Natural Language Processing

Info 159/259

Lecture 6: Corpora & Annotation (Feb 5, 2024)

Many slides & instruction ideas borrowed from:

David Bamman & Dan Jurafsky

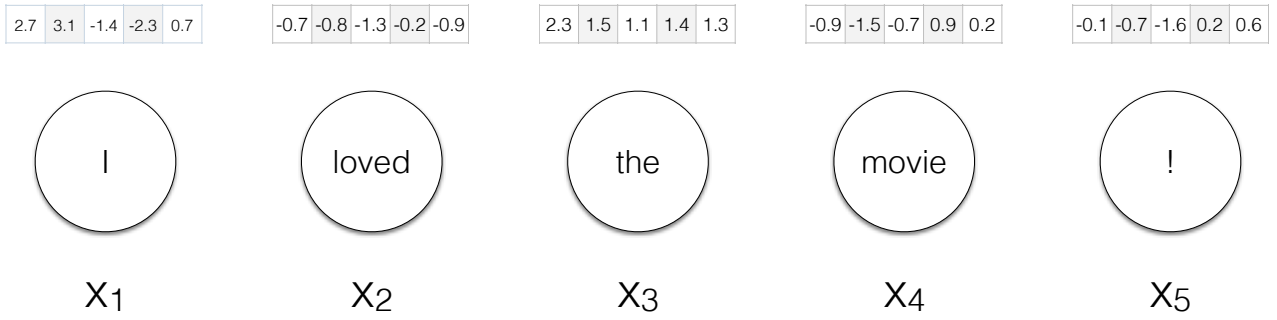
Attention

- Let's incorporate structure (and parameters) into a network that captures which elements (tokens) in the input we should be attending to (and which we can ignore).

$$v \in \mathcal{R}^H$$

2.7	3.1	-1.4	-2.3	0.7

Define v to be a vector to be learned; think of it as an "important word" vector. The dot product here measures how similar each input vector is to that "important word" vector



