Start Recording

- Today:
  - Tries
  - Corpus linguistics
  - Annotation
  - Argmax

Reminders

- Project 2 due tonight at 11:45 PM
- Project 3 due 8/21
- Assignment 3 due 8/23

Assignment 2

- Arithmetic errors
- Incorrect guesses– precision matters!

Binary trees

- Tree data structures are useful when you'll be doing a lot of inserting and deleting and you want to retain $O(\log n)$ access
  - Don't confuse with decision tree classifiers, which you'll get to implement in 572
- In practice, I have found that HashMaps (dictionaries) are better suited for the computational linguistics problems I've encountered
- However, one problem that cannot be solved with everyday data structures is the "all substrings" problem
  - For this, we'll look at a special kind of tree, the prefix trie, or simply "trie"

Trie

- Also known as “prefix trie”
- Pronounced either way: “tree” or “try”
- You will implement a trie for Project 4
- A trie is the most efficient way to search for multiple arbitrary-length substrings in a string

Example

- Identify an arbitrary number of string features in web browser user agent strings

```
Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:2.0a1pre) Gecko/2008041102 Minefield/4.0a1pre
Mozilla/5.0 (Windows; U; Windows NT 5.1; zh-CN; rv:1.9) Gecko/2008052906 Firefox/3.0
Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 5.1; Trident/4.0; .NET CLR 1.1.4322; .NET CLR 2.0.50727; .NET CLR 3.0.4506.2152; .NET CLR 3.5.30729)
Mozilla/5.0 (compatible; OSS/1.0; Chameleon; Linux) MOT-U9/R6632_G_81.11.29R BER/2.0 Profile/MIDP-2.0 Configuration/CLDC-1.1
SAMSUNG-SGH-E250/1.0 Profile/MIDP-2.0 Configuration/CLDC-1.1 UP.Browser/6.2.3.3.c.1.101 (GUI) MMP/2.0
LG/U8130/v1.0
SonyEricssonC901/R1EA Browser/NetFront/3.4 Profile/MIDP-2.1 Configuration/CLDC-1.1 JavaPlatform/JP-8.4.2
ZTE-V8301/MB6801_V1_Z1_VN_F1BPa101 Profile/MIDP-2.0 Configuration/CLDC-1.1 Obigo/Q03C
Mozilla/5.0 (iPhone; U; CPU iPhone OS 3_0 like Mac OS X; en-us) AppleWebKit/420.1 (KHTML, like Gecko) Version/3.0 Mobile/1A542a Safari/419.3
Mozilla/5.0 (iPhone; U; CPU iPhone OS 4_0 like Mac OS X; en-us) AppleWebKit/532.9 (KHTML, like Gecko) Version/4.0.5 Mobile/8A293 Safari/6531.22.7
Mozilla/5.0 (iPod; U; CPU iPhone OS 3_1_1 like Mac OS X; en-us) AppleWebKit/528.18 (KHTML, like Gecko) Mobile/7C145
Mozilla/5.0 (iPad; U; CPU OS 3_2 like Mac OS X; en-us) AppleWebKit/531.21.10 (KHTML, like Gecko) Version/4.0.4 Mobile/7B367 Safari/531.21.10
SIE-S68/36 UP.Browser/7.1.0.e.18 (GUI) MMP/2.0 Profile/MIDP-2.0 Configuration/CLDC-1.1
SIE-EF81/58 UP.Browser/7.0.0.1.181 (GUI) MMP/2.0 Profile/MIDP-2.0 Configuration/CLDC-1.1
9700 Bold: BlackBerry9700/5.0.0.423 Profile/MIDP-2.1 Configuration/CLDC-1.1 VendorID/100
```

Problem statement

```
String[] targets =
{ "Mozilla/5.0", "Trident/4.0", "Firefox/3",
    "iPhone", "Mac OS X", "BlackBerry", "Safari" };

String ua = "Mozilla/5.0 (iPod; U; CPU iPhone OS 3_1_1 like Mac OS X; en-us) AppleWebKit/528.18 (KHTM
foreach (String feat in AllSubstrings(ua, targets))
    Console.WriteLine(feat);

//...

IEnumerable<String> AllSubstrings(String s_in, IEnumerable<String> targets)
{
    yield return "hmm...";
}
```

$O(n^2)$?

Naïve solution

- This is the first solution that should come to mind
  - It is appropriate for small inputs
  - It is easy to understand and trivial to implement
  - Test this on the input set. Maybe it's adequate
  - It provides a baseline for optimization work

```
IEnumerable<String> AllSubstrings(String s_in, IEnumerable<String> targets)
{
    for (int i0 = 0; i0 < s_in.Length; i0++)
        foreach (String t in targets)
            if (i0 + t.Length <= s_in.Length && s_in.Substring(i0, t.Length) == t)
                yield return t;
}
```

- Ok, what if it's not adequate?

