LING 473: Day 1

START THE RECORDING

LING 473

- Instruction
- Instructor
- Prerequisites
- UW Degree and Certificate Programs in Comp. Ling.
- Program Faculty
- Getting Set Up
- Class Tools
- Software
- Programming Languages
- Assignments
- Grading

Instruction

Meeting Time: Tuesday & Thursday, 4:40 – 6:20 pm
July 20 – September 7, 2017

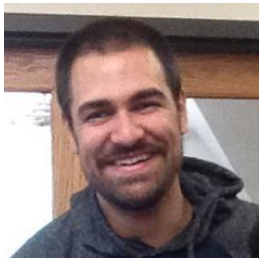
Location: Denny 212

Online via AdobeConnect

Afterwards via recording

This option does not allow for participation, so please use sparingly.

Instructor



David Inman
davinman@uw.edu
PhC in Linguistics at UW
BA Linguistics, BS Computer Science from UT Austin

Research

Nuuchahnulth morphosyntax, computational syntax

Past

Software Tester at Microsoft

LING 473 Prerequisites

CSE 326 or 373: Data Structures

Abstract data types and their implementations as data structures. Efficient use of algorithms employing these data structures; asymptotic analyses. Dictionaries: balanced search trees, hashing. Priority queues: heaps. Disjoint sets with union, find. Graph algorithms: shortest path, minimum spanning tree, topological sort, search. Sorting.

STAT 390 or 391: Probability and Statistics for Computer Science

Fundamentals of probability and statistics from the perspective of the computer scientist. Random variables, distributions and densities, conditional probability, independence. Maximum likelihood, density estimation, Markov chains, classification. Applications in computer science.

(or equivalents)

LING 473 is required for the NLT Certificate Program

CLMS Program: determination by placement test

UW NLT: Certificate in Natural Language Technology

- http://www.pce.uw.edu/certificates/natural-language-technology.html
- 11 credits
- LING 473
- LING 570

  Techniques and algorithms for associating relatively surface-level structures and information with natural language corpora, including POS tagging, morphological analysis, preprocessing/segmentation named-entity recognition, chunk parsing, and word-sense disambiguation. Examines linguistic resources that can be leveraged for these tasks (e.g., WordNet).
- LING 571

  Algorithms for associating deep or elaborated linguistic structures with naturally occurring linguistic data (parsing/semantics/discourse), and to produce natural language strings from input semantic representations (generation).
- Consider GNM status if you think you might want to convert to CLMS

  https://www.pce.uw.edu/help/applying/graduate-nonmatriculated-status

CLMS: Professional M.S. in Computational Linguistics

- http://www.compling.washington.edu/compling/
- 43 credits
- Ling 450 Phonetics

  Be sure to take 550 instead if you think you might convert to UW Ph.D.
- Ling 566 Syntax for computational linguistics

  Introduction to syntactic analysis and concepts with emphasis on the formally precise encoding in linguistic hypotheses and the design of grammars that can be scaled to practical applications. Coursework progressively builds up a consistent grammar for a fragment of English, while also considering data and phenomena from other languages
- Ling 570, 571, 572, 573 (core sequence)
- 3 electives (1 comp. ling., 1 ling., and 1 related area)
- Thesis or internship/report option (10 credits)

UW Ph.D in Computational Linguistics

- https://linguistics.washington.edu/phd-linguistics#compling
- Full Ph.D. in computational linguistics at UW

UW Computational Linguistics Faculty

- Emily Bender, Faculty Director
  - computational syntax, computational semantics, linguistic typology
- Fei Xia
  - statistical and hybrid methods, grammar, machine learning, automatic document understanding
- Gina-Anne Levow
  - speech and intonation processing, prosody, human-computer dialogue

UW Linguistics

- https://linguistics.washington.edu/
- Richard Wright, Department Chair
- Michael Furr, Administrator
- Catherine Carrera, Program Coordinator
  `linguw@uw.edu`
- Joyce Parvi, Academic Counselor
  `phoneme@uw.edu`
- Brandon Graves, Systems Administrator
  `bmgraves@uw.edu`

Getting Set Up

- UW NetID
- Husky Card
- Computational Linguistics Cluster Account
- Keypad access to “Treehouse” lab in Guggenheim (local students)
- Recommended textbooks

