# Artificial Intelligence Foundations and Applications

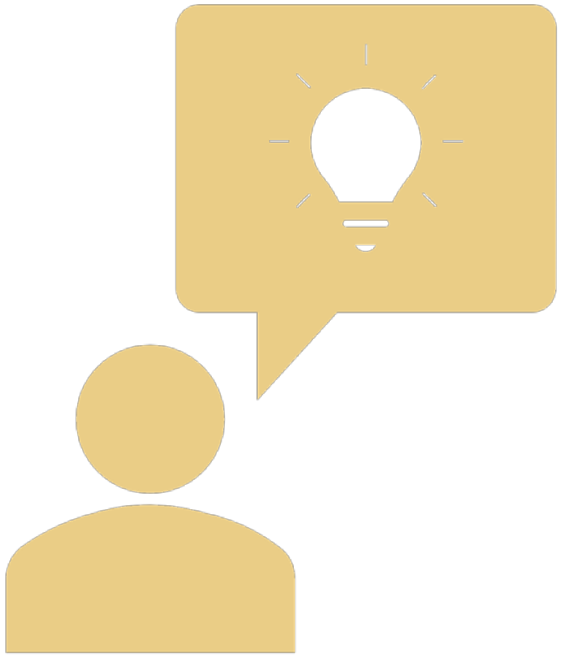
## Introduction to Natural Language Processing
## 13 Nov 2022

Centre of Excellence in Artificial Intelligence
IIT Kharagpur

# Language is the Tool for Communication

Language is the Vehicle for

- Learning knowledge
- Transmitting information
- Expressing thoughts, perceptions, feelings, information
- Making sense of complex and abstract thought

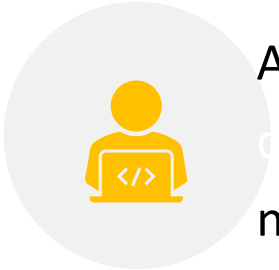


Communication is two-way

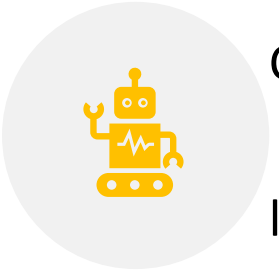
- Convey own ideas
- Receive thought of others

# Natural Language Processing

Building computational systems for analyzing and understanding human language input and/or producing natural language output.



Allow computers to communicate with people using natural language.



Computational methods for understanding of human language.

- Automating Language
  - Analysis Language → Representation
  - Generation Representation → Language
  - Acquisition Obtaining the representation and necessary algorithms, from knowledge and data

# Important Skills

Interact with our world using natural language

- E.g., Conversational agents

- Have computers read all the text out there
  - Retrieve
  - Answer questions
  - Summarize
  - Find new insights, Intelligence

# Some Applications

- Search
- Language Translation
- Chatbots
- Question Answering
- Text Summarization
- Sentiment Analysis
- Topic Extraction
- Named Entity Recognition
- Relation Extraction
- Social Media Monitoring