Using a trie for the all substrings problem

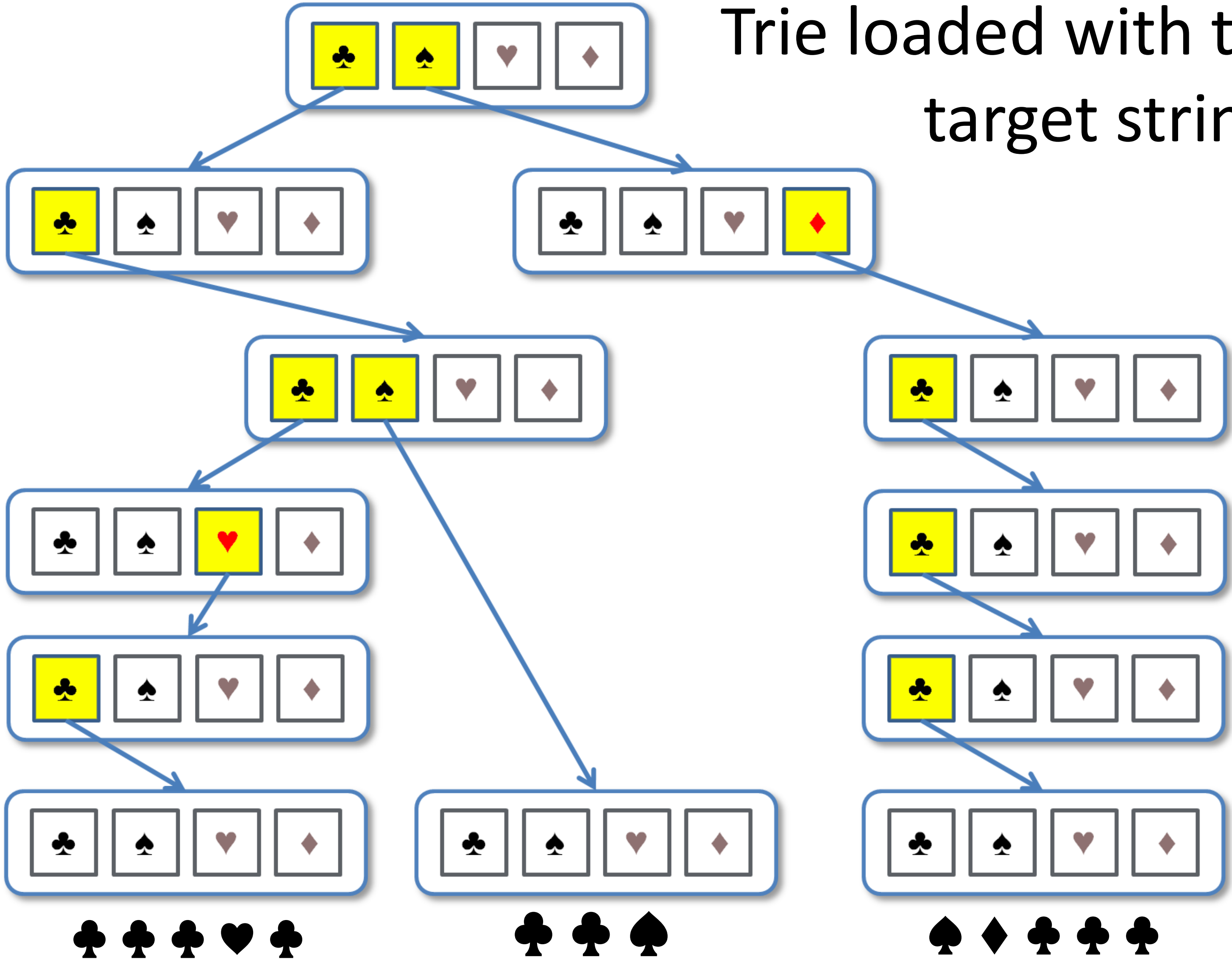
1. Build a trie from the target strings
   – The targets are all now stored implicitly in the trie, you can release them
2. Scan through the corpus once, using the trie to simultaneously check for all targets