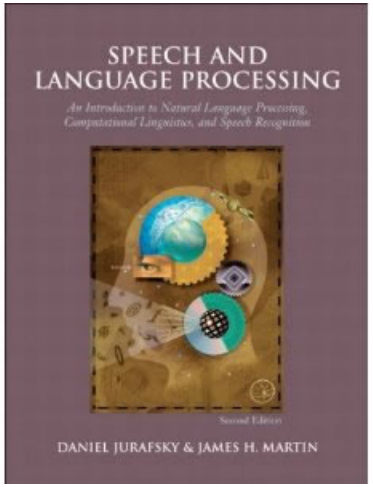
Daniel Jurafsky and James H. Martin. (2008) *Speech and Language Processing (2nd edition)*. New Jersey: Prentice-Hall.

Christopher D. Manning and Hinrich Schütze. (1999) *Foundations of Statistical Natural Language Processing*. Cambridge: MIT Press.

Both of these texts are required for CLMS and NLT students when taking Ling 570/571. If doing so in the fall, purchase now.

Getting Set Up

- Daniel Jurafsky and James H. Martin. (2008) *Speech and Language Processing (2nd edition)*. New Jersey: Prentice-Hall.



Getting Set Up

- Get a Patas account - today!
  - https://vervet.ling.washington.edu/db/accountrequest-form.php
  - You can also get here from the Treehouse Wiki (link on the course website)

Class Resources

- Catalyst Tools
  - GoPost
    - https://catalyst.uw.edu/gopost/board/davinman/44098/
  - CollectIt
    - https://catalyst.uw.edu/collectit/dropbox/davinman/40705
  - GradeBook
    - https://catalyst.uw.edu/gradebook/davinman/102032
- Treehouse wiki
  - http://depts.washington.edu/uwcl/twiki/bin/view.cgi/Main/WebHome
- CLMS Survival Guide:
  - http://depts.washington.edu/uwcl/twiki/bin/view.cgi/Main/CLMASurvivalGuide
- Computational Linguistics Linux cluster
  - Interactive head nodes: patas.ling.washington.edu; dryas.ling.washington.edu
  - About 50 processors on ~20 machines for batch jobs
  - Files for certain assignments may be found in /opt/dropbox
  - Licensed corpora database: https://vervet.ling.washington.edu/db/index.php
  - Condor batch submission system; Network File System

Software

- LaTeX
  - Consider if you're continuing in MA or Ph.D
  - Can create PDF files
  - On Windows: Install MiKTeX and Texmaker
  - On Mac: MacTex + TexWorks, TexPad ($$)
- Ability to create PDF files
  - Adobe Acrobat (academic pricing at UW Bookstore)
  - Open Office?
  - PDF creator
  - http://depts.washington.edu/uwcl/twiki/bin/view.cgi/Main/WordToPDF
- SSH Terminal Programs available from UWICK
  - http://www.washington.edu/uware/uwick/
  - WinSCP
  - SSH Tectia
  - Tera Term
  - Putty

Programming Languages

- You can use any programming language so long as it's available on Patas.
  - C/C++
  - C# (mono)
  - Java
  - Python
  - Perl

I am most familiar with C#, Java, and Python. Other languages are fine to use but my ability to debug them may vary.

This isn't a programming class. We will not cover how to create, edit, run or debug programs.

Written Assignments

- Probability and statistics problems
- Preferred submission format: PDF
- Please don't write out by hand
  - LaTeX or an equation tool layout (Word can do this!)

Programming Projects

- Code must run on UW Compling Cluster
  - No credit if it does not. This is because some projects may reference licensed corpora which you may not copy.
  - I won't spend (much) time figuring out why your code doesn't run.
  - You can develop on a home machine, but make sure you test thoroughly on the cluster.
- Please follow instructions
- TAR and submit to CollectIt:
  - all required source code and files (except public files)
  - point to public files where possible
  - output file captured from stdout
  - prose description (write-up) of your work

Grading

- 3(?) Written Assignments: 30%
- 5(?) Program Projects: 55%
- Paper Review: 10%
- Participation: 5%

- Policies:
  - http://courses.washington.edu/ling473/syllabus.html
  - No late work! (But there is extra credit)

Computational Linguistics

- A quick survey
- Why is comp ling hard?
- Linguistic structure
- Analytical and statistical approaches
- The Penn Treebank
- Assignment 1

What is Computational Linguistics?