# Some application domains

Healthcare

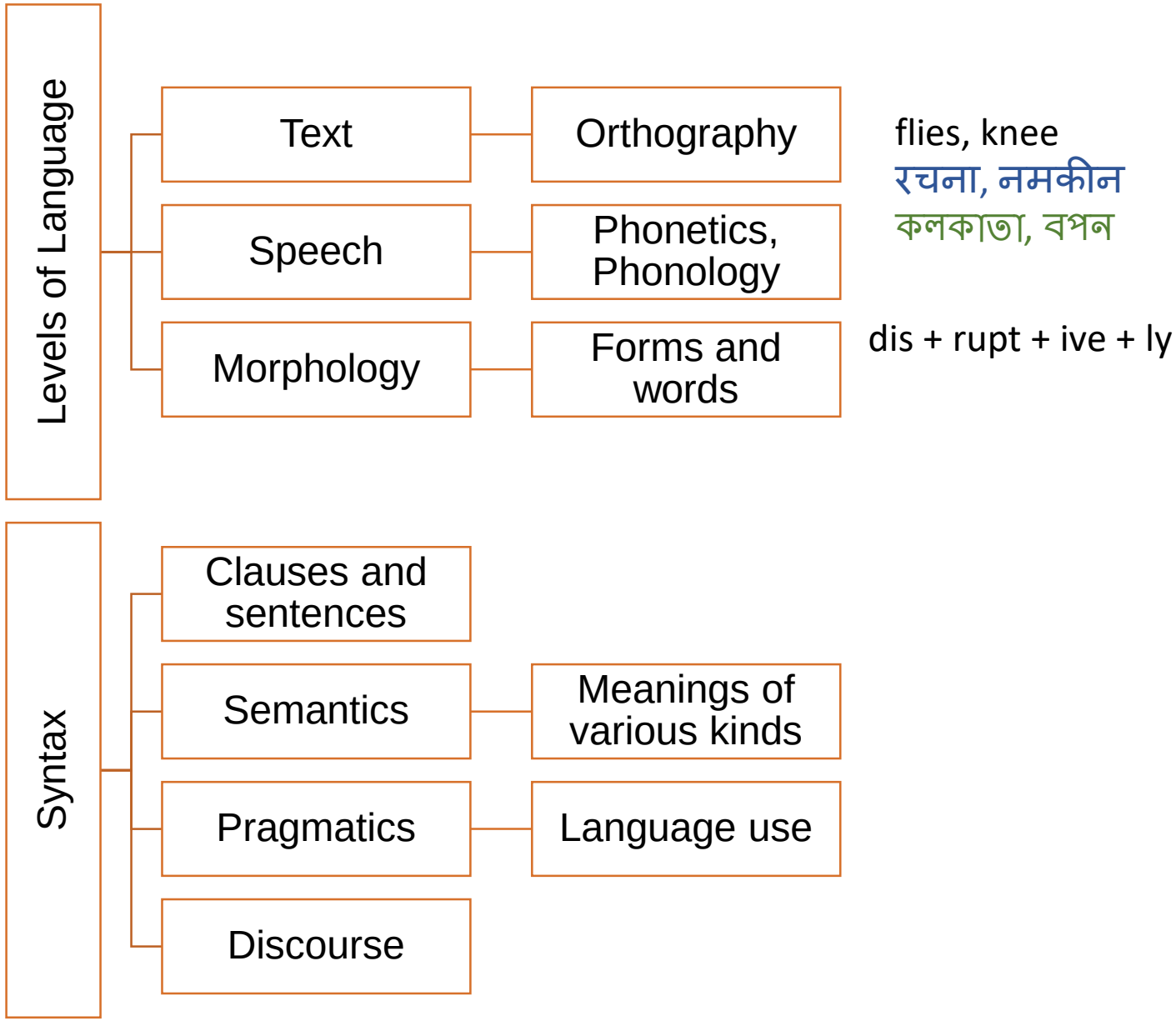
Law Enforcement

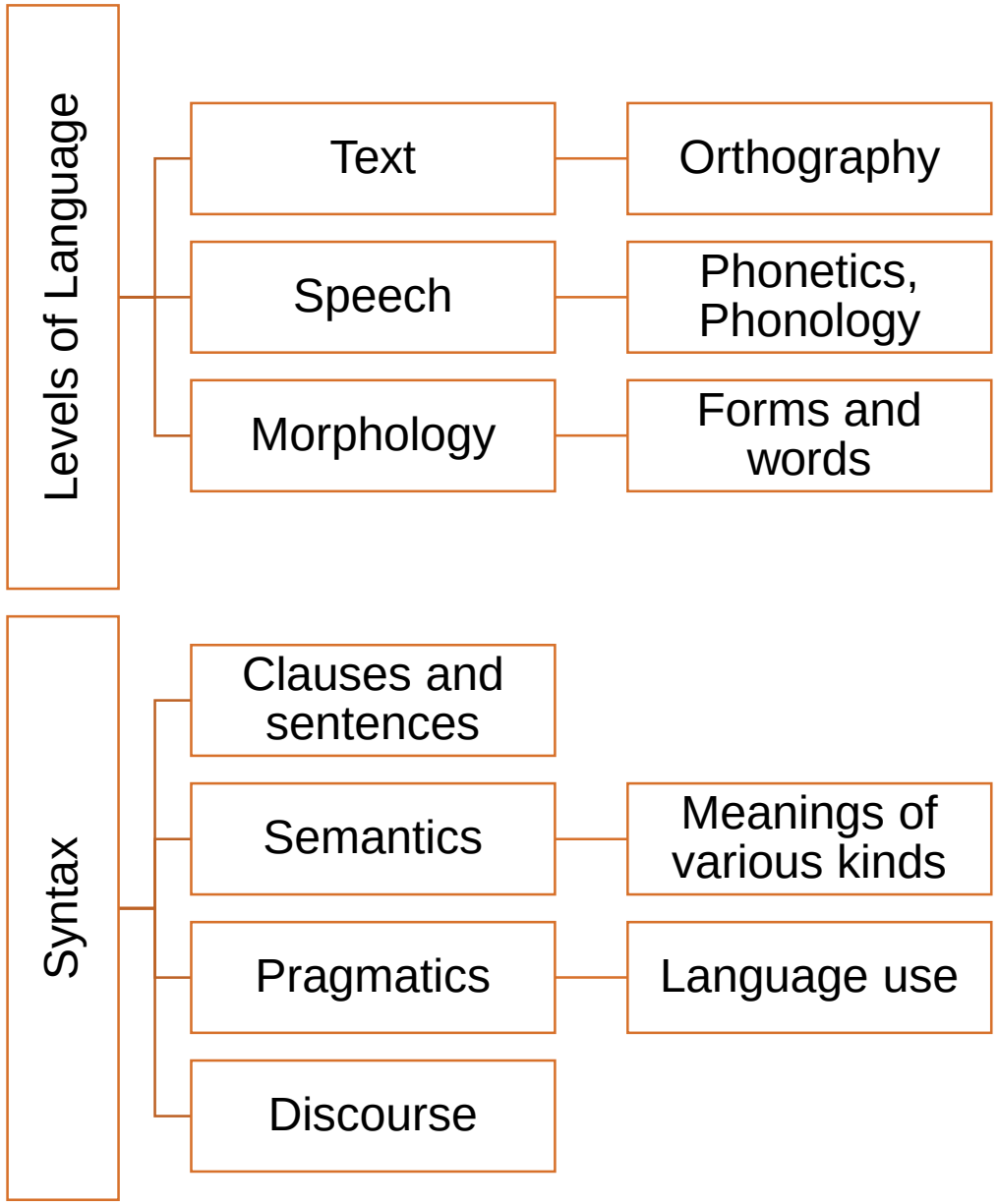
Education

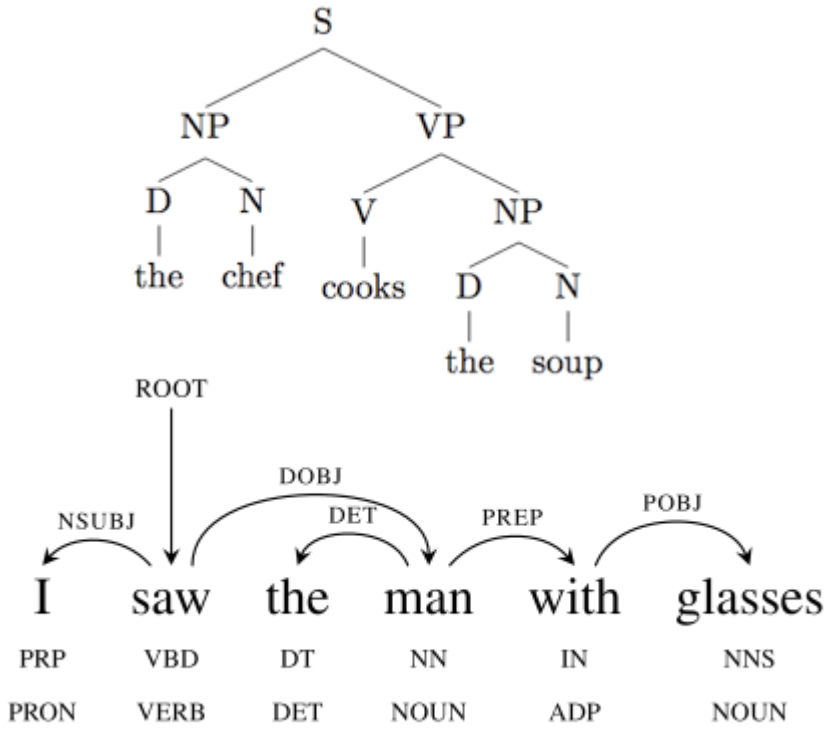
Analyze public feedback

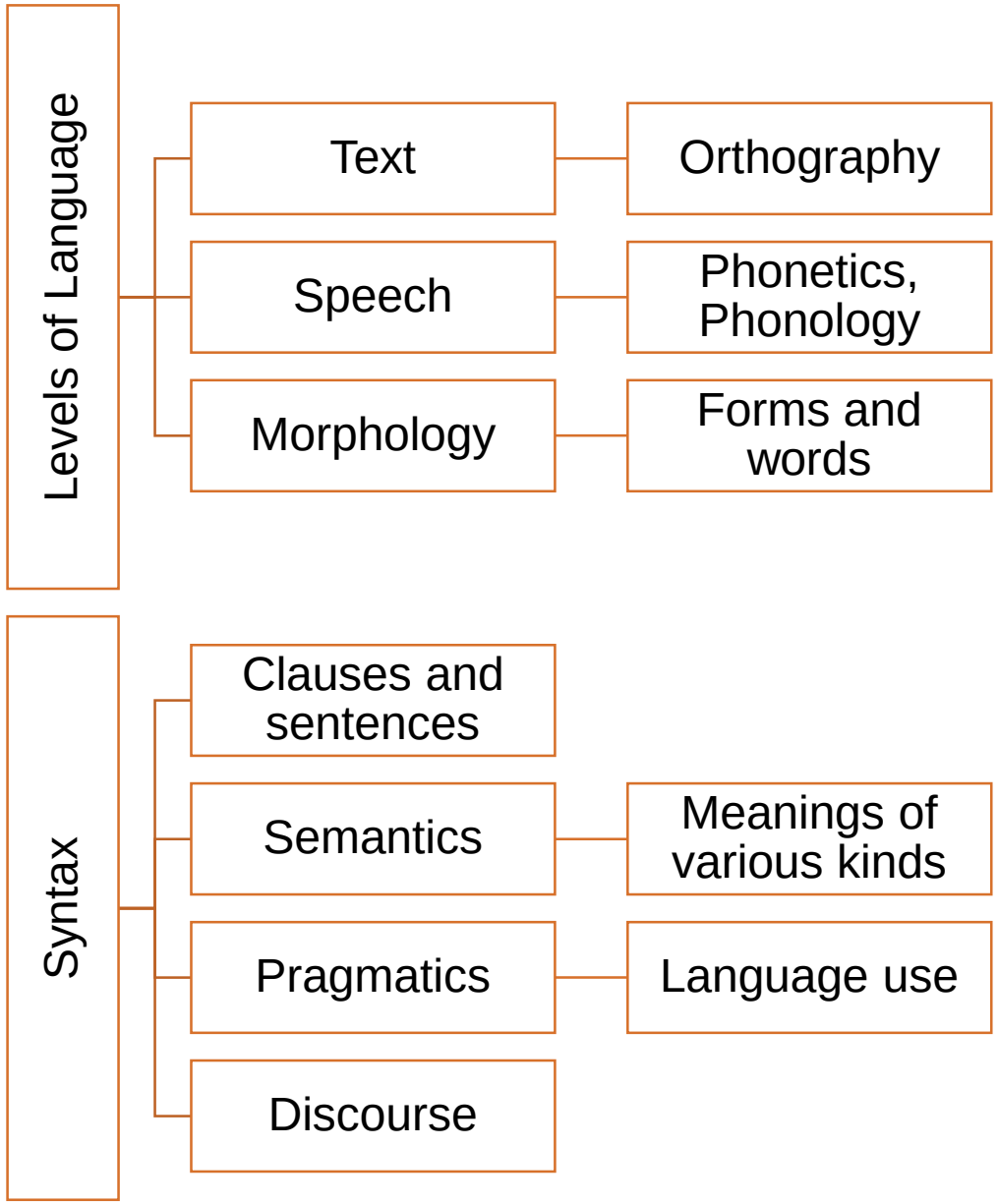
Agriculture

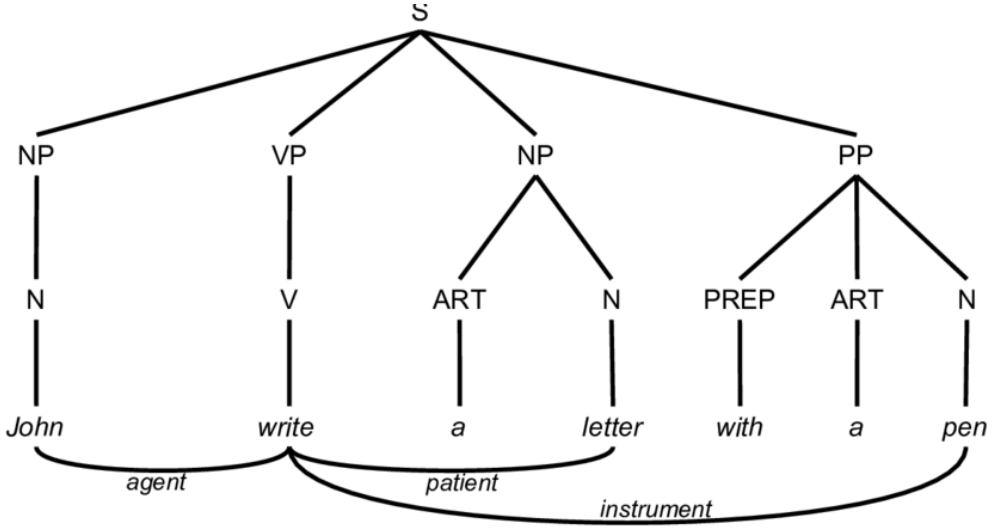
Improve Regulatory Compliance


Levels of Language
Text
Orthography
flies, knee
रचना, नमकीन
কলকাতা, বপন
Speech
Phonetics, Phonology
Morphology
Forms and words
dis + rupt + ive + ly
Syntax
Clauses and sentences
Semantics
Meanings of various kinds
Pragmatics
Language use
Discourse

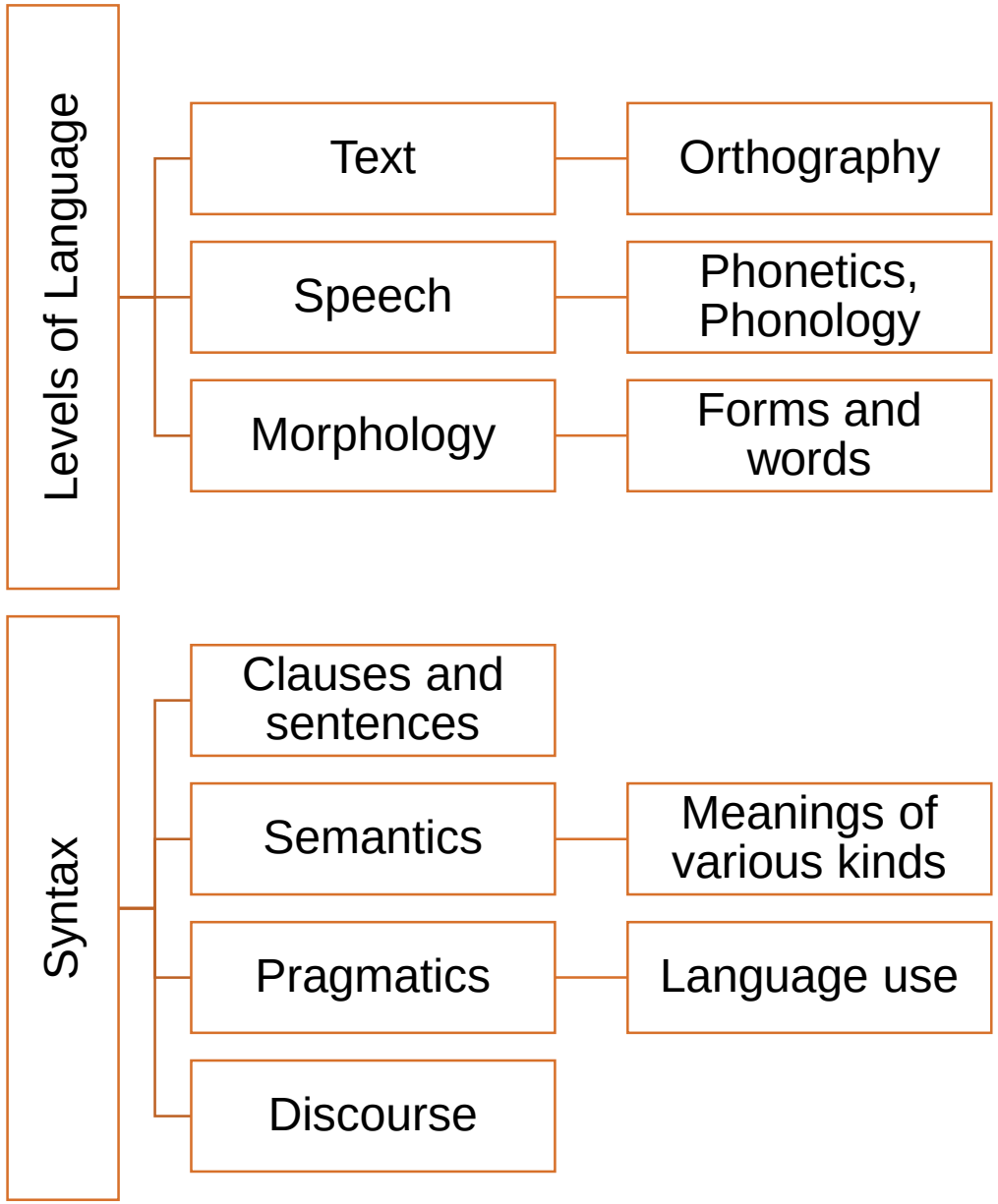

Levels of Language
Text
Orthography
Speech
Phonetics, Phonology
Morphology
Forms and words
Syntax
Clauses and sentences
Semantics
Meanings of various kinds
Pragmatics
Language use
Discourse


S
NP
VP
D
N
V
NP
the
chef
cooks
D
N
the
soup
ROOT
NSUBJ
DOBJ
DET
PREP
POBJ
I
saw
the
man
with
glasses
PRP
VBD
DT
NN
IN
NNS
PRON
VERB
DET
NOUN
ADP
NOUN


Levels of Language
Text
Orthography
Speech
Phonetics, Phonology
Morphology
Forms and words
Syntax
Clauses and sentences
Semantics
Meanings of various kinds
Pragmatics
Language use
Discourse


S
NP
VP
NP
PP
N
V
ART
N
PREP
ART
N
John
write
a
letter
with
a
pen
agent
patient
instrument

- Levels of Language
  - Text
    - Orthography
  - Speech
    - Phonetics, Phonology
  - Morphology
    - Forms and words
- Syntax
  - Clauses and sentences
  - Semantics
    - Meanings of various kinds
  - Pragmatics
    - Language use
  - Discourse

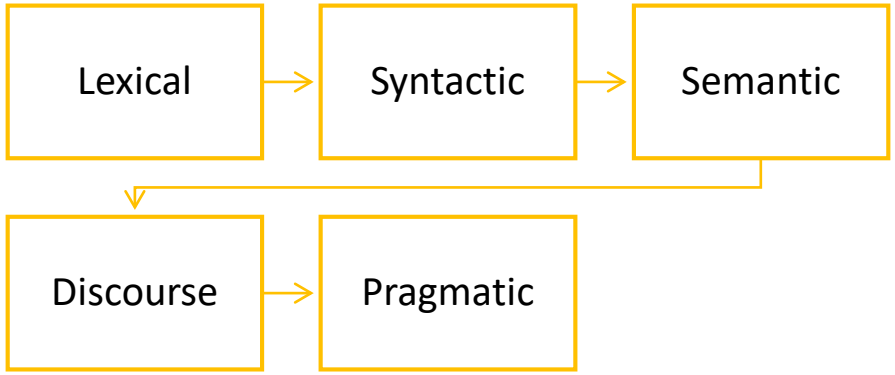



**This lesson** explains **syntactic analysis**.

**It** discusses the algorithms for **this task**.

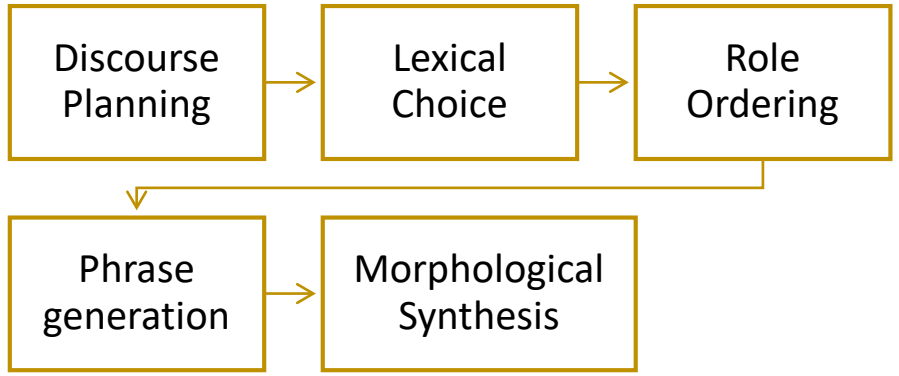
Analysis

Language → Representation




Generation

Representation → Language




Acquisition

Obtaining the representation from knowledge and data

# Morphology

- The identification, analysis and description of the structure of words
- Challenges
  - Ambiguity (flies, bears)
  - Segmenting text into words (Thai)
  - Sandhi splitting (Sanskrit)
  - Morphological variations
  - Words with multiple meanings (based on context, domain)
  - Multiword expression