Trie example

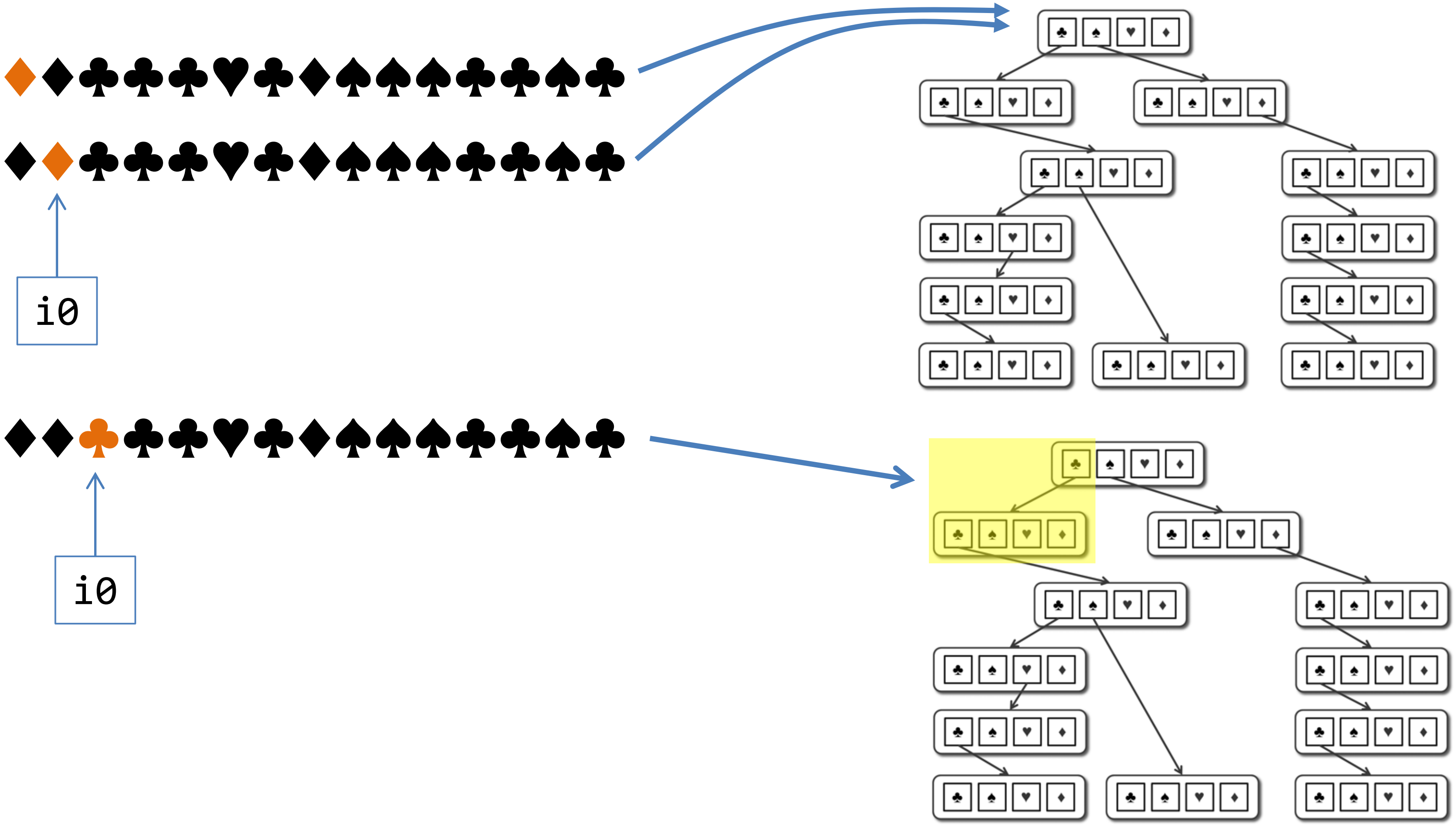
```
String[] targets = { "♣♣♠", "♣♣♣♥♣", "♠♦♣♣♣" };


String corpus = "♦♦♣♣♣♥♣♦♠♠♠♣♣♠♣";
//               0         1
//               012345678901234
```