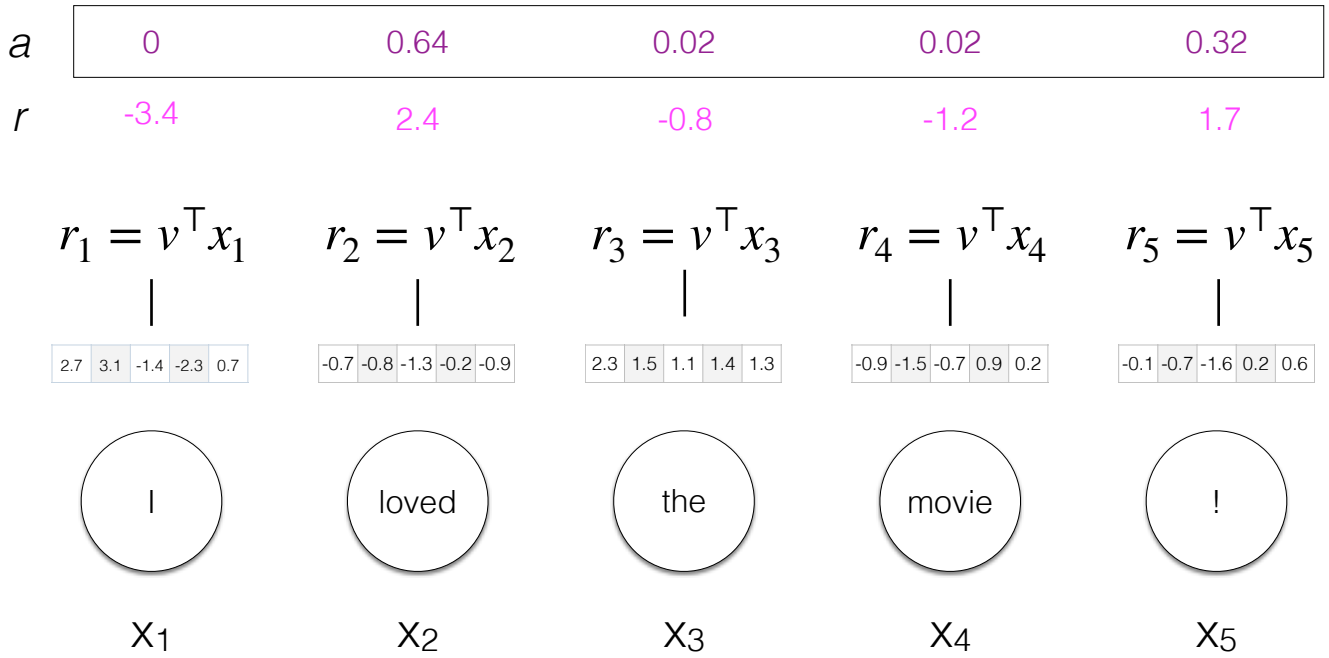
$$v \in \mathcal{R}^H$$

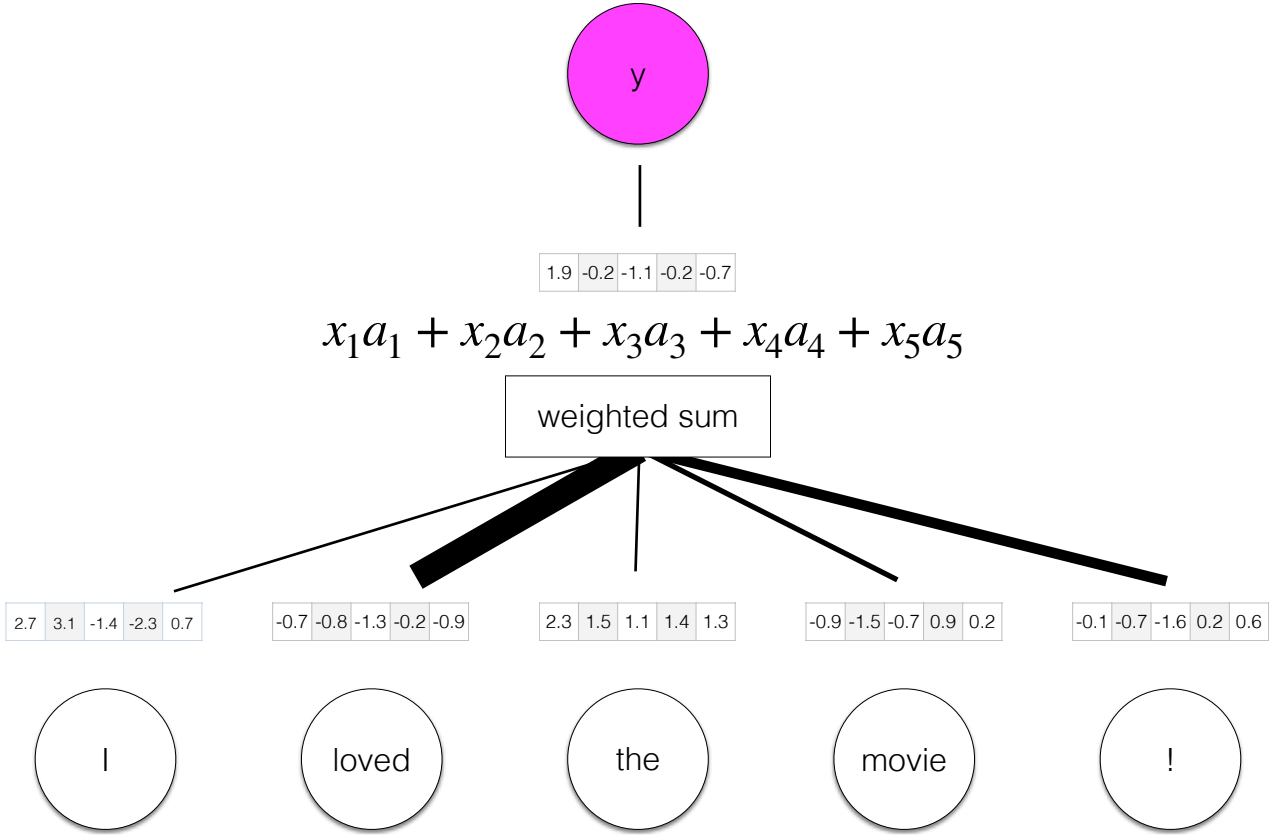
2.7	3.1	-1.4	-2.3	0.7

-3.4	2.4	-0.8	-1.2	1.7
$r_1 = v^\top x_1$	$r_2 = v^\top x_2$	$r_3 = v^\top x_3$	$r_4 = v^\top x_4$	$r_5 = v^\top x_5$
|	|	|	|	|
2.7 3.1 -1.4 -2.3 0.7	-0.7 -0.8 -1.3 -0.2 -0.9	2.3 1.5 1.1 1.4 1.3	-0.9 -1.5 -0.7 0.9 0.2	-0.1 -0.7 -1.6 0.2 0.6
I	loved	the	movie	!
x_1	x_2	x_3	x_4	x_5

Convert r into a vector of normalized weights that sum to 1.