# Part of Speech

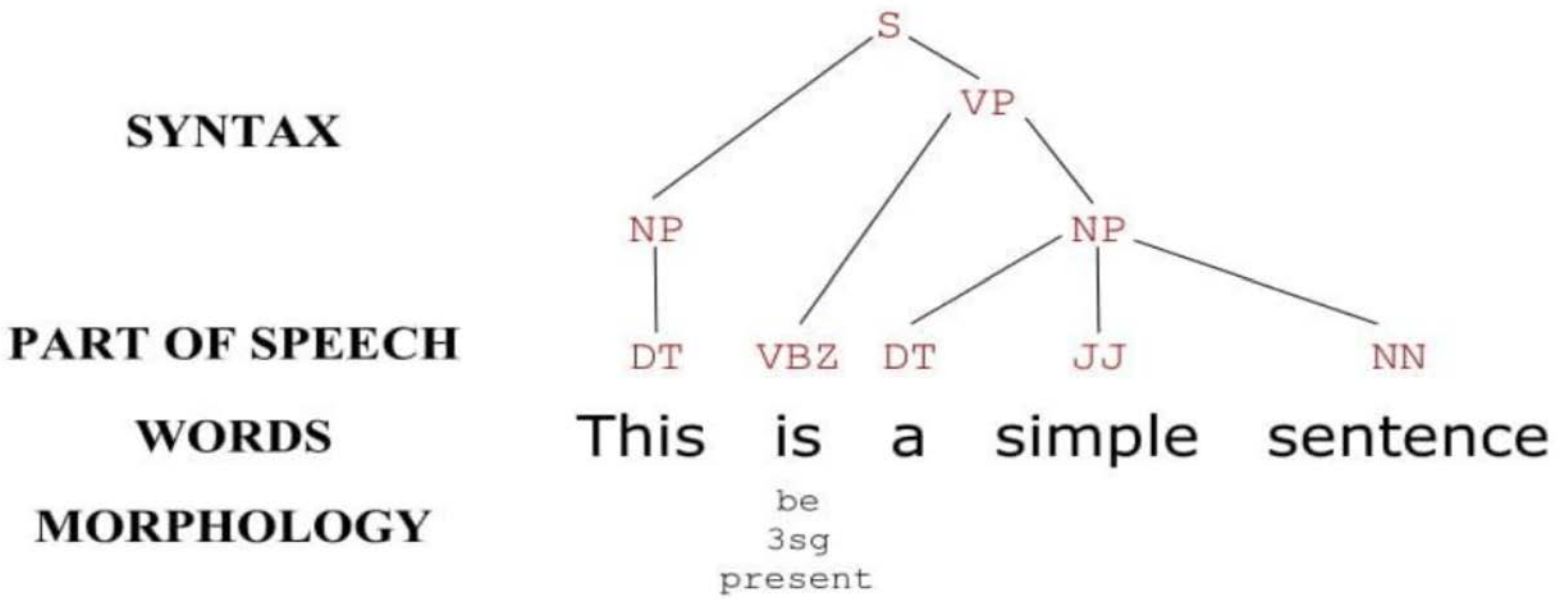
Part of speech tagging

# Syntax

- Syntax concerns the way in which words can be combined together to form (grammatical) sentences
  1. revolutionary new ideas appear infrequently
  2. colourless green ideas sleep furiously
  3. *ideas green furiously colourless sleep
- Words combine syntactically in certain orders in a way which mirrors the meaning conveyed
- John gave her dog biscuits
  - (john (gave (her) (dog biscuits)))
  - (john (gave (her dog) (biscuits)))

# Syntax

- Syntactic parsing



# Semantics

- The manner in which lexical meaning is combined morphologically and syntactically to form the meaning of a sentence
  - Concerns the meaning of words, phrases and sentences
  - The meaning of a sentence is usually a productive combination of the meaning of its words

  - Named entity recognition
  - Word sense disambiguation
  - Semantic role labeling

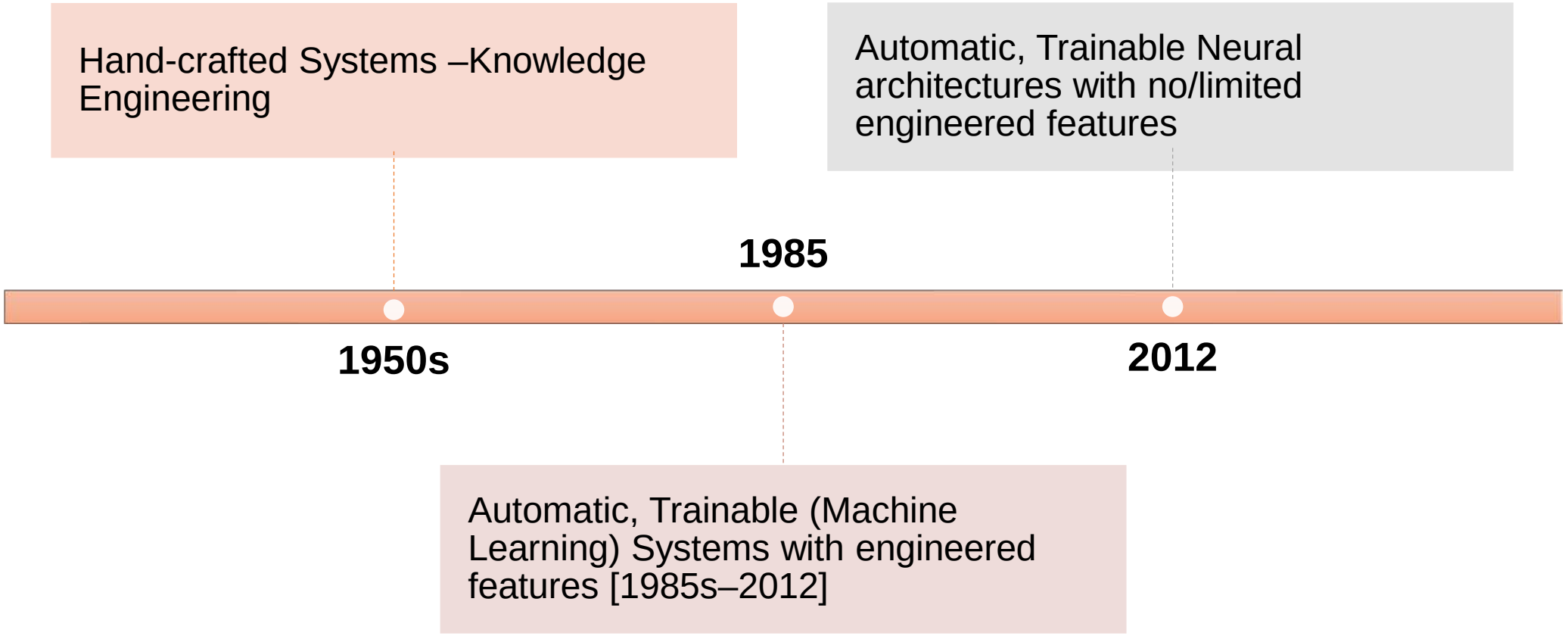
# Discourse analysis

- The meaning of a sentence depends upon the sentences that preceded it and also invokes the meaning of the sentences that follow it.
- The discourse structure of connected text, i.e. the nature of the discourse relationships between sentences (e.g. elaboration, explanation, contrast)

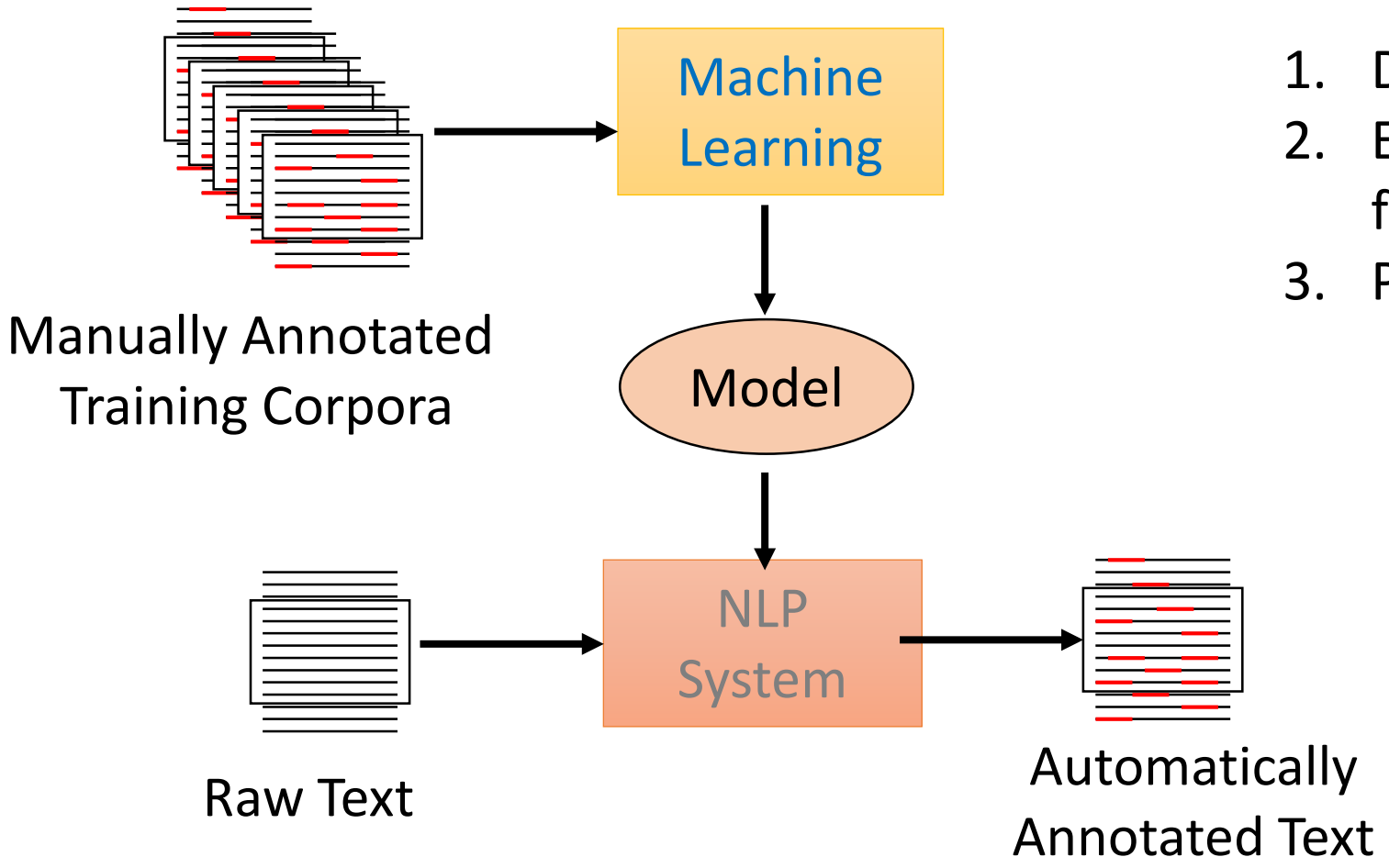
# Hardness of NLP

- Ambiguity
- Richness (Variability)
  - Any meaning may be expressed many ways, and there are immeasurably many meanings.
- Linguistic diversity across languages, dialects, genres, styles

# Three Generations of NLP



# Machine Learning Approach to NLP




1. Data
2. Extraction of “features” from text data
3. Prediction of output

# How do we represent words?

$$x_{1,1} = \begin{bmatrix} 0 \\ 0 \\ . \\ 0 \\ 1 \\ 0 \\ . \\ 0 \end{bmatrix}$$

dimensionality = number of possible words

index of this word

This means basically nothing by itself

# Vector Embedding Dimension

An embedding method should give different embeddings to different words, but a naïve (one-hot encoding) is very expensive:

- Dog [0, 0, 0, 0,…, 0, 0, 1, 0, 0, 0, 0,…,0]
- Man [0, 0, 0, 0,…, 0, 0, 0, 0, 1, 0, 0,…,0]

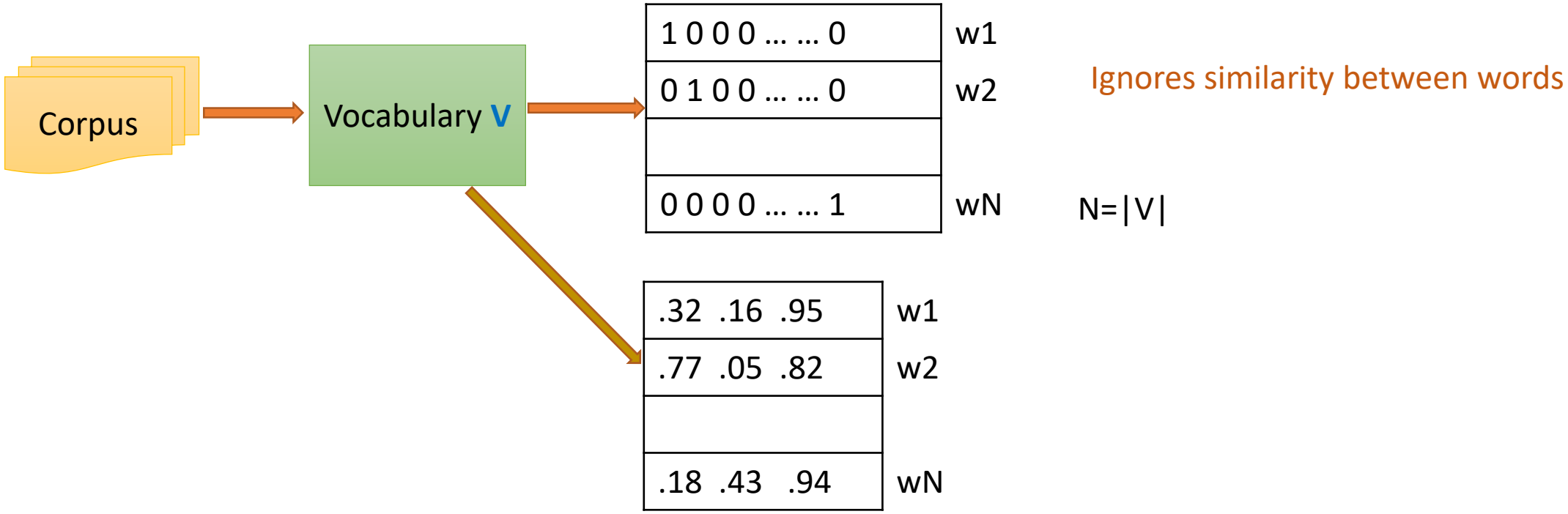
Vector size = vocabulary size

- But words are too different (inner products always zero) in this representation.