- target “♣♣♠” found at position 11
- target “♣♣♣♥♣” found at position 2

Trie loaded with the target strings



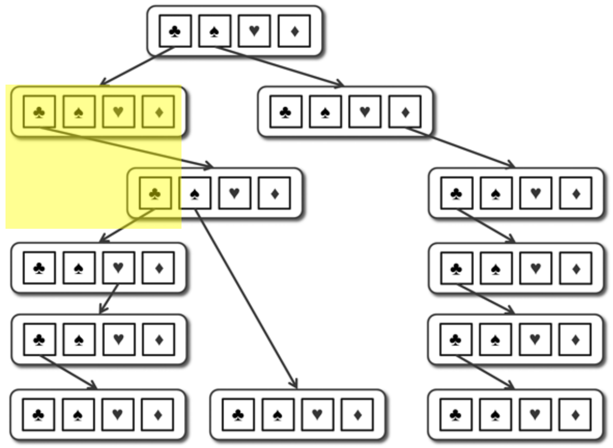
slide adapted from Glenn Slaydon


♦♦♣♣♣♥♣♦♠♠♠♣♣♠♣
♦♦♣♣♣♥♣♦♠♠♠♣♣♠♣
i0
♦♦♣♣♣♥♣♦♠♠♠♣♣♠♣
i0

♦♦♣♣♣♥♣♦♠♠♠♣♣♠♣

i0 i1

At this point, we're simultaneously checking two active targets, ♣♣♠ and ♣♣♣♥♣

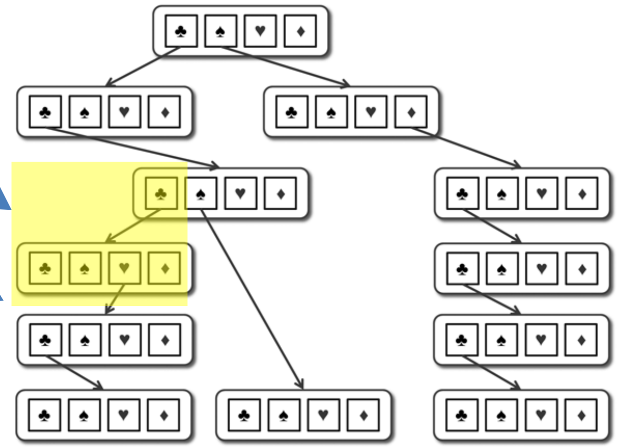


i1

♦♦♣♣♣♥♣♦♠♠♠♣♣♠♣

♦♦♣♣♣♥♣♦♠♠♠♣♣♠♣

♦♦♣♣♣♥♣♦♠♠♠♣♣♠♣



Found: `corpus.Substring(i0, i1 – i0 + 1)`
Output and continue from `i0 + 1`

Walking through a trie

- To use a trie for finding all substrings, for each character c_i:
  - attempt to navigate the trie while moving forward $c_{j+1 \ldots}$, returning substrings, as they are found in the trie
  - After reaching the end of the trie, advance to next character c_{i+1} and repeat
- Recursion is not necessary because we always progress downwards from the trie root
  - You only need a reference to the “current” trie node
  - For this same reason, a trie node does not need to have a back-pointer to its parent node

Trie applications

- A trie guarantees that there is exactly one unique node for every possible prefix of every target
- If several targets share the same prefix, they share those trie nodes

Q: Considering this, what characteristics would a set of target strings have, such that a trie is a good candidate data structure?

Corpus linguistics

> The fundamental goal of analysis is to maximize the probability of the observed data.
>
> John Goldsmith, Univ. of Chicago

- Data is important
- It makes (machine) learning possible
- In computational linguistics, our data is organized into corpora. This word is the plural of corpus.

Data statements

- •Corpora usually include metadata describing the data and the population it came from
- •When you ***use*** a corpus to do anything in NLP, you should include a ***data statement*** that describes:
  - *Whose* language is represented
  - *Who* annotated the data
  - The speech situation
  - The rationale behind the curation.

Corpora

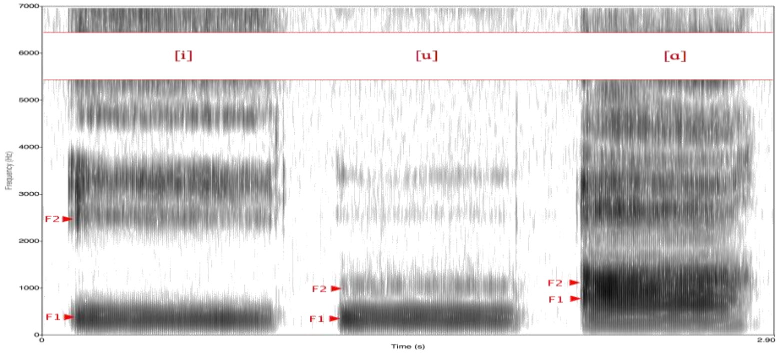
- What is a corpus?
  - A collection of text or recorded speech—typically in machine-readable form—compiled to be representative of a particular kind of language.
  - Used as a starting point for quantitative, empirical linguistic research or language description
- Corpus characteristics:
  - Raw
  - Tagged/Annotated (i.e. Penn Treebank)
    - Automatic tagging (issues?)
    - Human annotation (issues?)
    - Hybrid approach: automated system refers cases it is unsure of to human annotation

Annotation

- In Project 2 (unigram tallies), we gathered statistics from a raw corpus
- In Project 4 (DNA targets), we are searching a raw corpus, a basic form of Information Extraction (IE)
- In Project 1 (PTB constituents), we gathered statistics from an annotated corpus
- Annotation adds value to a corpus by increasing the number of statistical dimensions we can attempt to correlate. This applies to:
  - automatic methods (machine learning)
  - rule-based (analytical methods)