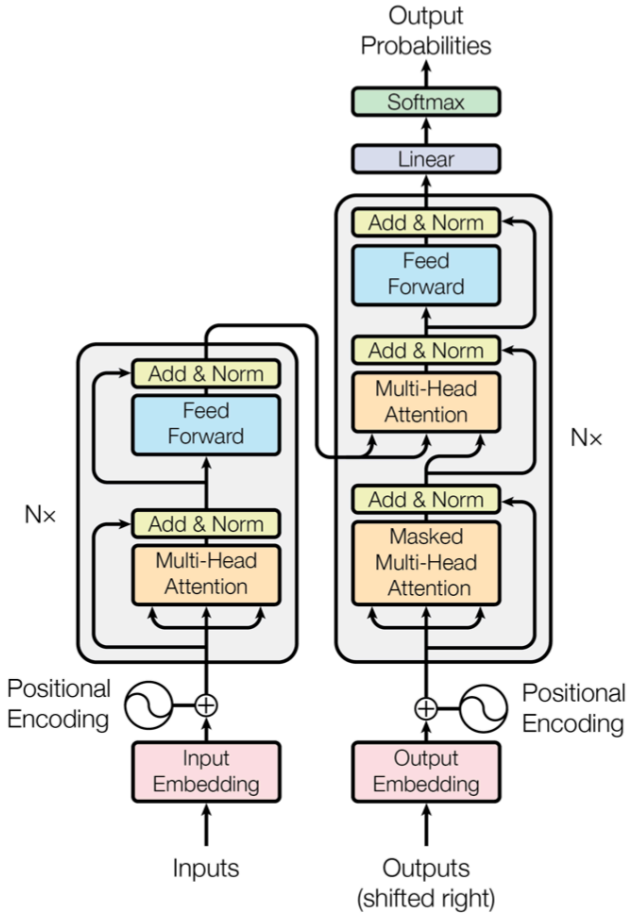
$$a = \text{softmax}(r)$$



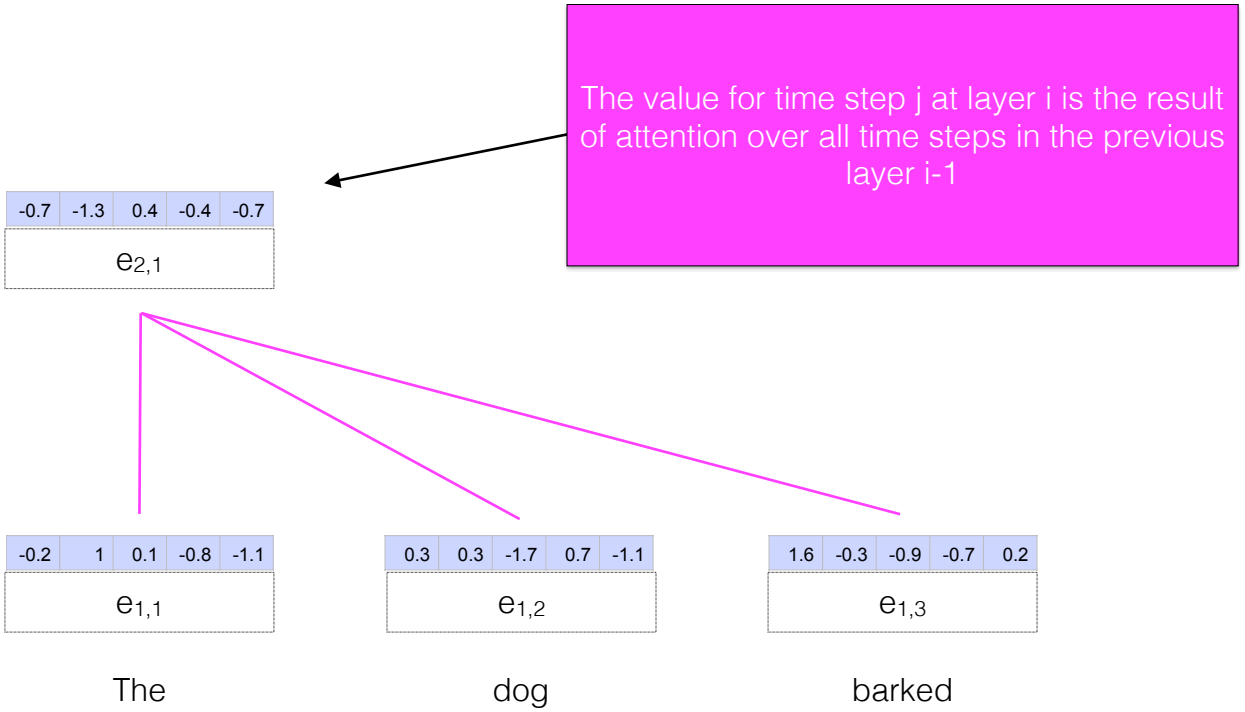

y
1.9 -0.2 -1.1 -0.2 -0.7
$x_1a_1 + x_2a_2 + x_3a_3 + x_4a_4 + x_5a_5$
weighted sum
2.7 3.1 -1.4 -2.3 0.7
-0.7 -0.8 -1.3 -0.2 -0.9
2.3 1.5 1.1 1.4 1.3
-0.9 -1.5 -0.7 0.9 0.2
-0.1 -0.7 -1.6 0.2 0.6
I
loved
the
movie
!

Transformers

- Vaswani et al. 2017, “Attention is All You Need”
- Transforms map an input sequence of vectors to an output sequence of vectors of the same dimensionality



Self-Attention



- Let's separate out the different functions that an input vector has in attention by transforming it into separate representations for its role in a weighted sum (the value) from the roles used to assess compatibility (the query and key).

query

key

value

original value

$q_{1,1} \in \mathbb{R}^{37} \quad (e_{1,1}W^Q)$

$k_{1,1} \in \mathbb{R}^{37} \quad (e_{1,1}W^K)$

$v_{1,1} \in \mathbb{R}^{100} \quad (e_{1,1}W^V)$

$e_{1,1} \in \mathbb{R}^{100}$

$e_{1,1}$

The

$$W^Q \in \mathbb{R}^{100\times37}$$

$$W^K \in \mathbb{R}^{100\times37}$$

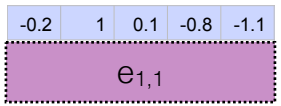
$$W^V \in \mathbb{R}^{100\times100}$$

These are all parameters we *learn.* 100 is the original input dimension; 37 is a hyper-parameter we choose.

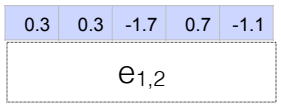
- The compatibility score between two words is the dot product between their respective query and key vectors.

$$score(e_i, e_j) = q_i \cdot k_j$$

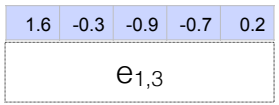
a	0.07	0.58	0.35	a = softmax(scores)
scores	-1.4	0.64	0.14	
	$q_1 \cdot k_1$	$q_1 \cdot k_2$	$q_1 \cdot k_3$	