**Maybe** if we had a more meaningful representation of words, then learning downstream tasks would be much easier!

Meaningful = vectors corresponding to similar words should be close together

# Word Representations




Word2vec: Represent each word with a low-dimensional dense vector

Model more generalizable

# Word Representation

- Continuous Representation: based on context
- Distributional hypothesis

You can get a lot of value by representing a word by means of its neighbors

"You shall know a word by the company it keeps"

(J. R. Firth 1957: 11)

One of the most successful ideas of modern NLP

| | | |
|---|---|---|
| government debt problems turning into | banking | crises as has happened in |
| saying that Europe needs unified | banking | regulation to replace the hodgepodge |

↖ These words will represent *banking* ↗

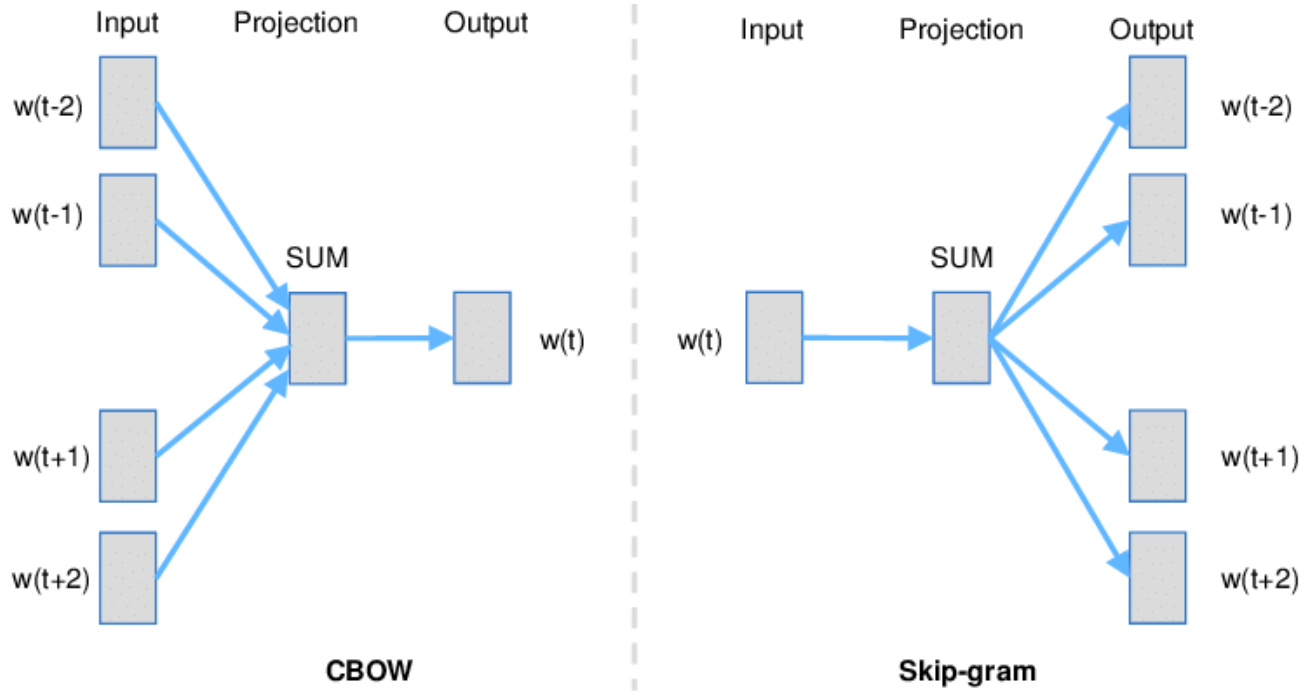
# Word2vec Representations

"You shall know a word by the company it keeps"

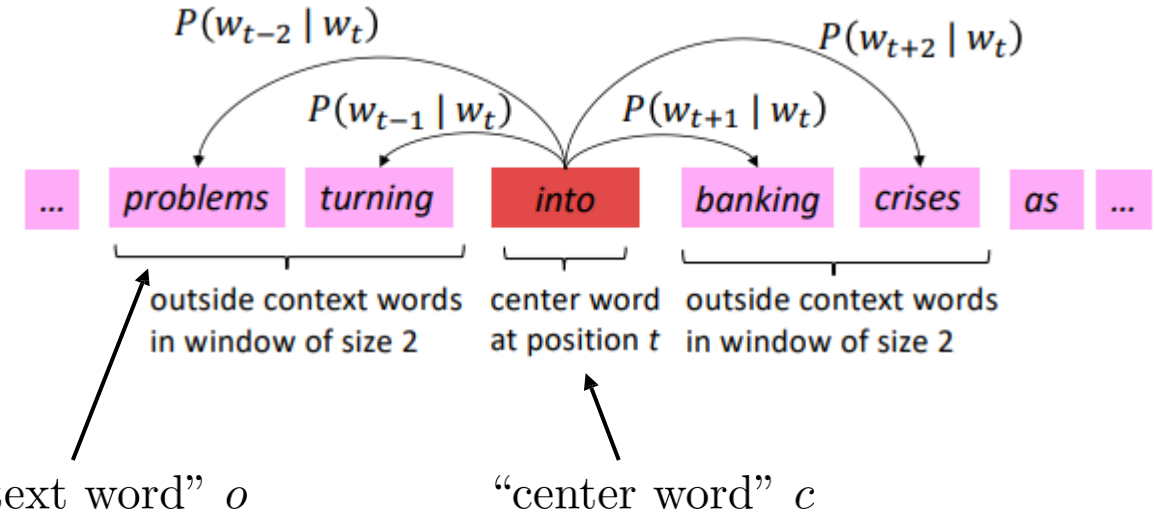
Key idea: Predict surrounding words of every word

Assign each word a vector such that similar words have similar vectors

1. CBOW: P(Word|Context)
2. Skipgram: P(Context|Word)



Can we **predict** the neighbors of a word from its **embedding**?




(learned) vector representation of $o$

(learned) vector representation of $c$

$$p(o|c) = \frac{\exp(u_o^T v_c)}{\sum_{w \in V} \exp(u_w^T v_c)}$$

all possible words (vocabulary)

looks a bit like a **logistic regression** model

how to train? $\arg \max_{u_1,\ldots,u_n,v_1,\ldots,v_n} \sum_{c,o} \log p(o|c)$

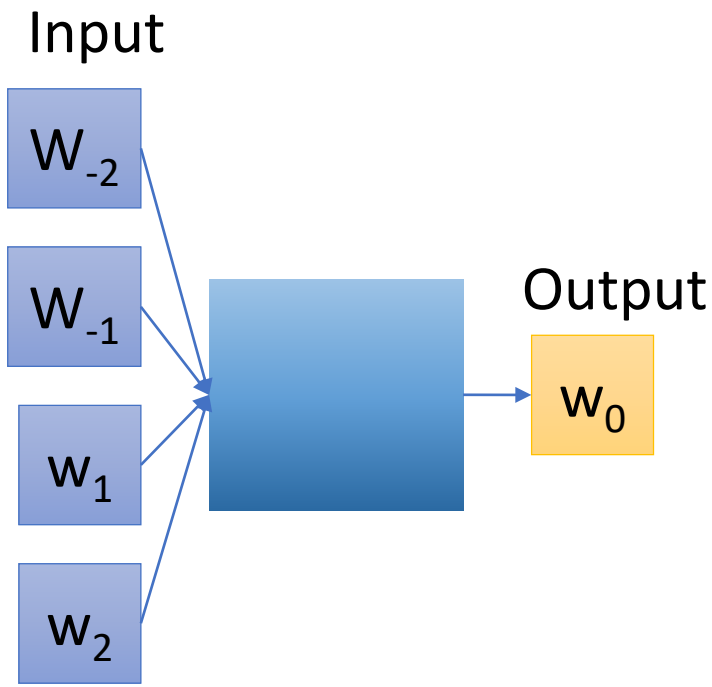
$u$ and $v$ vectors for all possible words

all possible $c$ and $o$ combinations
e.g., for each word $c$, pick all words $o$ that are within 5 step

Examples & Images: Christopher Manning, CS224n

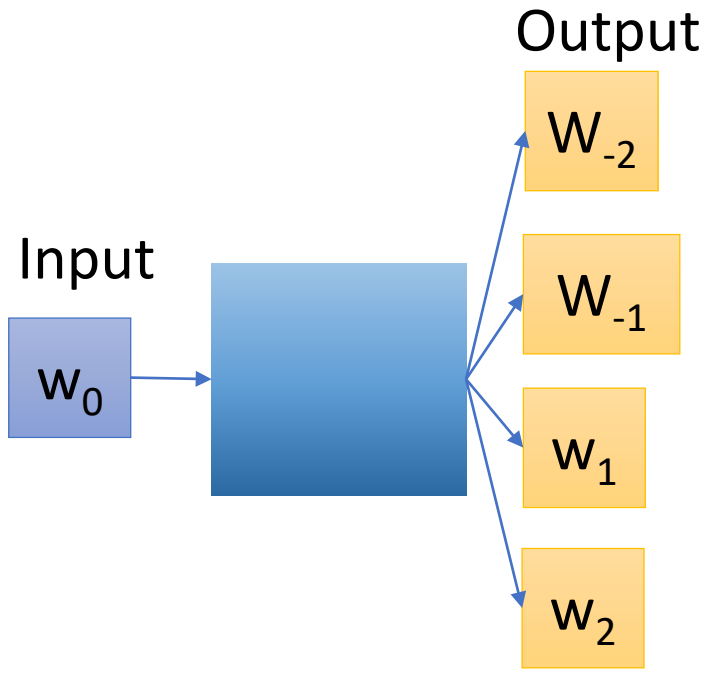
# Word2Vec: main context representation models

Continuous Bag of Words (CBOW)




P(word|context)

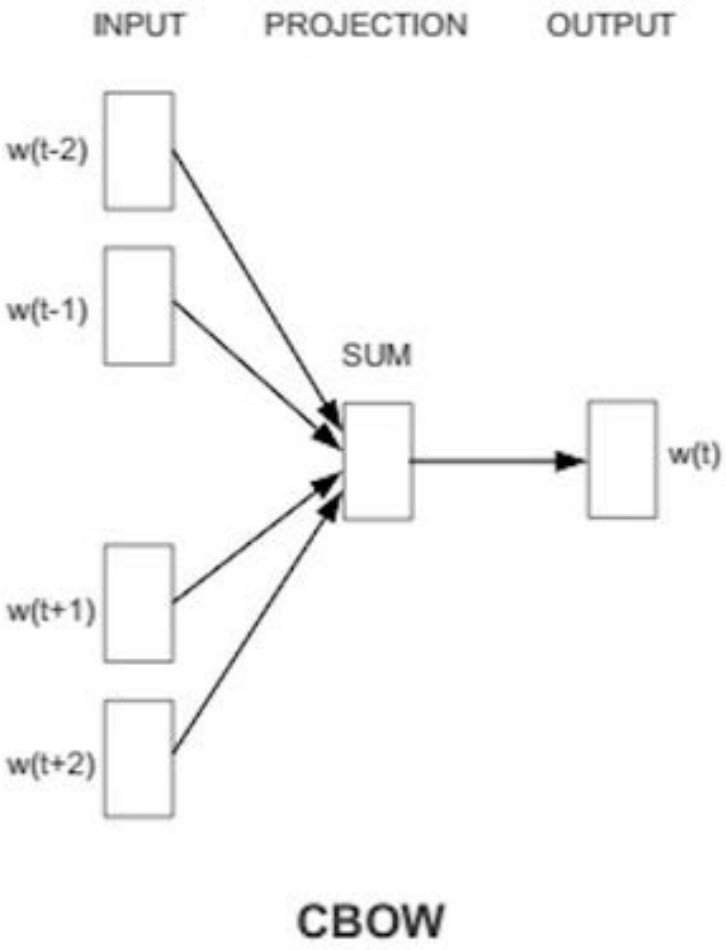
Skip-gram




P(context|word)

# CBOW

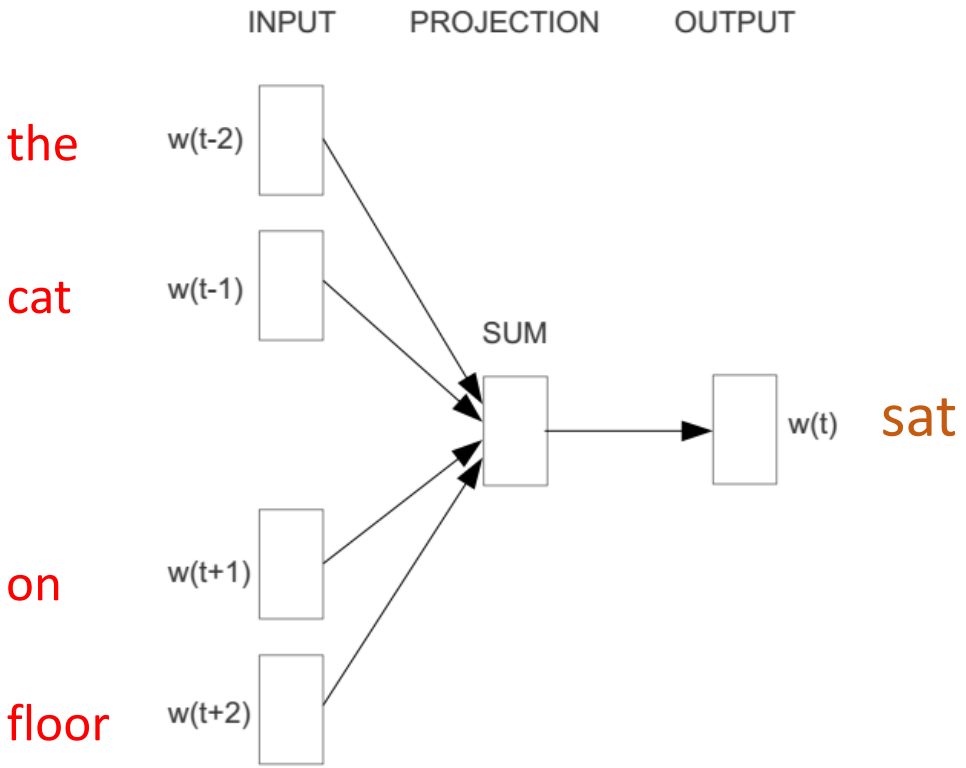
- Bag of words
- Gets rid of word order.
- Takes vector embeddings of n words before target and n words after and adds them (as vectors).