Annotating audio corpora

- Phonetic transcription
- Phonemic transcription
- Text transcription
- Speaker ascription (discourse/dialogue)
- Formant analysis (vowel resonances)
- Prosody
- Start/stop timings
- FFT analysis (frequencies)
- Gesture correlation



PRAAT is an amazing free tool for phonetic analysis of human speech
http://www.fon.hum.uva.nl/praat/

Annotating text corpora

- Sentence identification (sentence breaking)
- Word identification (tokenization, word-breaking)
- Part-of-speech (POS)
  - http://cst.dk/online/pos_tagger/uk/index.html
- Named entities (NER)
  - http://alias-i.com/lingpipe/web/demo-ne.html
- Anaphora resolution
- Semantic analysis
  - http://redwoods.stanford.edu/

Example: annotating information structure

- Linguistic *information structure* is concerned with the management of elaboration between speaker and hearer in discourse
- This sub-field introduces the notions of:
  - topic (what a proposition is "about")
  - focus (the new information that is being asserted about the topic)

Example: annotating information structure

topic

(I hope Sandy likes the iPod Kim gave her.)
"It's a [BOOK$_F$] that Kim gave Sandy." (not an iPod)
(correctional focus)

(What's in the bag?)
"It's [a book that Kim gave Sandy$_F$]."
(argument focus)

(What did Allie do?)
"She [went to the cricket match$_F$]."
(predicate focus)

Part-of-speech (POS) tagging

- Automatic POS-tagging of a corpora is a fundamental task in computational linguistics
- This task is a prerequisite for building many types of statistical models

Corpus priors **POS n-grams** **lemma n-grams**

DT	NN	VBD	RB	IN	DT	NN	,	CC	DT	VBG	NNS	VBD	DT	NN	VBD
the	cold	passed	reluctantly	from	the	earth	,	and	the	retiring	fogs	revealed	an	army	stretched

IN	IN	DT	NNS	,	VBG	.	IN	DT	NN	VBN	IN	JJ	TO	VB	,	DT	NN	VBN
out	on	the	hills	,	resting	.	as	the	landscape	changed	from	brown	to	green	,	the	army	awakened

,	CC	VBD	TO	VB	IN	NN	IN	DT	NN	IN	NNS	.
,	and	began	to	tremble	with	eagerness	at	the	noise	of	rumors	.

POS tagging

Objective: given sentence

$$S = (w_0, w_1, \dots w_n),$$

determine tags

$$T = (t_0, t_1, \dots t_n).$$

DT	NN	VBD	RB	IN	DT	NN	,	CC	DT	VBG	NNS	VBD	DT	NN	VBD
the	cold	passed	reluctantly	from	the	earth	,	and	the	retiring	fogs	revealed	an	army	stretched

IN	IN	DT	NNS	,	VBG	.	IN	DT	NN	VBN	IN	JJ	TO	VB	,	DT	NN	VBN
out	on	the	hills	,	resting	.	as	the	landscape	changed	from	brown	to	green	,	the	army	awakened

,	CC	VBD	TO	VB	IN	NN	IN	DT	NN	IN	NNS	.
,	and	began	to	tremble	with	eagerness	at	the	noise	of	rumors	.

Human annotation

- How to proceed with human tagging is obvious

If cost is no object, is this certainly the “best” thing to do?

- Not necessarily. It is very hard to get consistent results
  - Clear standards and procedures must be defined
  - Empirical quality control sampling is advisable
  - Automatic methods are likely to be more consistent

Automatic tagging does not "pollute" the corpus

POS Tagging

annotation

raw

DT	NN	VBD	RB	IN	DT	NN	,	CC	DT	VBG	NNS	VBD	DT	NN	VBD
the	cold	passed	reluctantly	from	the	earth	,	and	the	retiring	fogs	revealed	an	army	stretched

IN	IN	DT	NNS	,	VBG	.	IN	DT	NN	VBN	IN	JJ	TO	VB	,	DT	NN	VBN
out	on	the	hills	,	resting	.	as	the	landscape	changed	from	brown	to	green	,	the	army	awakened

,	CC	VBD	TO	VB	IN	NN	IN	DT	NN	IN	NNS	.
,	and	began	to	tremble	with	eagerness	at	the	noise	of	rumors	.

Machine Translation

Document Summarization

Information Extraction

Deep (semantic) Processing

Question Answering

Note on notation

"probability that a noun follows a determiner"

- Before, when we looked at n-gram probabilities such as $P(\mathrm{NN}|\mathrm{DT})$, the conditional "given" symbol '|' meant "reading left-to-right," more precisely:

  $$P(t_i \mid t_{i-1})$$

- We can also refer to the probability of a tag given its word, $P(t_i \mid w_i)$ or the reverse, $P(w_i \mid t_i)$.