The

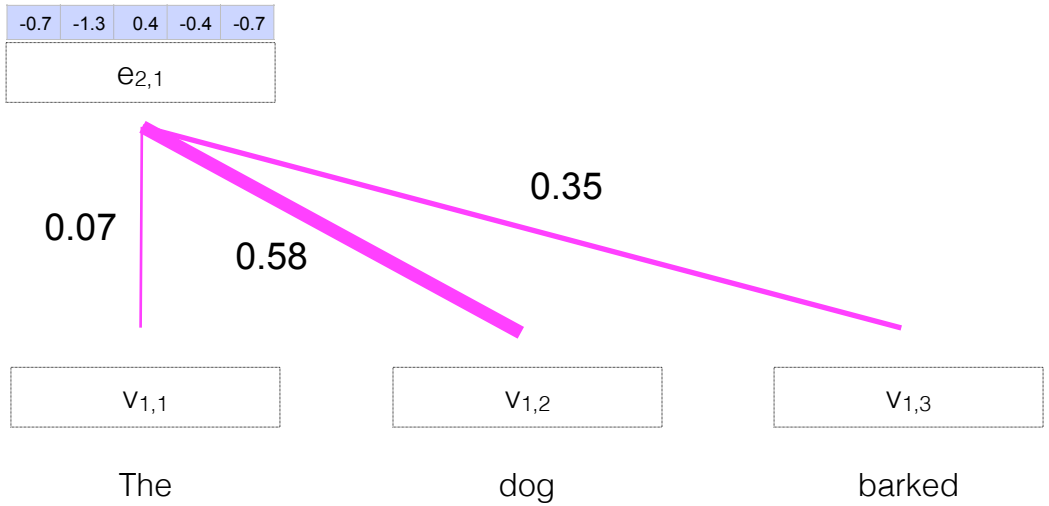


dog



barked

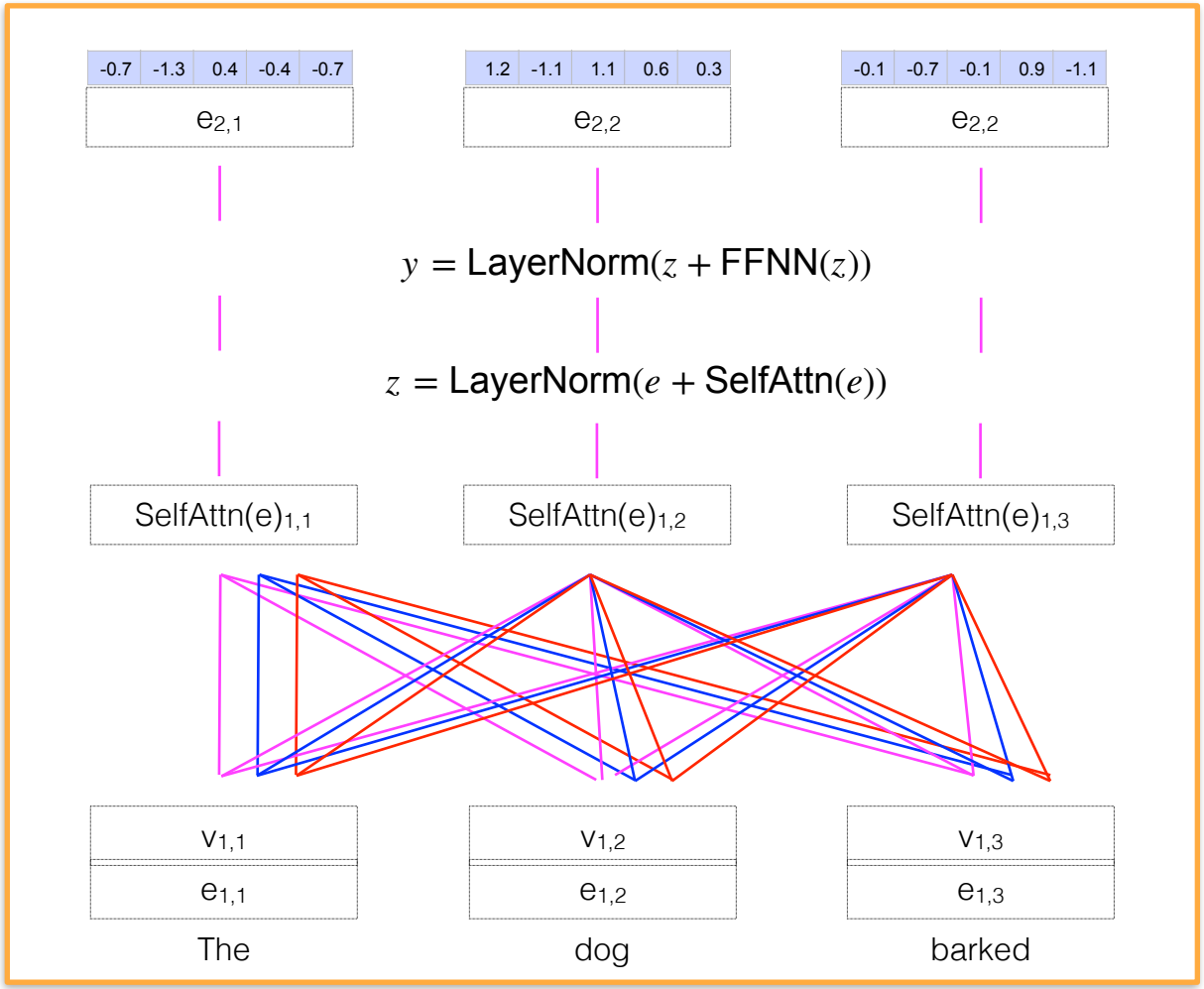
- The output of attention is a weighted sum over the values of the previous layer.



Output

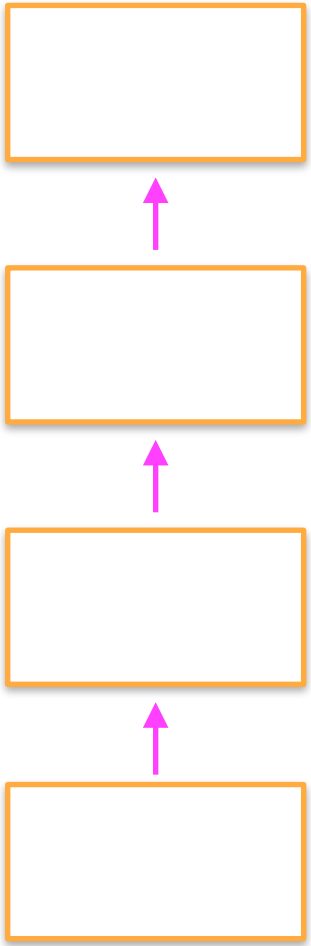
This whole process defines one attention block. The input is a sequence of (e.g. 100-dimensional) vectors; the output of each block is a sequence of (100-dimensional) vectors.