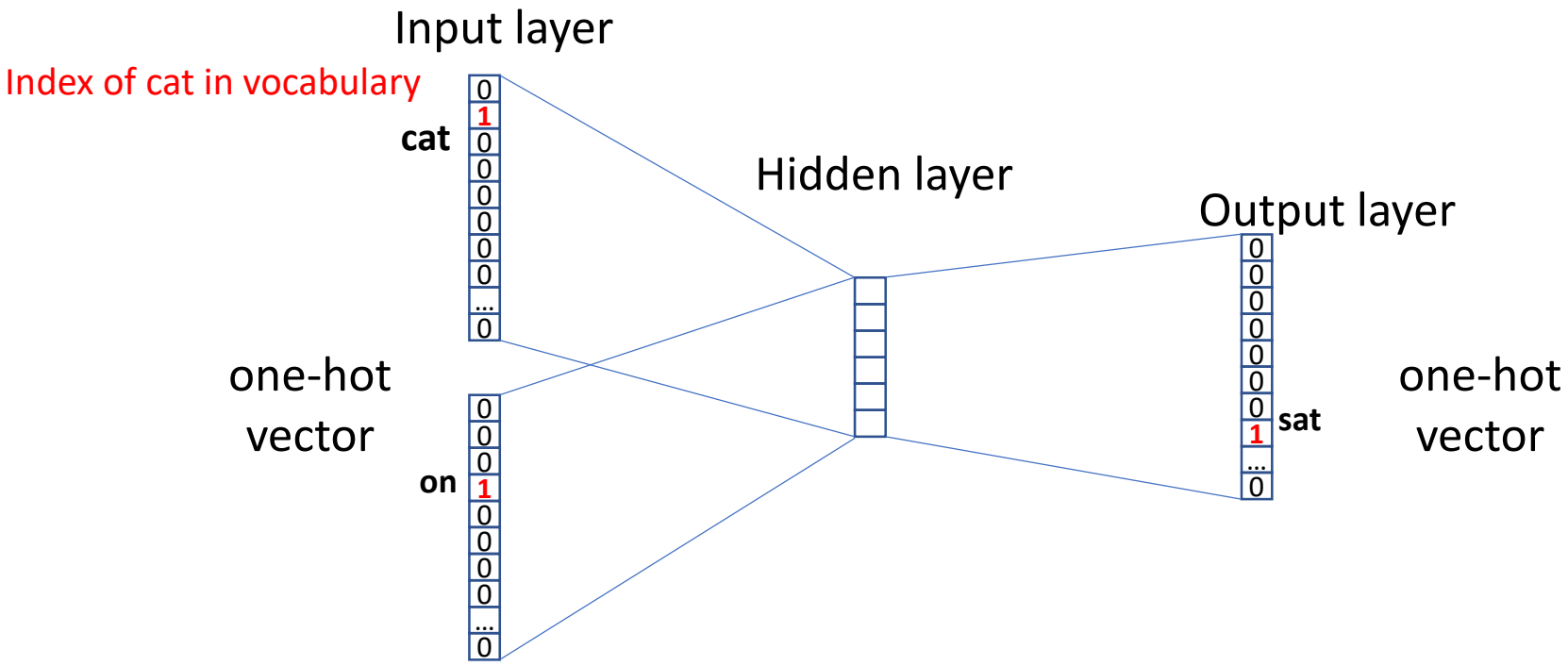


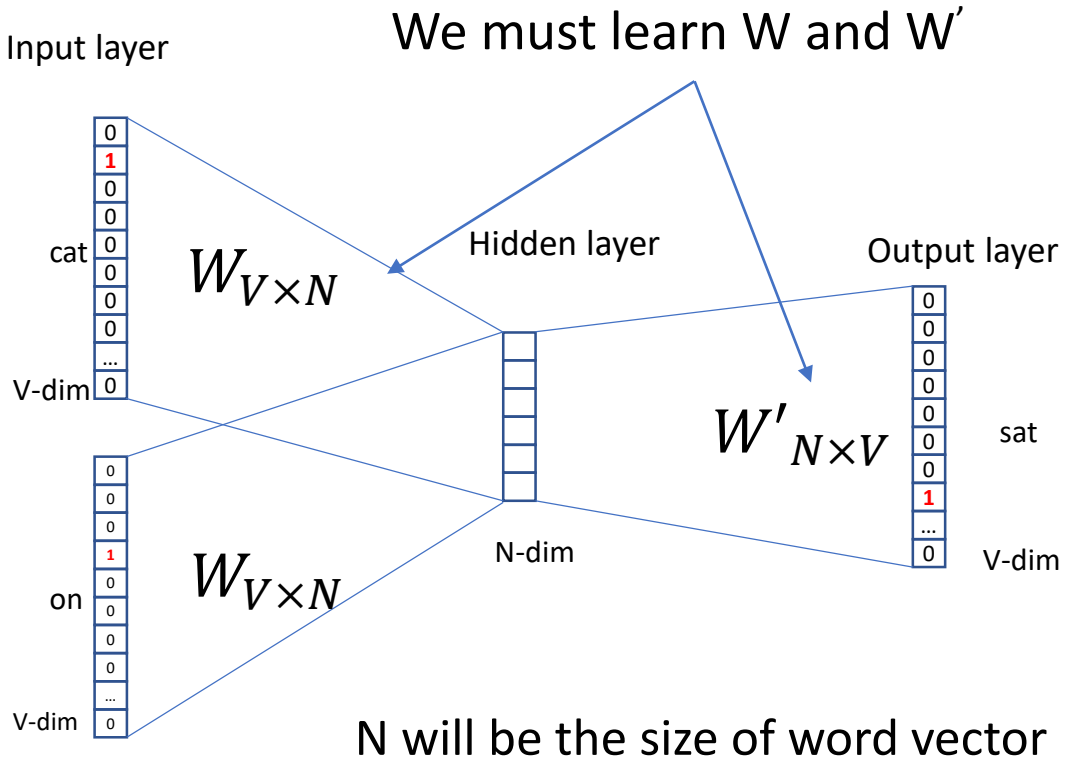
# Word2vec – Continuous Bag of Word

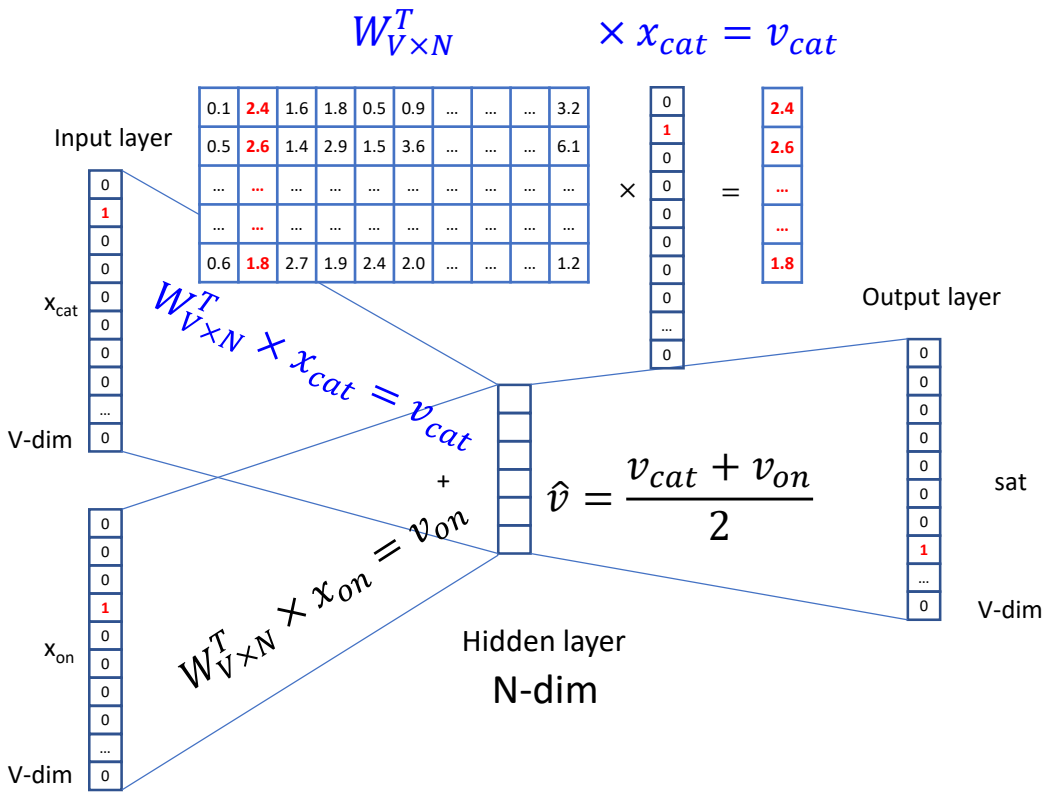
E.g. “The cat sat on floor”

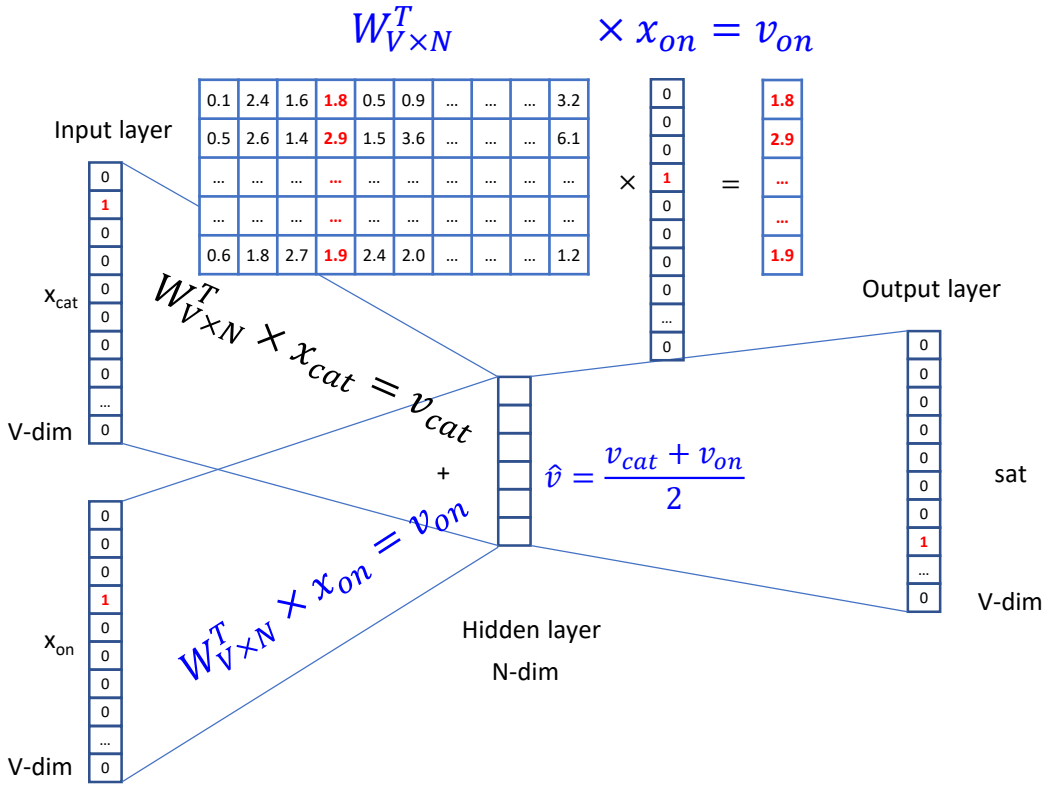
- Window size = 2

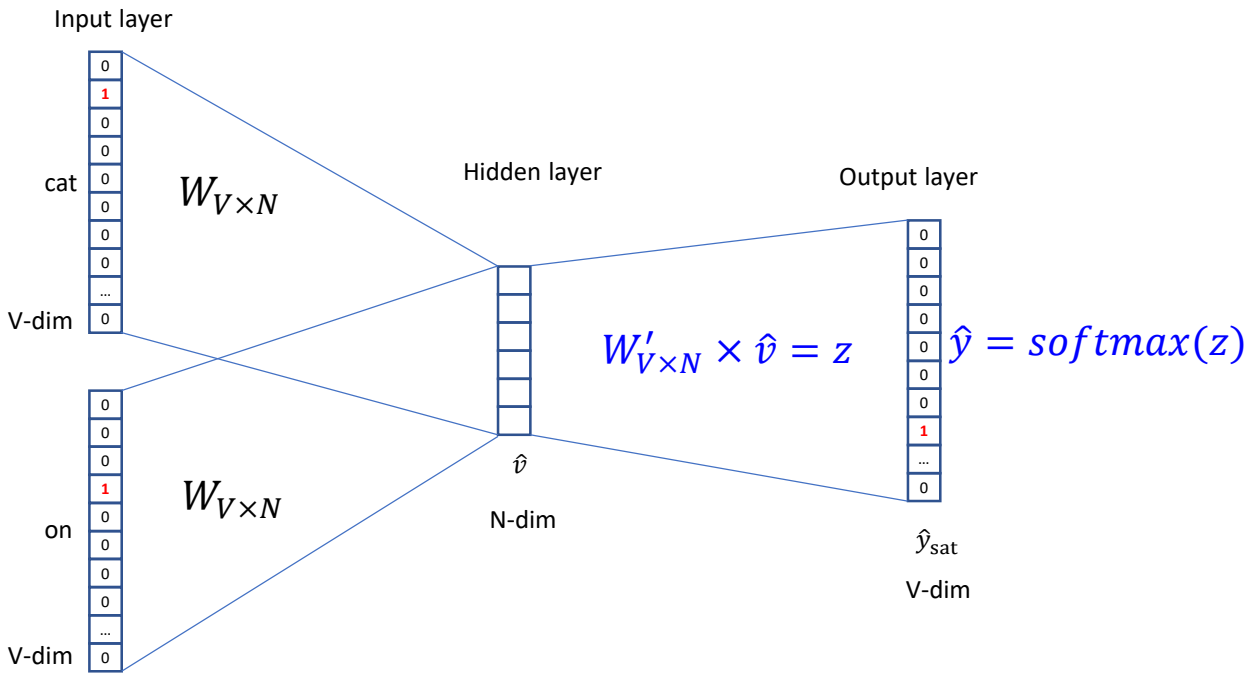



Input layer
Index of cat in vocabulary
cat
one-hot vector
on
Hidden layer
Output layer
sat
one-hot vector