- So we need to pay careful attention to the variables and the subscripts

Deciphering subscript-less notation

$P(NN)$	Probability of a noun (versus all POS unigrams)
$P(NN\|DT)$	Probability that a noun follows a determiner
$P(NN\ DT)$ sometimes you'll see: $P(NN, DT)$	Probability of the POS bigram "NN DT" (versus all POS bigrams)
$P(\text{the}\|DT)$	Probability of a determiner being the word "the"
$P(DT\|\text{the})$	(i.e.) Probability of tagging the word "the" as a determiner
$P(NN\|DT\ JJ)$	Probability of a noun following the bigram "DT JJ"

$P(t_{i-1}, t_i)$

$P(t_i|t_{i-2}, t_{i-1})$

$P(w_i|t_i)$

$P(t_i|w_i)$

The $P(NN, DT)$ notation (with a comma) is confusing, because it implies joint probability, which is normally *commutative*, but we have an ordering constraint such that $P(NN\ DT) \neq P(DT\ NN)$. This problem is avoided by using subscripts in $P(t_{i-1}, t_i)$, where the comma is ok. In either case, terms should always be written in sentence order.

$P(t|w)$

- This type of notation can refer to either:
  - a corpus prior, that is the *observed* (counted) probability of tag t in the corpus, restricted by word w.

    i.e. appearing on the **right** side of Bayes' theorem

  - a model term, which is typically used as part of the model's maximized objective function.

    i.e. appearing on the **left** side of Bayes' theorem

  - What's a *maximized objective function*? First, let's define a handy math notation helper, called **argmax**…

$$\mathrm{argmax}_x f(x)$$

The result of this expression is:

the value (or values) x such that $f(x)$ is maximized.

$\boxed{\mathrm{argmin}}$ works in a similar way

- We don't care about the actual evaluation result of the function $f(x)$. It is discarded.
- You will see this notation often in computational linguistics

ArgMax<TSrc,TArg>

```
public static TSrc ArgMax<TSrc, TArg>(this IEnumerable<TSrc> seq, Converter<TSrc, TArg> objective)
    where TArg : IComparable<TArg>
{
    IEnumerator<TSrc> e = seq.GetEnumerator();
    if (!e.MoveNext())
        throw new InvalidOperationException("Sequence has no elements.");

    TSrc t = e.Current;
    if (e.MoveNext())
    {
        TArg v, max_val = objective(t);
        do
        {
            TSrc t_try = e.Current;
            v = objective(t_try);
            if (v.CompareTo(max_val) > 0)
            {
                t = t_try;
                max_val = v;
            }
        }
        while (e.MoveNext());
    }
    return t;
}
```

example

$$f(x) = (x-3)^2$$

$$\mathrm{argmin}_x f(x) = 3$$

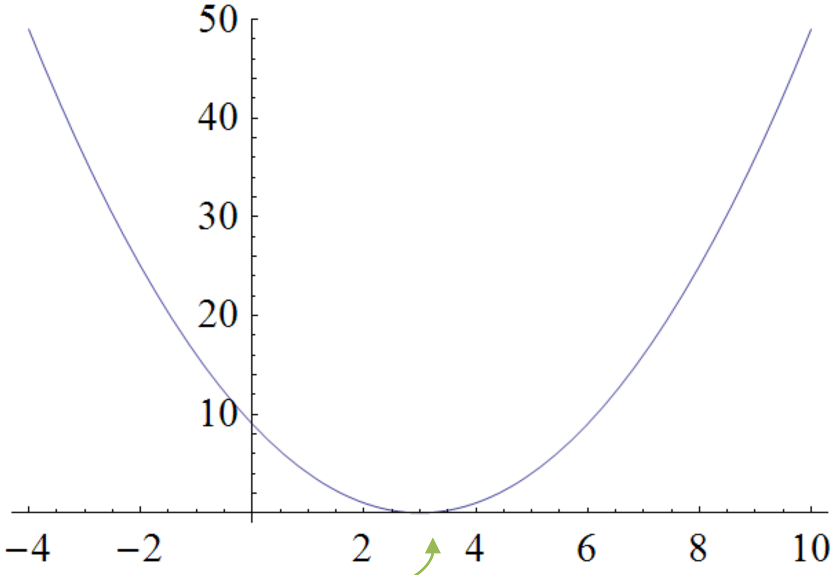
or

$$\mathrm{argmin}_x (x-3)^2 = 3$$

In[3]:=

```
Plot[(x - 3)^2, {x, -4, 10}]
```

Out[3]=



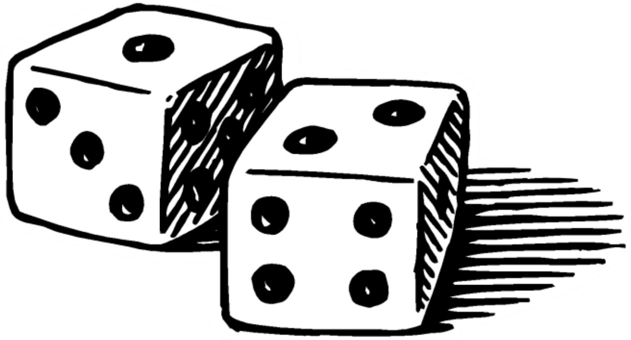
The value of $f(x)$ at 3 is 0, but argmin doesn't care about that, so long as it's the minimum value

