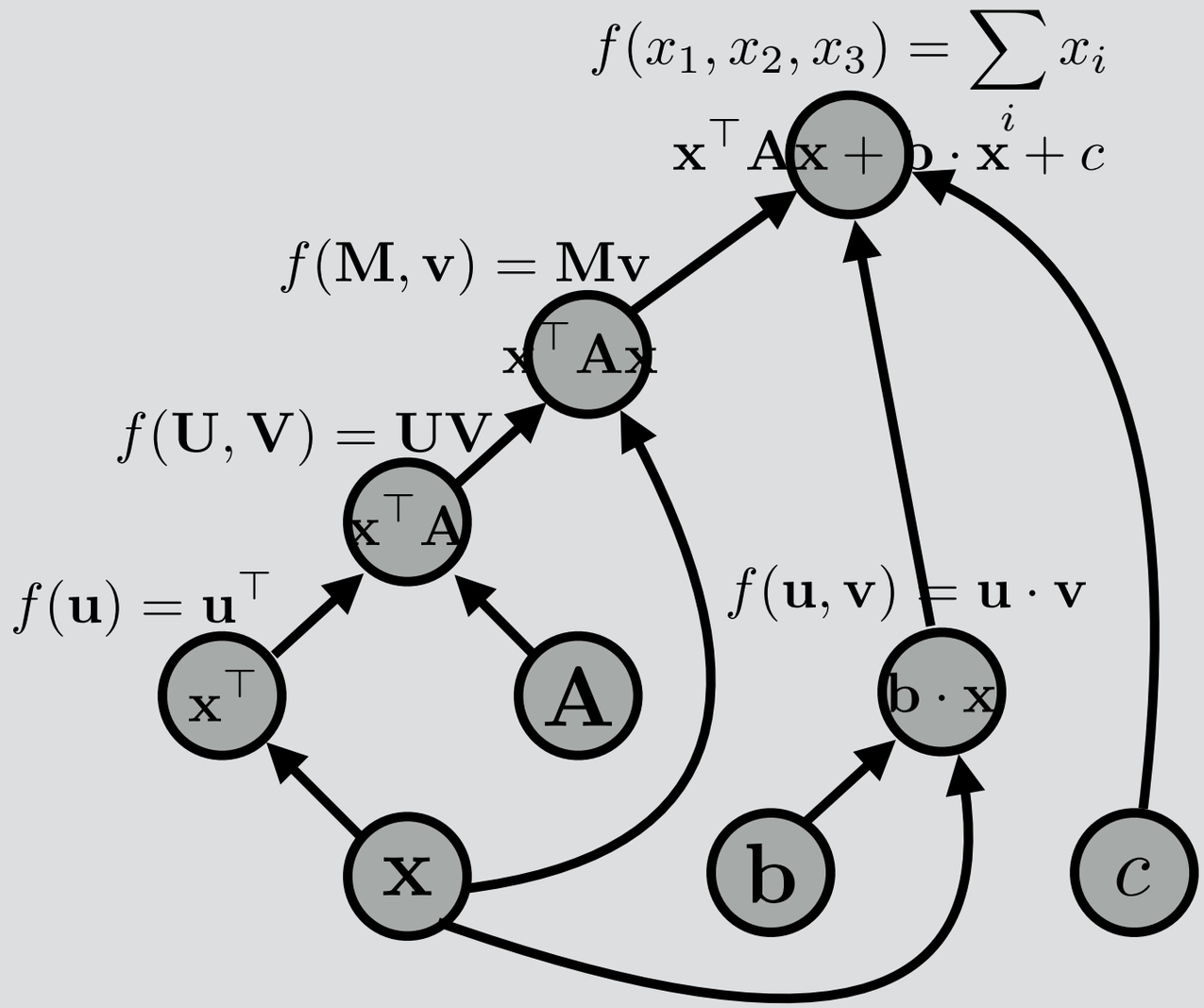
Forward Propagation

graph:



Forward Propagation

graph:



Algorithms (2)

- **Back-propagation:**
  - Process examples in reverse topological order
  - Calculate the derivatives of the parameters with respect to the final value
    (This is usually a “loss function”, a value we want to minimize)
- **Parameter update:**
  - Move the parameters in the direction of this derivative
    $W -= \alpha * dl/dW$

Concrete Implementation Examples

Neural Network Frameworks

Static Frameworks

theano

Caffe

mxnet

TensorFlow

+Gluon

+Eager

Dynamic Frameworks (Recommended!)

∂y/net

Chainer

PYTORCH

Basic Process in Dynamic Neural Network Frameworks

- Create a model
- For each example
  - **create a graph** that represents the computation you want
  - **calculate the result** of that computation
  - if training, perform **back propagation and update**

Things to Remember Going Forward

Things to Remember

- Neural nets are powerful!
  - They are universal function approximators, can calculate any continuous function
- But language is hard, and data is limited.
  - We need to design our networks to have inductive bias, to make it easy to learn things we'd like to learn.

Outline for Today

- Class logistics

- Why NLP, and What is it?

- Scope of this course

- Text Processing Basics

- Neural Network Basics

HW 1

- HW 1 is out today. Due: Sept 9th.
- Implementing a feedforward neural net for sentiment classification
- PyTorch: https://pytorch.org/tutorials/beginner/deep_learning_60min_blitz.html