Input

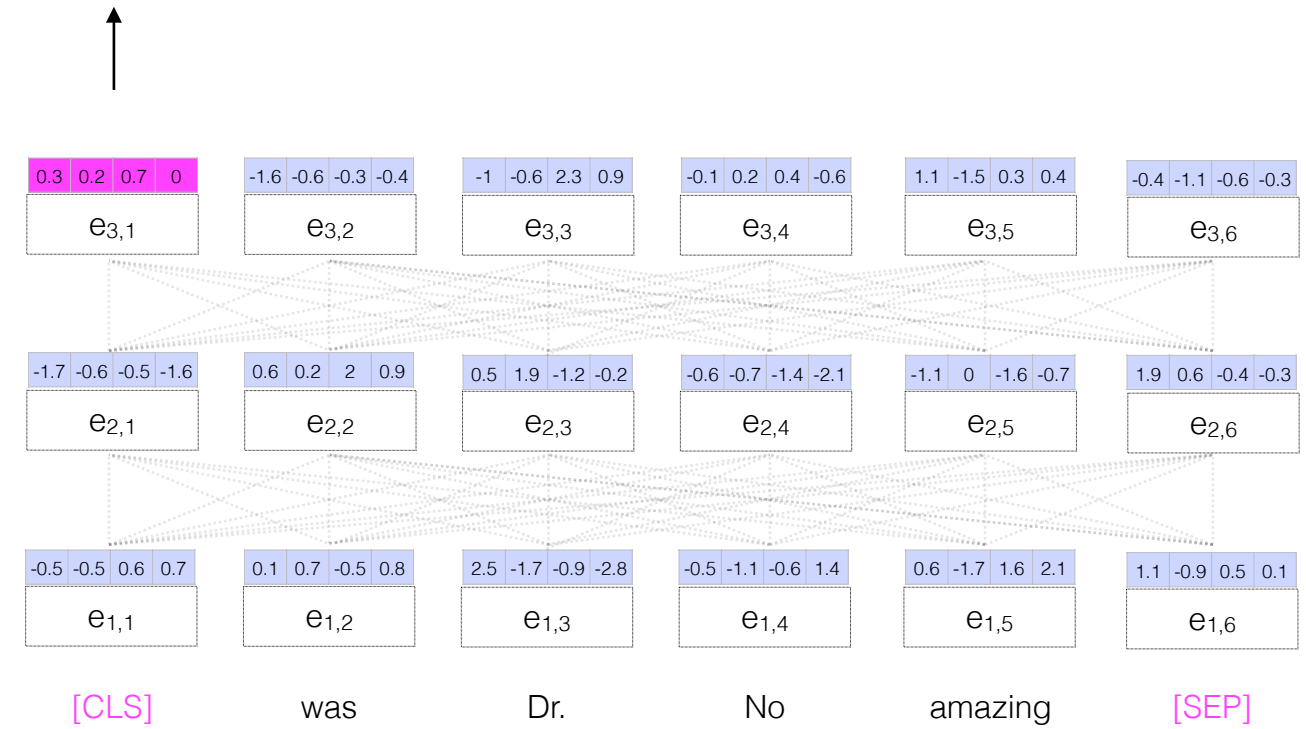


This whole process defines one attention block. The input is a sequence of (e.g. 100-dimensional) vectors; the output of each block is a sequence of (100-dimensional) vectors.

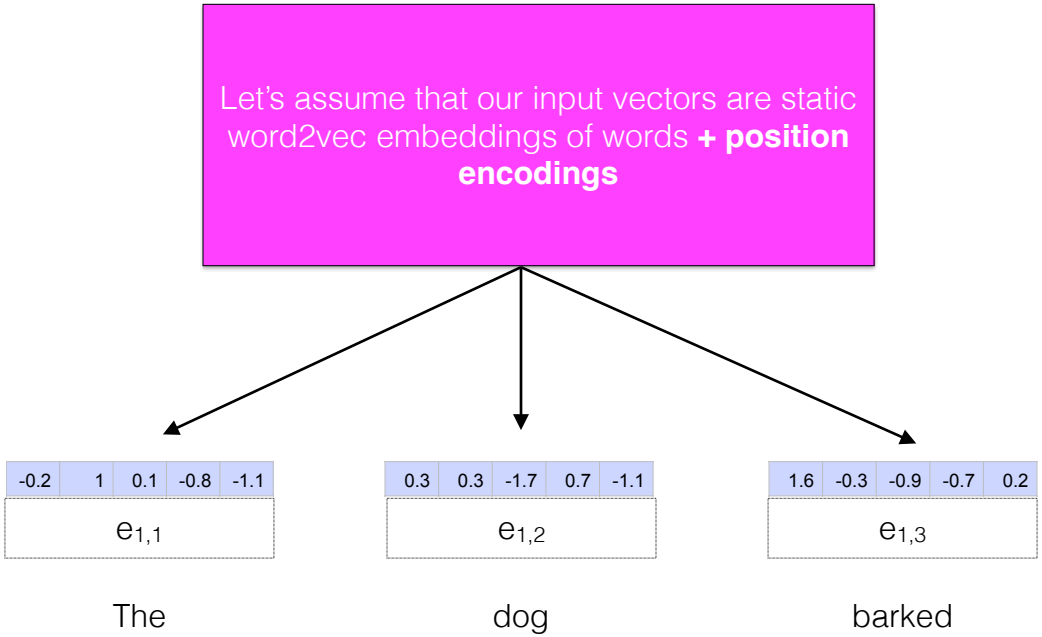
Transformers can stack many such blocks; where the output from block b is the input to block b+1.



- Does a transformer encode any intrinsic information about the order of words within a sequence? Would the output probability for “Dr. No was amazing” be different from “was Dr. No amazing”?



Position encoding



Position embeddings

One option is to add learnable position embeddings pe[i] to each word embedding *e* at position *i* (or concatenate them)

We can add two vectors if they're the same dimensionality

$$e_i = e_i + \text{pe}[i]$$

Or concatenate them if not

$$e_i = e_i \oplus \text{pe}[i]$$

0	2	-0.5	1.1	0.3	0.4	-0.5
1	-1.4	0.4	-0.2	-0.9	0.5	0.9
2	-1.1	-0.2	-0.5	0.2	-0.8	0
3	0.7	-0.3	1.5	-0.3	-0.4	0.1
4	-0.8	1.2	1	-0.7	-1	-0.4
5	0	0.3	-0.3	-0.9	0.2	1.4
6	0.8	0.8	-0.4	-1.4	1.2	-0.9
7	1.6	0.4	-1.1	0.7	0.1	1.6
…	…	…	…	…	…	…

position embeddings (pe)

Transformers

- Transformers have been extremely influential in NLP (Vaswani et al. 2017 has 35K citations!)
- We'll see them much more in this class in the context of specific applications:
    - Contextual language models, including causal self-attention (GPT), and bidirectional attention (BERT).
    - Machine translation
    - Text generation

Natural Language Processing

Info 159/259

Lecture 6: Annotation (Feb 5, 2024)

Many slides & instruction ideas borrowed from:

David Bamman & Dan Jurafsky

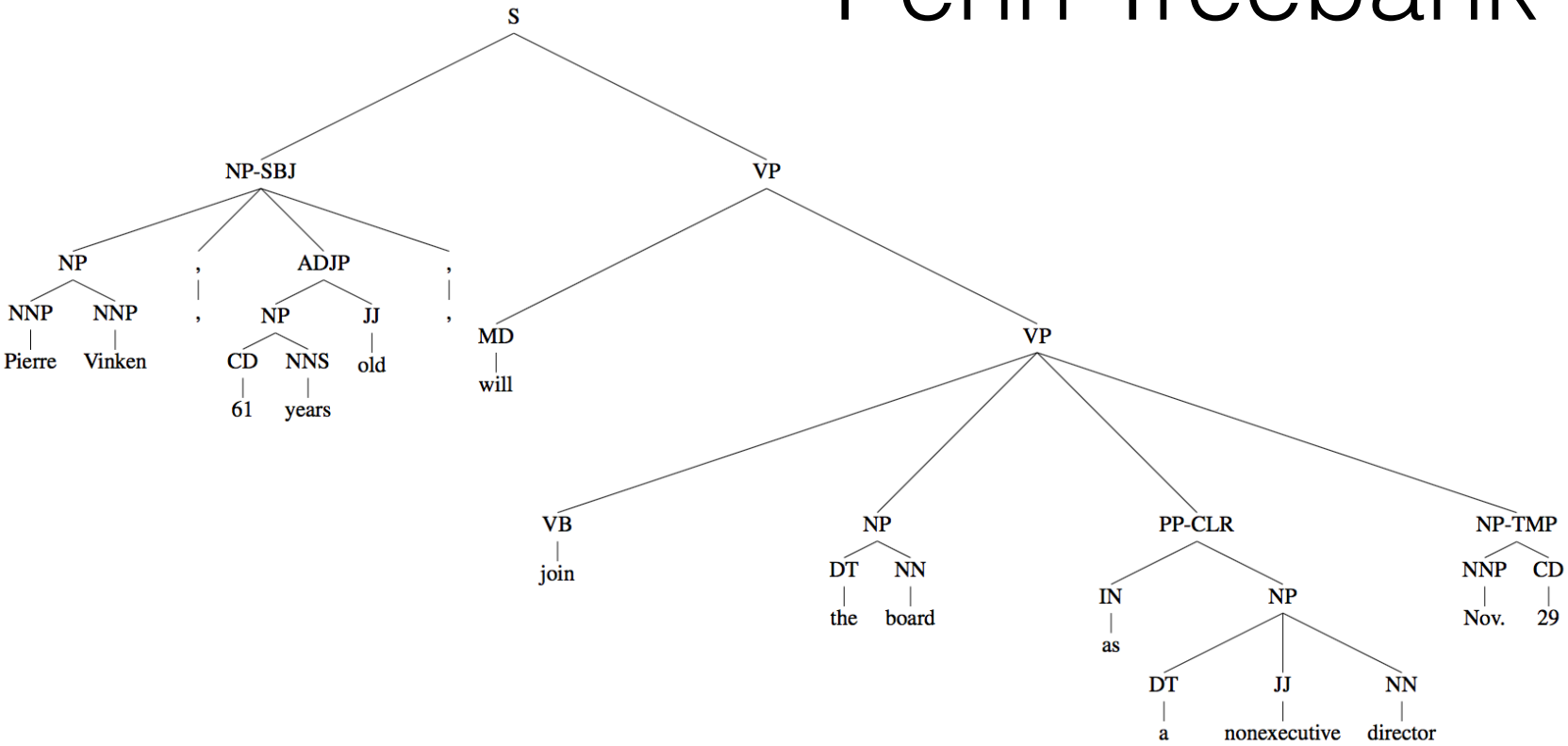
Modern NLP is driven by annotated data

- Penn Treebank (1993; 1995;1999); morphosyntactic annotations of WSJ
- OntoNotes (2007–2013); syntax, predicate-argument structure, word sense, coreference
- FrameNet (1998–): frame-semantic lexical annotations
- MPQA (2005): opinion/sentiment
- SQuAD (2016): annotated questions + spans of answers in Wikipedia

Modern NLP is driven by annotated data

- In most cases, the data we have is the product of human judgments.
    - What's the correct part of speech tag?
    - Syntactic structure?
    - Sentiment?

Penn Treebank



Propbank

(22.11) **agree.01**

Arg0: Agreer
Arg1: Proposition
Arg2: Other entity agreeing

Ex1: [$_{\text{Arg0}}$ The group] *agreed* [$_{\text{Arg1}}$ it wouldn't make an offer].
Ex2: [$_{\text{ArgM-TMP}}$ Usually] [$_{\text{Arg0}}$ John] *agrees* [$_{\text{Arg2}}$ with Mary] [$_{\text{Arg1}}$ on everything].

(22.12) **fall.01**

Arg1: Logical subject, patient, thing falling
Arg2: Extent, amount fallen
Arg3: start point
Arg4: end point, end state of arg1
Ex1: [$_{\text{Arg1}}$ Sales] *fell* [$_{\text{Arg4}}$ to $25 million] [$_{\text{Arg3}}$ from $27 million].
Ex2: [$_{\text{Arg1}}$ The average junk bond] *fell* [$_{\text{Arg2}}$ by 4.2%].

Squad

In meteorology, precipitation is any product of the condensation of atmospheric water vapor that falls under **gravity**. The main forms of precipitation include drizzle, rain, sleet, snow, **graupel** and hail... Precipitation forms as smaller droplets coalesce via collision with other rain drops or ice crystals **within a cloud**. Short, intense periods of rain in scattered locations are called "showers".