argmax example #1

$$X = \{ \text{ the total showing on two fair dice } \}$$

What is the value of:

$$\text{argmax}_x P(X = x)$$



7

argmax example #2

$$X = \{\text{ a sample of English language text }\}$$
$$f(x) = Count(x), x \in \{ '\text{a}', '\text{b}', '\text{c}', \ldots '\text{z}' \}$$

What is the value of:

$$\text{argmax}_x f(x)$$

‘e’

Tagging objective function

Predict a sequence of tags $\hat{t}$ based on the probability of tags and words $P(t_i|w_i)$. Given sentence

"t-hat"

$$S = (w_0, w_1, \ldots w_n),$$
$$\hat{t} = \mathrm{argmax}_{t_i} P(t_i|\, w_i).$$

"$\hat{t}$ is the best sequence of tags that match a tag t_i to its word w_i."

This material is also covered in section 5.5 (p.139) of Jurafsky & Martin, 2nd ed.

Simplistic tagger

$$\mathrm{S} = (w_0, w_1, \ldots w_n)$$
$$\hat{t} = \mathrm{argmax}_{t_i} P(t_i |\, w_i)$$

This is surely the function we want to maximize, but it's not clear how to calculate the probabilities $P(t|w)$.

Simplistic tagger: Why don't we use probabilities calculated from a corpus ?

– as in Assignment 3

Simplistic tagger

DT	NN	VBD	RB	IN	DT	NN	,	CC	DT	VBG	NNS	VBD	DT	NN	VBD
the	cold	passed	reluctantly	from	the	earth	,	and	the	retiring	fogs	revealed	an	army	stretched

IN	IN	DT	NNS	,	VBG	.	IN	DT	NN	VBN	IN	JJ	TO	VB	,	DT	NN	VBN
out	on	the	hills	,	resting	.	as	the	landscape	changed	from	brown	to	green	,	the	army	awakened

,	CC	VBD	TO	VB	IN	NN	IN	DT	NN	IN	NNS	.
,	and	began	to	tremble	with	eagerness	at	the	noise	of	rumors	.

$$\text{argmax}_t P(t|\text{the}) = \texttt{DT}$$ ✓

$$\text{argmax}_t P(t|\text{cold}) = \texttt{JJ}$$ ✗

How well does the simplistic tagger work?

- Such a POS tagger is not really usable

Most probable POS tag: JJ
Correct tag: NN

Most probable POS tag: VBG
Correct tag: JJ

DT		VBD	RB	IN	DT	NN	,	CC	DT		NNS	VBD	DT	NN	VBD
the	cold	passed	reluctantly	from	the	earth	,	and	the	retiring	fogs	revealed	an	army	stretched

IN	IN	DT	NNS	,	VBG	.	IN	DT	NN	VBN	IN	JJ	TO	JJ	,	DT	NN	VBN
out	on	the	hills	,	resting	.	as	the	landscape	changed	from	brown	to	green	,	the	army	awakened

,	CC	VBD	TO	VB	IN	NN	IN	DT	NN	IN	NNS	.
,	and	began	to	tremble	with	eagerness	at	the	noise	of	rumors	.

Use Bayes Theorem

Of course, you have this memorized

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Remember, this was our objective function

$$\hat{t} = \operatorname{argmax}_t P(t_i|\, w_i)$$

$$\hat{t} = \operatorname{argmax}_t \frac{P(w_i|t_i)P(t_i)}{P(w_i)}$$

This is one of the most important slides of this entire class

For each evaluated value of i, $P(w_i)$ will be the same. We can cancel it.

$$\hat{t} = \text{argmax}_t \frac{P(w_i|t_i)P(t_i)}{\cancel{P(w_i)}}$$

$$\hat{t} = \text{argmax}_t P(w_i|t_i)P(t_i)$$

$$\hat{t} = \text{argmax}_t P(w_i|t_i)P(t_i)$$

"likelihood"

"prior"

"We compute the most probable tag sequence… by multiplying the **likelihood** and the **prior probability** for each tag sequence and choosing the tag sequence for which this product is greatest.