Using quantitative, computational techniques for the analysis and processing of human (natural) language.

- Inherently involves multiple fields
  - Computer science
  - Linguistics
  - Mathematics
  - Electrical engineering

Linguistics

- Within linguistics, many traditional fields can (and do) use computational methods in their research:
  - Phonetics
  - Phonology
  - Morphology
  - Syntax
  - Lexical Semantics
  - Compositional Semantics
  - Pragmatics
  - Discourse Analysis
  - Information Structure
  - Typology

Computer Science

- Fundamental CS data structures are relevant, incl Big-O notation
  - Comp ling is one of the few areas I know where it is relatively easy to accidentally write a program that runs for days instead of minutes
- Text processing (at large scale), regexes, encoding, data conversion
- Databases, especially for high throughput
- Special techniques: string hashes, tries, dynamic programming

Mathematics

- Probability and statistics
  - Modeling the occurrence(s) of events
  - Application of probabilistic models
- Set Theory
  - Intersection, Union, Exclusion
- Logic
  - Boolean logic
  - First order logic (predicate calculus)
  - Markov (probabilistic) logic, entailment
- Information Theory
  - Entropy
- Etc
  - Kernel functions (for Support Vector Machines…)
  - Matrix decomposition (i.e. Singular Value Decomposition…)
  - Graphs (for unification grammar, semantic relations)

Ambitions and Jargon

- Natural Language Processing (NLP)
  - Processing human language in some capacity with a computer
  - Can be clever or not
- Natural Language Understanding (NLU)
  - Processing human language in a way that the computer "understands" what the language is saying
- Natural Language Technology (NLT)
  - Generic

Natural Language Understanding

- What is "understanding"?
  - How can you measure it?
  - Do we have a theory of what human beings know when they have an understanding of something?
- "Understanding" is in some way linked to the real world
  - When you know something, you can apply it to novel situations in the real world.
  - Computational linguistics is very poorly (in the best case) linked up to real-world models.
  - True real-world models (that extend beyond "circle" vs "triangle") don't really exist right now.

NLP Subfields

- IE: Information Extraction
- IR: Information Retrieval
- MT: Machine Translation
- Document Summarization
- QA: Question-answering
- ASR: Automatic Speech Recognition
- NLG: Natural Language Generation
- CALL: Computer Assisted Language Learning
- Alternative Input Methods

NLP Subtasks

- Tokenization (speech stream or string stream into words)
- Sentence breaking
- Morphological analysis
  - Part-of-speech (POS)
  - Stemming
- Named Entity Recognition (NER)
- Word Sense Disambiguation
- Anaphora and (co)reference resolution
- Parsing and generation
- Dialog and discourse analysis
- Clustering/Classification
- Treebanking and corpora curation

NLP Approaches

- Analytical (rule-based)
  - Compose a set of rules that govern the application
    - (from data, from human linguistic understanding)
  - Implement rules
  - Evaluate
- Stochastic or Statistical
  - Create a model that alters performance based on input/training data
  - Train the model on inputted data
  - Use the model to evaluate/make predictions on new data

Rule-Based NLP

- Human language works on a set of rules, operating in human brains. Let's figure out these rules and write them down.

- Where is this going to have trouble?

1. No one fully knows how language works in the human brain.
2. What kinds of rules should we use? Which linguistic theory?
3. Can these rules work on hardware the way they do on wetware?
4. How will we know if we're on the right track?

Machine Translation

- The first goal of NLP was machine translation.

> *When I look at an article in Russian, I say: "This is really written in English, but it has been coded in some strange symbols."*
>
> Weaver 1947

Machine Translation

- 19 years later…

> There has been no machine translation of general scientific text, and none is in immediate prospect… After 8 years of work, [the project] had to resort to post-editing [which] took slightly longer to do and was more expensive than conventional human translation."
>
> ALPAC Report 1966

Machine Translation

- 41 years later…

> *Progress on combining rule-based and data-driven approaches to MT will depend on a sustained stream of state-of-the-art, MT-oriented linguistics research... Despite frequent cycles of overly high hopes and subsequent disillusionment, [MT] is the type of application that may demand knowledge-heavy, 'deep' approaches to NLP for its ultimate, long-term success.*
>
> *Oepen et al. 2007*

Statistical NLP

- We will have a model self-learn from inputted language data.