What causes precipitation to fall?
gravity

What is another main form of precipitation besides drizzle, rain, snow, sleet and hail?
graupel

Where do water droplets collide with ice crystals to form precipitation?
within a cloud

Rajpurkar et al 2016

Dogmatism

Fast and Horvitz (2016), "Identifying Dogmatism in Social Media: Signals and Models"

Given a comment, imagine you hold a well-informed, different opinion from the commenter in question. We'd like you to tell us how likely that commenter would be to engage you in a constructive conversation about your disagreement, where you each are able to explore the other's beliefs. The options are:

(5): It's unlikely you'll be able to engage in any substantive conversation. When you respectfully express your disagreement, they are likely to ignore you or insult you or otherwise lower the level of discourse.

(4): They are deeply rooted in their opinion, but you are able to exchange your views without the conversation degenerating too much.

(3): It's not likely you'll be able to change their mind, but you're easily able to talk and understand each other's point of view.

(2): They may have a clear opinion about the subject, but would likely be open to discussing alternative viewpoints.

(1): They are not set in their opinion, and it's possible you might change their mind. If the comment does not convey an opinion of any kind, you may also select this option.

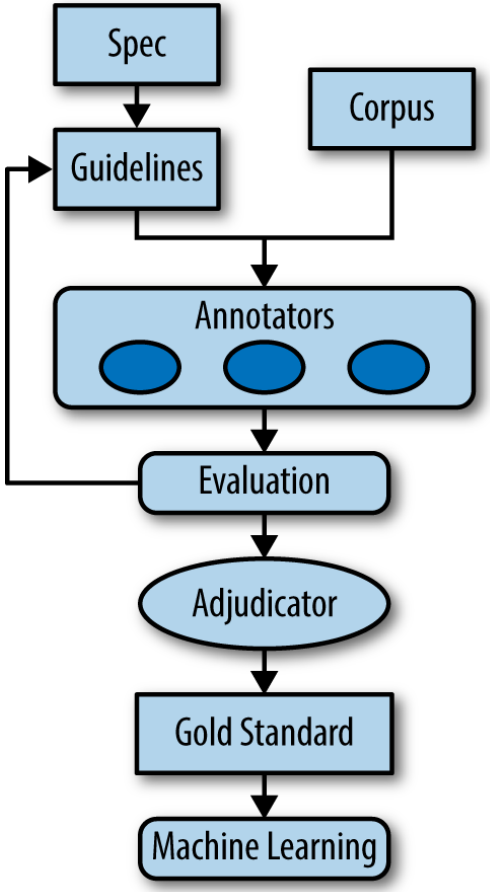
Sarcasm

“In many respects you know they honor President Obama. ISIS is honoring President Obama! He is the founder of ISIS. He’s the founder of ISIS, O.K.! He’s the founder, he founded ISIS and I would say the co-founder would be crooked Hillary Clinton. Co-founder, crooked Hillary Clinton. And that’s what it’s about.”

Fake News

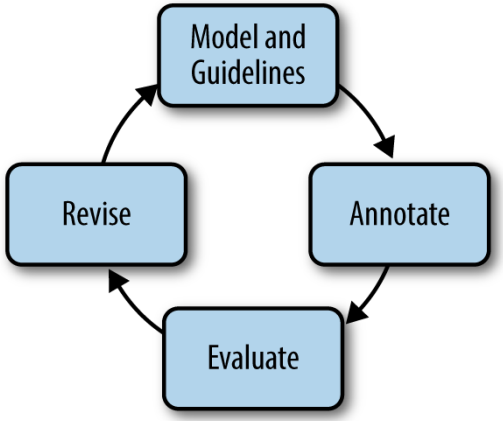
http://www.fakenewschallenge.org

Annotation pipeline



Pustejovsky and Stubbs (2012),
Natural Language Annotation for Machine Learning

Annotation pipeline



Pustejovsky and Stubbs (2012),
Natural Language Annotation for Machine Learning

Annotation guidelines

- Our goal: given the constraints of our problem, how can we formalize our description of the annotation process to encourage multiple annotators to provide the same judgment?

Annotation guidelines

- What is the goal of the project?
- What is each tag called and how is it used? (Be specific: provide examples, and discuss gray areas.)
- What parts of the text do you want annotated, and what should be left alone?
- How will the annotation be created? (For example, explain which tags or documents to annotate first, how to use the annotation tools, etc.)

Pustejovsky and Stubbs (2012), Natural Language Annotation for Machine Learning

Why not do it alone?

- Expensive/time-consuming
- Multiple people provide a measure of consistency: is the task well enough defined?
- Low agreement = not enough training, guidelines not well enough defined, task is bad

Adjudication

- Adjudication is the process of deciding on a single annotation for a piece of text, using information about the independent annotations.
- Can be as time-consuming (or more so) as a primary annotation.
- Does not need to be identical with a primary annotation (both annotators can be wrong by chance)

Inter-annotator agreement

annotator A

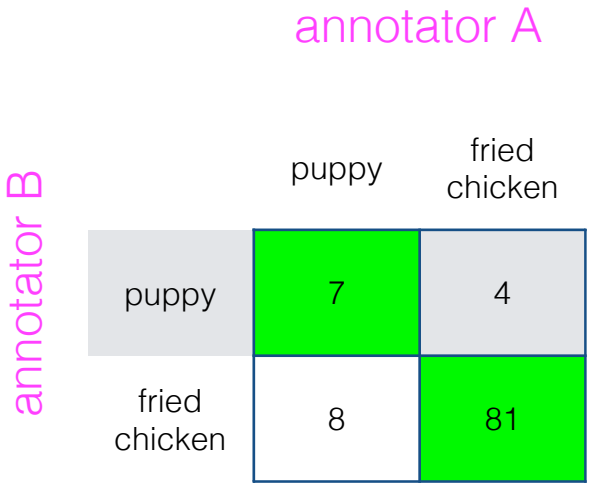
annotator B

	puppy	fried chicken
puppy	6	3
fried chicken	2	5

observed agreement = 11/16 = 68.75%

Cohen's kappa

- If classes are imbalanced, we can get high inter annotator agreement simply by chance