"Unfortunately, this is still too hard to compute directly…"

Jurafsky & Martin (paraphrase) p.140

We still need to make some assumptions.

we'll come back to this part later

$$\hat{t} = \mathrm{argmax}_t P(w_i|t_i) P(t_i)$$

Assumption 1: If we want to use corpus probabilities to estimate $P(w_i|t_i)$, we need to formally note that we're assuming it.

$$P'(w_i|t_i) = \prod_i P(w_i|t_i t_{i-1} t_{i-2} \dots)$$

$$\approx \prod_i P(w_i|t_i)$$

“The only POS tag a word depends on is its own.”

Is this true?

Any progress?

- So wait: if we're assuming the only *POS tag* a *word* depends on is its own, how is this going to be better than the simplistic tagger from before, which assumed that the only *word* a *POS tag* depends on is its own?
- In other words, Why is $P(w|t)$ going to work better than $P(t|w)$?
- Hint: $|\Omega|$
- Hint: $|\mathrm{T}| \ll |\mathrm{W}|$

Answer: because there are a lot more distinct words than tags, conditioning on *tags* rather than *words* increases the resolution of the corpus measurements

example

$$P(\text{cold}|\texttt{NN}) = .00002$$
$$P(\text{cold}|\texttt{JJ}) = .00040$$

$$P(\texttt{JJ}|\text{cold}) = .97$$
$$P(\texttt{NN}|\text{cold}) = .03$$

This value will drown out our calculation and we'd never tag "cold" as a noun!

$$\hat{t} = \text{argmax}_t P'(w_i|t_i)P(t_i)$$

Assumption 2: The only tags that a tag t_i depends on are the n previous tags, $t_{i-n-1} \ldots t_{i-1}$. For example, in a POS bigram model:

$$P'(t_i) = \prod_i P(t_i|t_{i-1}t_{i-2}t_{i-3} \ldots)$$

$$\approx \prod_i P(t_i|t_{i-1})$$

This is known as the bigram assumption: "The only POS tag(s) a POS tag depends on are the ones immediately preceding it."

Putting it together

$$\hat{t} = \text{argmax}_t P(w_i|t_i)P(t_i)$$

$$P'(w_i|t_i) \approx \prod_i P(w_i|t_i)$$

$$P'(t_i) \approx \prod_i P(t_i|t_{i-1})$$

$$\hat{t} = \text{argmax}_t \prod_i P(w_i|t_i) \prod_i P(t_i|t_{i-1})$$

$$\hat{t} = \text{argmax}_t \prod_i P(w_i|t_i)P(t_i|t_{i-1})$$

Reminder: estimating $P(w_i|t_i)$ from a corpus

Definition of conditional probability

$$P(A|B) = \frac{P(A,B)}{P(B)}$$

$$P(A|B) = \frac{\frac{\text{count}(A,B)}{\cancel{|\Omega|}}}{\frac{\text{count}(B)}{\cancel{|\Omega|}}}$$

word likelihood

$$P(w_i|t_i) = \frac{\text{count}(w_i, t_i)}{\text{count}(t_i)}$$

Reminder: estimating $P(t_i|t_{i-1})$ from a corpus

Definition of conditional probability

$$P(A|B) = \frac{P(A,B)}{P(B)}$$

$$P(A|B) = \frac{\frac{\text{count}(A,B)}{\cancel{|\Omega|}}}{\frac{\text{count}(B)}{\cancel{|\Omega|}}}$$

$$P(t_i|t_{i-1}) = \frac{\text{count}(t_{i-1},t_i)}{\text{count}(t_{i-1})}$$

POS tagging objective function

$$\hat{t} = \operatorname{argmax}_t \prod_i \left(\frac{\text{count}(w_i, t_i)}{\text{count}(t_i)} \times \frac{\text{count}(t_{i-1}, t_i)}{\text{count}(t_{i-1})} \right)$$

Best POS tag sequence

How often does word w_i occur with tag t_i in the corpus?

How often does t_i follow t_{i-1} in the the corpus?

This might seem a little backwards (especially if you aren't familiar with Bayes' theorem). We're trying to find the best *tag sequence*, but we're using $P(w|t)$, which seems to be predicting *words*.

This compares: "If we are expecting an **adjective** (based on the tag sequence), how likely is it that the adjective will be 'cold?'" ***versus*** "If we are expecting a **noun**, how likely is it that the noun will be 'cold?'"

DT		VBD	RB	IN	DT	NN	,
the	cold	passed	reluctantly	from	the	earth	,

“If we are expecting an **adjective**, how likely is it that the adjective will be ‘cold?’” (high) **WEIGHTED BY** our chance of seeing the sequence **DT JJ** (medium)

versus

“If we are expecting a **noun**, how likely is it that the noun will be ‘cold?’” (medium) **WEIGHTED BY** our chance of seeing the sequence **DT NN** (very high)

THE WINNER: NN