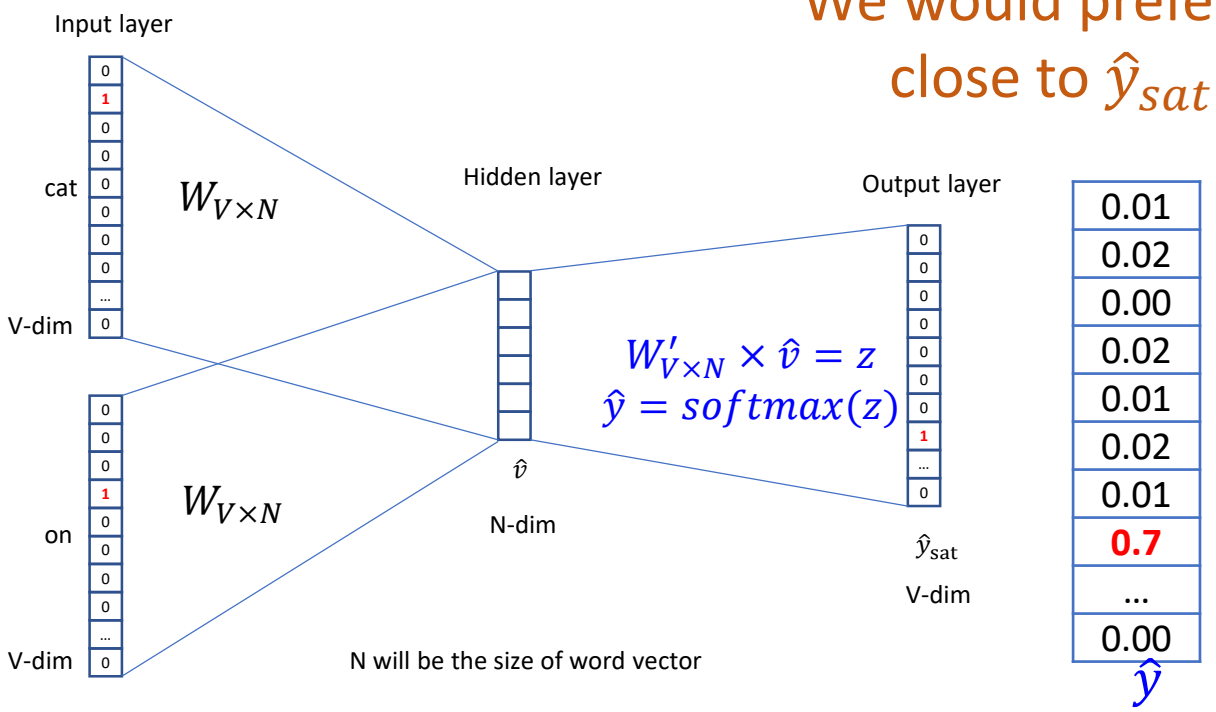

We must learn W and W'
Input layer
cat
V-dim
on
V-dim
$W_{V\times N}$
$W_{V\times N}$
Hidden layer
N-dim
$W'_{N\times V}$
Output layer
sat
V-dim
N will be the size of word vector

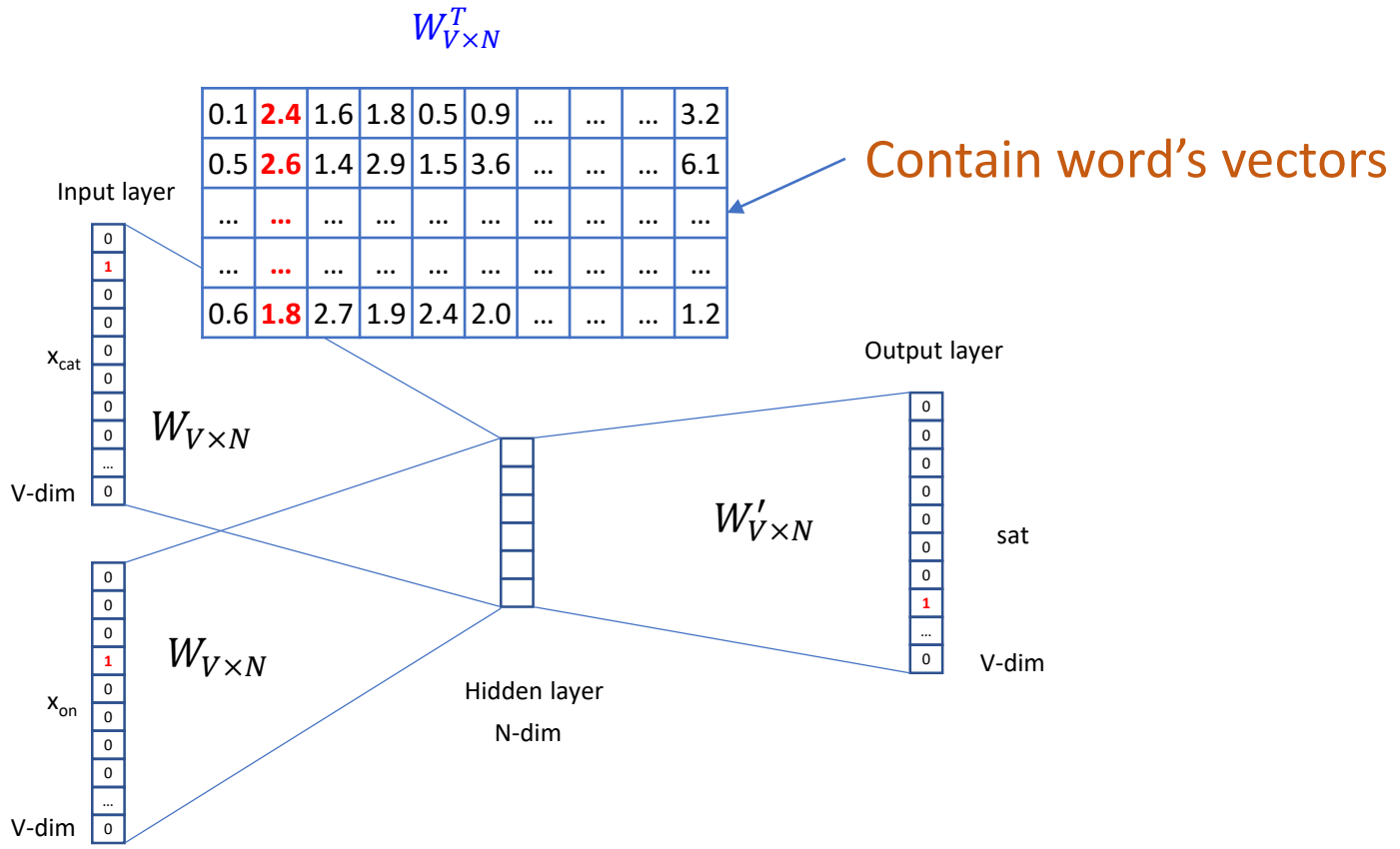

$W_{V\times N}^{T}$
$\times x_{cat} = v_{cat}$
Input layer
$x_{cat}$
V-dim
$W_{V\times N}^{T} \times x_{cat} = v_{cat}$
+
$W_{V\times N}^{T} \times x_{on} = v_{on}$
$x_{on}$
V-dim
$\hat{v} = \frac{v_{cat} + v_{on}}{2}$
Hidden layer
N-dim
Output layer
sat
V-dim


$W^T_{V\times N}$
$\times x_{on} = v_{on}$
Input layer
$x_{cat}$
V-dim
$x_{on}$
V-dim
$W^T_{V\times N} \times x_{cat} = v_{cat}$
+
$W^T_{V\times N} \times x_{on} = v_{on}$
$\hat{v} = \frac{v_{cat} + v_{on}}{2}$
Hidden layer
N-dim
Output layer
sat
V-dim


Input layer
cat
V-dim
on
V-dim
$W_{V\times N}$
$W_{V\times N}$
Hidden layer
$\hat{v}$
N-dim
$W'_{V\times N} \times \hat{v} = z$
Output layer
$\hat{y} = softmax(z)$
$\hat{y}_{\text{sat}}$
V-dim

We would prefer $\hat{y}$ close to $\hat{y}_{sat}$

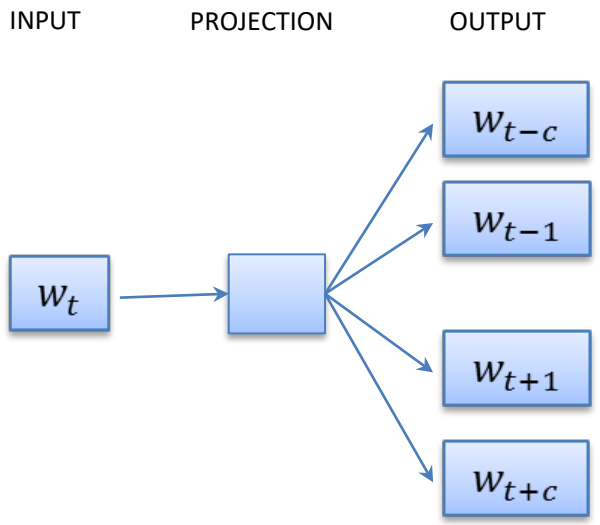





We can consider either W or W' as the word's representation. Or even take the average.

# Skipgram Model

- Input: Central word $w_t$
- Output: Words in its context: $\boldsymbol{w_{con}}$
  $\{w_{t-c}, \ldots, w_{t-1}, w_{t+1}, \ldots, w_{t+c}\}$
- Each input word represented by a 1-hot encoding of size V




Source Text:

**Deep Learning attempts to learn multiple levels of representation from data.**

Input output pairs :

**Positive samples**:

(representation, levels)

(representation, of)

(representation, from)

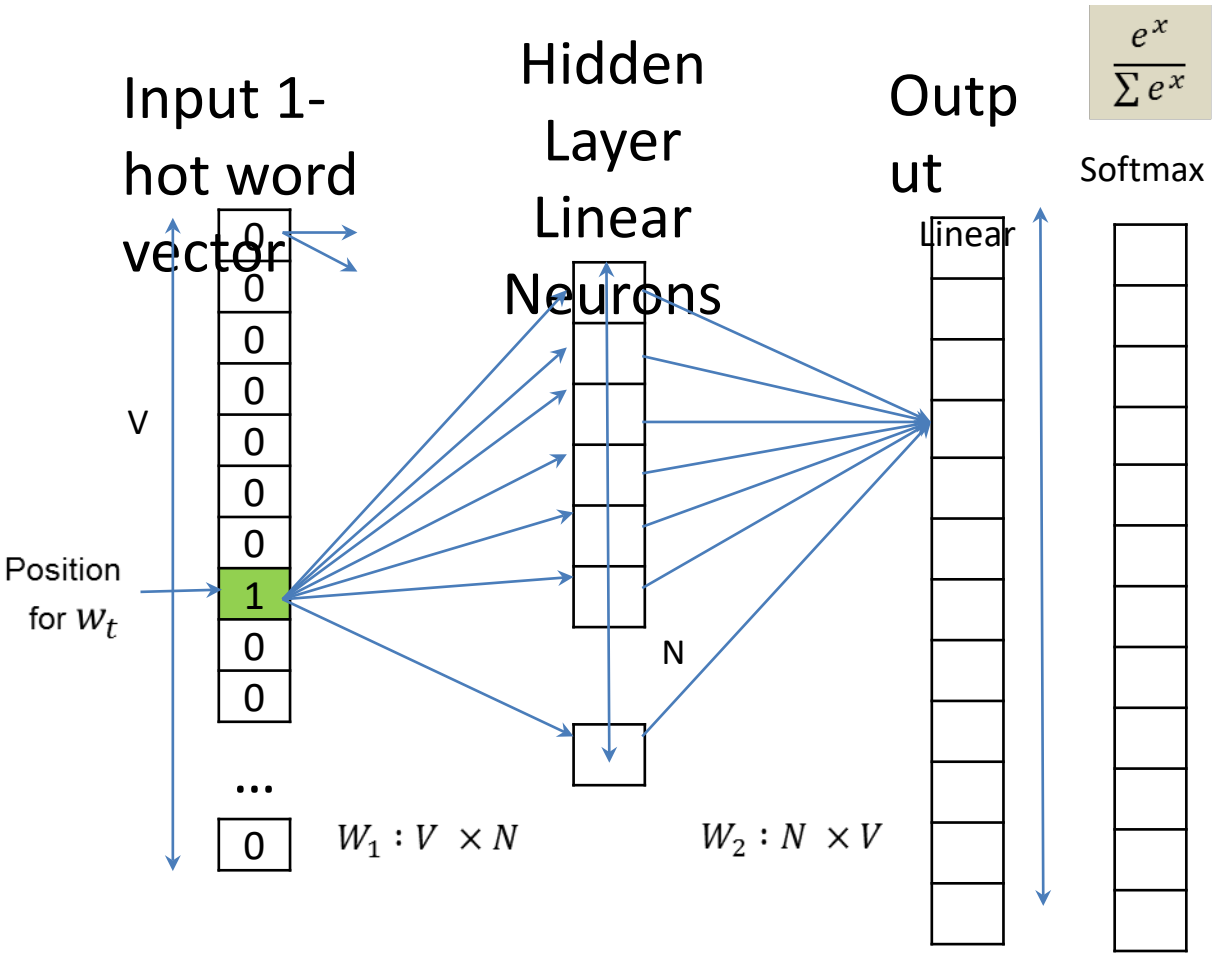
(representation, data)