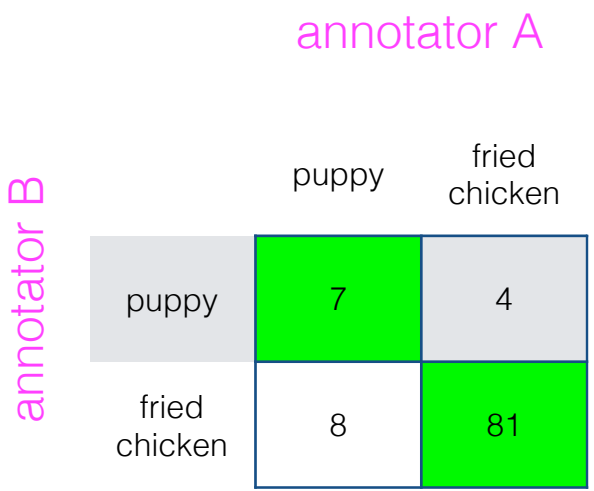


Cohen's kappa

- If classes are imbalanced, we can get high inter annotator agreement simply by chance

$$\kappa = \frac{p_o - p_e}{1 - p_e}$$

$$\kappa = \frac{0.88 - p_e}{1 - p_e}$$



Cohen's kappa

- Expected probability of agreement is how often we would expect two annotators to agree assuming independent annotations

$$p_e = P(A = \text{puppy}, B = \text{puppy}) + P(A = \text{chicken}, B = \text{chicken})$$

$$= P(A = \text{puppy})P(B = \text{puppy}) + P(A = \text{chicken})P(B = \text{chicken})$$

Cohen's kappa

$$= P(A = \text{puppy})P(B = \text{puppy}) + P(A = \text{chicken})P(B = \text{chicken})$$

P(A=puppy)	15/100 = 0.15
P(B=puppy)	11/100 = 0.11
P(A=chicken)	85/100 = 0.85
P(B=chicken)	89/100 = 0.89

$$= 0.15 \times 0.11 + 0.85 \times 0.89$$
$$= 0.773$$

annotator A

annotator B

	puppy	fried chicken
puppy	7	4
fried chicken	8	81

Cohen's kappa

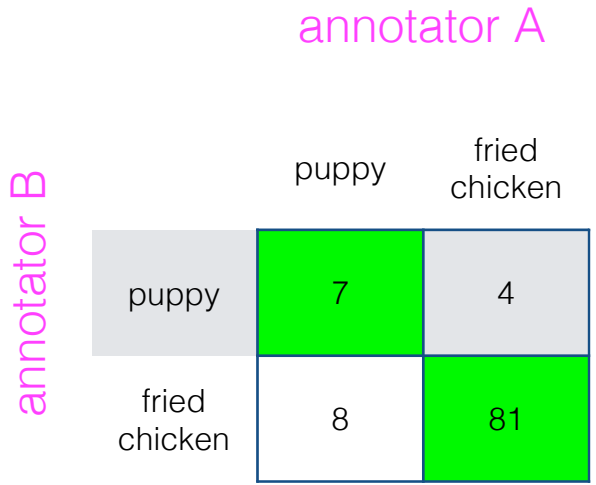
- If classes are imbalanced, we can get high inter annotator agreement simply by chance

$$\kappa = \frac{p_o - p_e}{1 - p_e}$$

$$\kappa = \frac{0.88 - p_e}{1 - p_e}$$

$$\kappa = \frac{0.88 - 0.773}{1 - 0.773}$$

$$= 0.471$$



Cohen’s kappa

- “Good” values are subject to interpretation, but rule of thumb:

0.80-1.00	Very good agreement
0.60-0.80	Good agreement
0.40-0.60	Moderate agreement
0.20-0.40	Fair agreement
< 0.20	Poor agreement

Inter-annotator agreement

- Cohen's kappa can be used for any number of classes.
- Still requires two annotators who evaluate the same items.
- Fleiss' kappa generalizes to multiple annotators, each of whom may evaluate different items (e.g., crowdsourcing)
- Krippendorf's alpha: Going from categorical labels to real valued
    - Ordinal numbers (review scores).

Fleiss' kappa

- Same fundamental idea of measuring the observed agreement compared to the agreement we would expect by chance.

- With N > 2, we calculate agreement among pairs of annotators

$$\kappa = \frac{P_o - P_e}{1 - P_e}$$

Fleiss' kappa

$$\kappa = \frac{P_o - P_e}{1 - P_e}$$

Average agreement among all items

$$P_o = \frac{1}{N} \sum_{i=1}^{N} P_i$$

Expected agreement by chance — joint probability two raters pick the same label is the product of their independent probabilities of picking that label

$$P_e = \sum_{j=1}^{K} p_j^2$$

Fleiss' kappa

Number of annotators (pairs) who assign category *j* to item *i*

$$n_{ij}$$

For item *i* with *n* annotations, how many annotators (pairs) agree, among all n(n-1) possible pairs

$$P_i = \frac{1}{n(n-1)} \sum_{j=1}^{K} n_{ij}(n_{ij} - 1)$$

Fleiss' kappa

For item *i* with *n* annotations, how many annotators agree, among all n(n-1) possible pairs

$$P_i = \frac{1}{n(n-1)} \sum_{j=1}^{K} n_{ij}(n_{ij} - 1)$$

Annotator

A	B	C	D
+	+	+	-

agreeing pairs of annotators →

A-B
B-A
A-C
C-A
B-C
C-B

Label	n_{ij}
+	3
-	1

$$P_i = \frac{1}{4(3)}(3(2) + 1(0))$$

Fleiss' kappa

Average agreement among all items

$$P_o = \frac{1}{N}\sum_{i=1}^{N} P_i$$

Number of annotators (pairs) who assign category *j* to item *i*

$$n_{ij}$$

Probability of category *j*

$$p_j = \frac{1}{Nn}\sum_{i=1}^{N} n_{ij}$$

Expected agreement by chance — joint probability two raters pick the same label is the product of their independent probabilities of picking that label

$$P_e = \sum_{j=1}^{K} p_j^2$$

Fleiss' kappa

- Same fundamental idea of measuring the observed agreement compared to the agreement we would expect by chance.

$$\kappa = \frac{P_o - P_e}{1 - P_e}$$