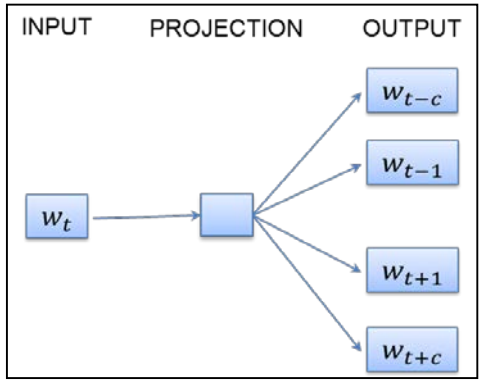
**Negative samples**:

(representation, x)

[x: all other words except the 4 positive]

# Skipgram Model




Probability that the word in a context position Is $w_i$

Positive sampling
Negative sampling

# Skipgram: Loss function

Maximize $\frac{1}{T}\sum_{t=1}^{T}\sum_{\text{context}}\log p(w_{\text{context}}|w_t)$

$p(w_{\text{con}}|w_t)$ is the output of softmax classifier

$$p(w_{Ccon}|w_t) = \frac{\exp\left(v'_{w_{con}}.v_{wt}\right)}{\sum_{w=1}^{W} exp(v'_w.v_{wt})}$$

Let the model parameters be $\theta$. The solution is given by

$$\underset{\theta}{argmax} \sum_{(w_t,w_C)\in D} \log p(w_{con}|w_t;\theta)$$

$$= \sum_{(w_t,w_C)\in D}\left(\log e^{(v'_{w_{con}}.v_{wt})} - \log\sum_{x} e^{(v'_x.v_{wt})}\right)$$

# Skip-Grams with Negative Sampling (SGNS)

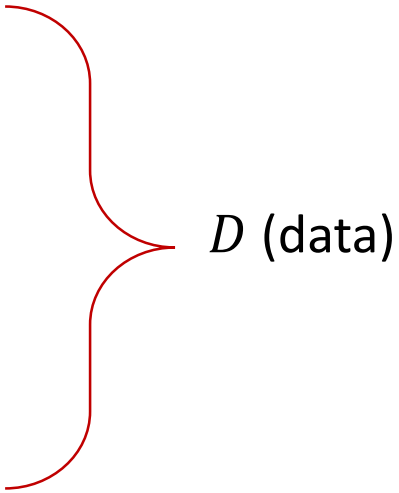
Marco saw a furry little wampimuk hiding in the tree.

"word2vec Explained…"
Goldberg & Levy, arXiv 2014

# Skip-Grams with Negative Sampling (SGNS)

Marco saw a furry little wampimuk hiding in the tree.

| **words** | **contexts** |
|---|---|
| wampimuk | furry |
| wampimuk | little |
| wampimuk | hiding |
| wampimuk | in |
| … | … |




"word2vec Explained…"
Goldberg & Levy, arXiv 2014

# Skip-Grams with Negative Sampling (SGNS)

**Maximize:** $\sigma(\vec{w} \cdot \vec{c})$

- $c$ was **observed** with $w$

| **words** | **contexts** |
|---|---|
| wampimuk | furry |
| wampimuk | little |
| wampimuk | hiding |
| wampimuk | in |

**Minimize:** $\sigma(\vec{w} \cdot \vec{c}')$

- $c'$ was **hallucinated** with $w$

| **words** | **contexts** |
|---|---|
| wampimuk | Australia |
| wampimuk | cyber |
| wampimuk | the |
| wampimuk | 1985 |

"word2vec Explained…" Goldberg & Levy, arXiv 2014

# Some interesting results

## Word Analogies

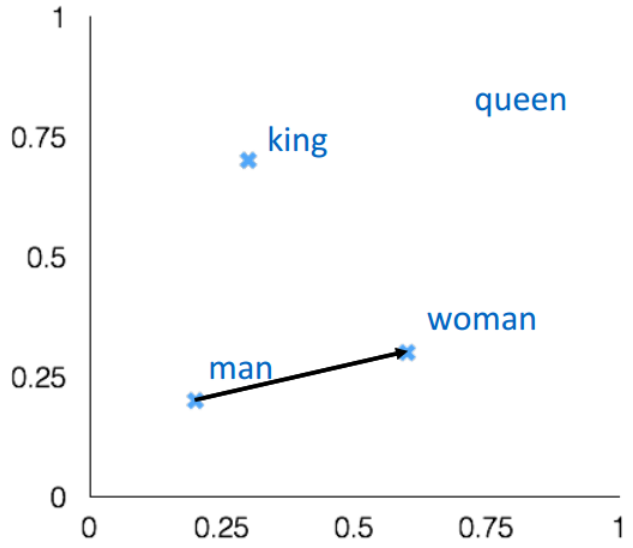
Test for linear relationships, examined by Mikolov et al. (2014)

a:b :: c:?

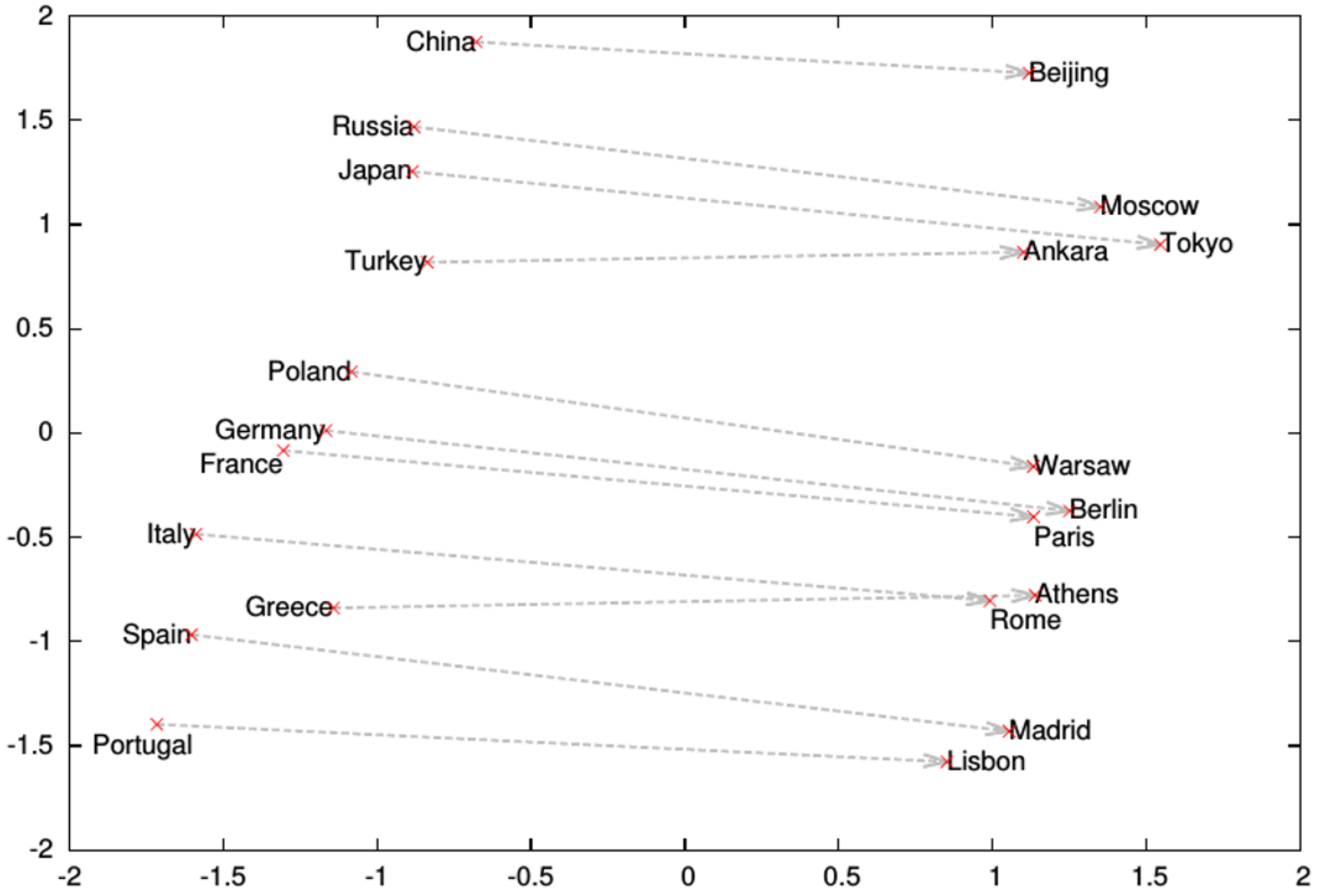
$$d = \arg\max_x \frac{(w_b - w_a + w_c)^T w_x}{||w_b - w_a + w_c||}$$

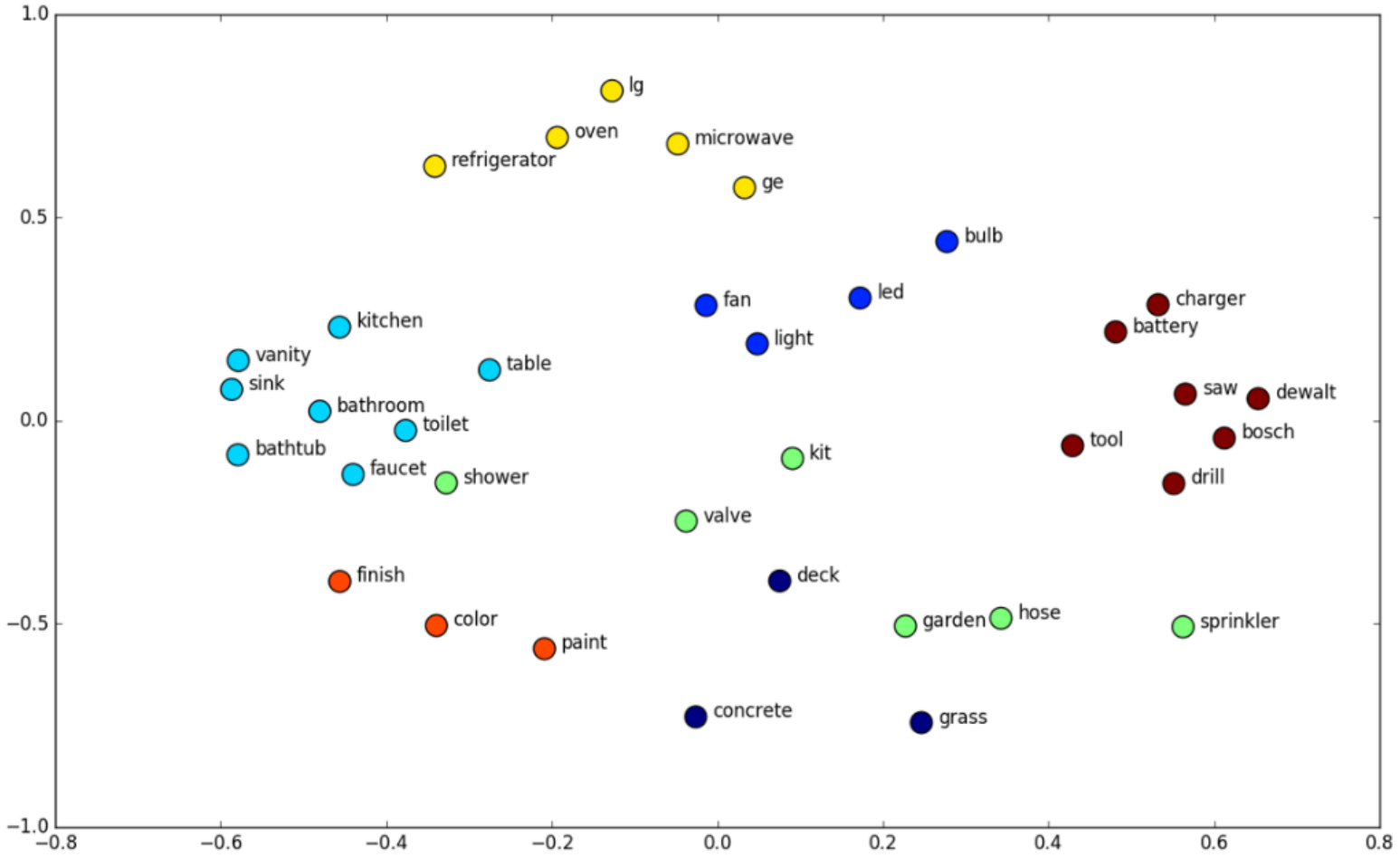
man:woman :: king:?

| | | |
|---|---|---|
| + | king | [ 0.30 0.70 ] |
| - | man | [ 0.20 0.20 ] |
| + | woman | [ 0.60 0.30 ] |
| | queen | [ 0.70 0.80 ] |

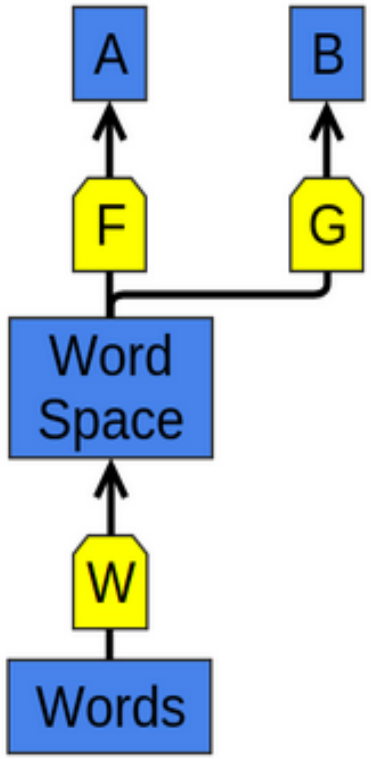


# Word analogies




lg
oven
microwave
refrigerator
ge
bulb
fan
led
charger
kitchen
battery
light
vanity
table
sink
saw
dewalt
bathroom
toilet
bosch
tool
bathtub
kit
faucet
shower
drill
valve
finish
deck
color
garden
hose
sprinkler
paint
concrete
grass
1.0
0.5
0.0
−0.5
−1.0
−0.8
−0.6
−0.4
−0.2
0.0
0.2
0.4
0.6
0.8

# Word embedding applications



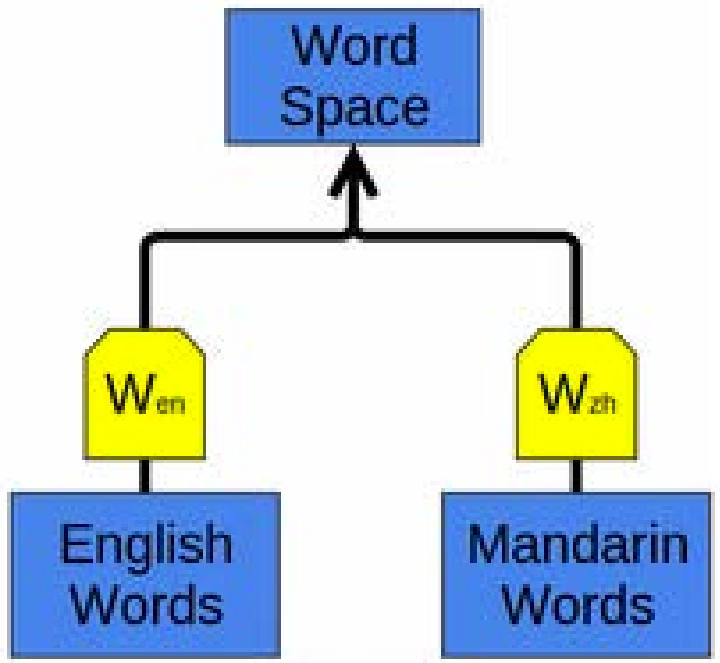

$W$ and $F$ learn to perform task A. Later, $G$ can learn to perform B based on $W$.

- The use of word representations… has become a key "secret sauce" for the success of many NLP systems
  - across tasks including named entity recognition, part-of-speech tagging, parsing, and semantic role labeling
- Learning a good representation on a task A and then using it on a task B is one of the major tricks in the Deep Learning toolbox.
  - Pretraining, transfer learning, and multi-task learning.
  - Can allow the representation to learn from more than one kind of data.

http://colah.github.io/posts/2014-07-NLP-RNNs-Representations/

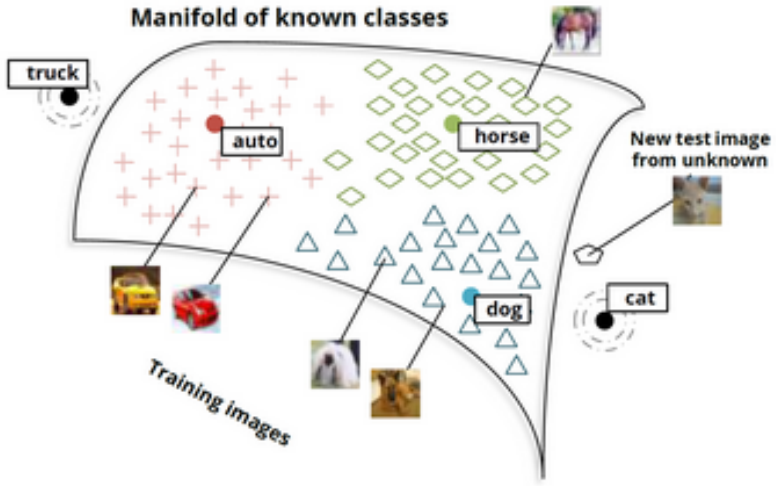
# Word embedding applications

- Can learn to map multiple kinds of data into a single representation.
  - E.g., bilingual word-embedding
- Embed as above, but words that are known as close translations should be close together.
- Words we didn't know were translations end up close together!
- Structures of two languages get pulled into alignment.

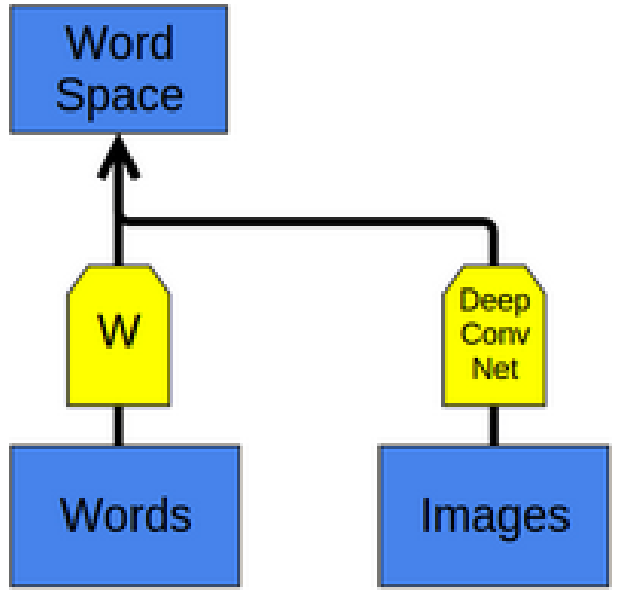


# Word embedding applications

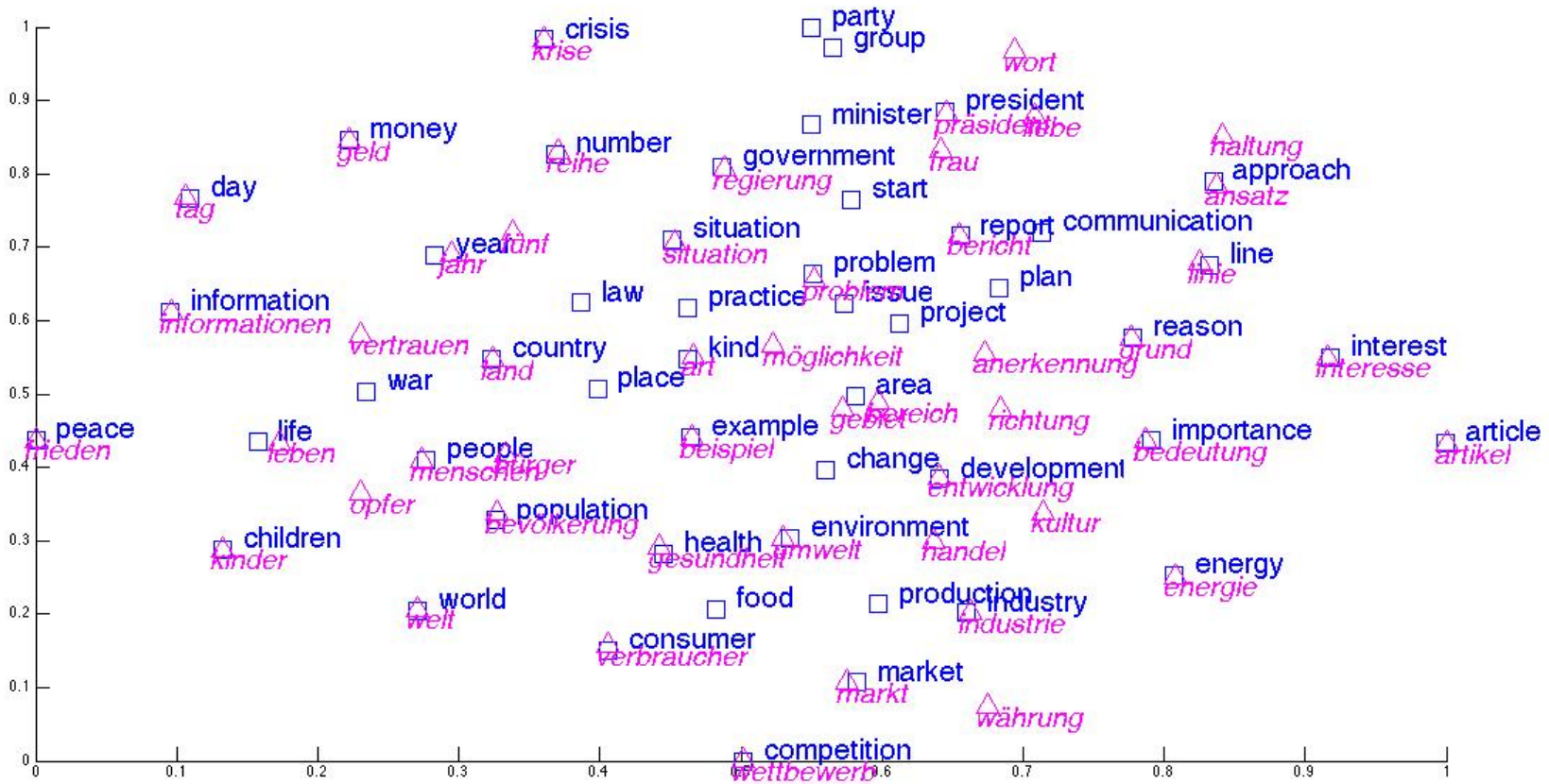
- Can apply to get a joint embedding of words and images or other multi-modal data sets.
- New classes map near similar existing classes:  e.g., if ‘cat’ is unknown, cat images map near dog.




(Socher *et al.* (2013b))



# Multilingual Embeddings




Luong et al 2015

# An interesting application

- Lawrence Berkeley lab material scientists applied word embedding to 3.3 million scientific abstracts published between 1922-2018.
  - 500k words.  Vector size: 200 dimension,skip-gram model
- Captured things like periodic table and structure-property relationship in materials:

  ferromagnetic − NiFe + IrMn ≈ antiferromagnetic
- Discovered new thermoelectric materials

Nature, July 2019 V. Tshitonya et al, “Unsupervised word embeddings capture latent knowledge from materials science literature”.

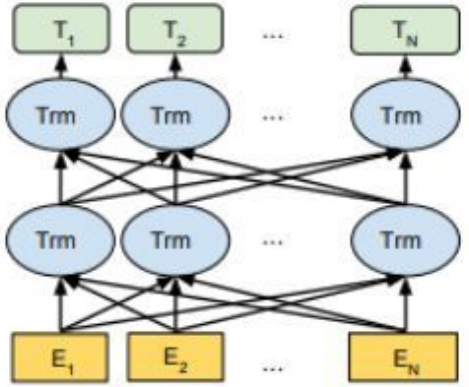
# Contextualized Word Vectors

Incorporating context into word embeddings

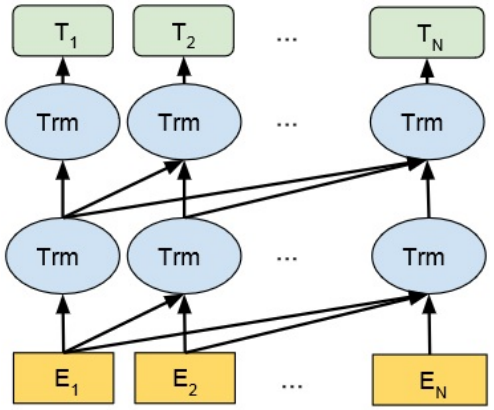
a watershed idea in NLP

- BERT: Bidirectional Encoder Representations from Transformers (BERT, 2018)
- GPT-2/3

Led to significant improvements on virtually every NLP task.




BERT Architecture




GPT

# Language Models

How likely is a sentence $(w_1, w_2, \ldots, w_n)$ ?

- Predict the next word
- Complete the sentence

P (I saw a bus) >> P (eyes awe a boss)

# Pre-Trained Language Models

- Instead of training the model from scratch, you can use another pre-trained model as the basis and only fine-tune it to solve the specific NLP task.

# Multilinguality

mBERT

Google’s multilingual BERT model generates language-independent cross-language sentence embeddings for 109 languages