- Where is this going to have trouble?

1. What is the basic model? What are the assumptions of the model?
2. How does the model know when to give up? (bad input)
3. What data will you use to train it?
4. Can it perform on data that is related to but different from the input?

Notational Conventions

- Sentences that are ungrammatical are marked with an asterisk.
  - **This example sentence of ungrammatical is.*
- Sentences that are marginal are marked with a question mark.
  - *?A student emailed me yesterday who was almost unable to enroll.*
- Grammatical sentences can be meaningless (sometimes marked with a pound sign).
  - *#Colorless green ideas sleep furiously*
- Grammatical sentences which don't convey the intended meaning or are semantically anomalous are marked with a pound sign.
  - *#It's raining outside but I don't believe that it's raining.*

Why is language hard?

- Let's just look at text. (But speech processing is also hard.)

I saw a man with a telescope.

  - a bunch of symbols
  - coming in order (a string) in Unicode or some other encoding
  - broken into "words"
  - forming a "sentence"
  - with some related semantic proposition

Words

Isa waman wi that eles cope.

- In English (and many other languages), the space character separates characters into words.*

*Not true in all written representations of language. Notably, Chinese does not have spaces. Even in English, words can sometimes have spaces in them: *White House*, *boa constrictor*

Words

- It may be difficult to identify a word in conventional written languages.

ผมเห็นผู้ชายกับโทรทรรศน์				
ผม	เห็น	ผู้ชาย	กับ	โทรทรรศน์
pʰǒm	hěn	pʰû: tɕʰa:j	kàp	tʰo: rá tʰát
1-sg	see	man	with	telescope
“I saw (a) man (who was) with a telescope.”				

“person-male” : 1 word or 2?

The International Phonetic Alphabet (IPA), is the standard form of phonetic transcription.

This type of formatting for linguistic examples is called “Interlinear Glossed Text,” or IGT

Word Class

- Is there a finite set of word classes that behave similarly that we can generalize over?
  - These are called parts of speech (verb, noun, determiner, etc)
  - In comp ling, often POS tag
  - Closed and open classes

Closed Word Classes

- A closed class
  - Has a limited number of members
  - Generally the language can't adopt new words into these classes
- Closed classes
  - Conjunction (and, or, but …)
  - Determiner (the, a, this, that, these, …)
  - Pronoun (he, she, it, I, we, they, …)
  - Auxiliary verb (have, do, been, …)
  - Preposition (in, of, over, …)
  - …

Open Word Classes

- An open class
  - Has a large number of members
  - Generally open to new words
  - New words may be generated by morphological processes
- Open classes
  - Noun
  - Verb
  - Adjective
  - Adverb

Sentences

the carnival. I saw the man with a

- In English (and many other languages) a symbol separates words into sentences.*

*But not all, and there still are problems with this, even in English. Sometimes periods are used in abbreviations, e.g. Mr. and Mrs., and even worse sometimes a period is both for an abbreviation and the end of a sentence.

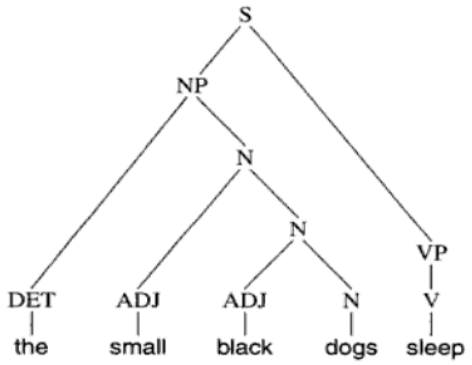
Sentences

**Man a telescope saw with I a*

- In English (and many other languages) word order is significant.

Constituents

- Can sentences be analyzed as containing sub-units of one or more words, which combine to form larger units?



(Hausser 1998)

- "Phrase structure"
- Assumption: finite set of rules and finite set of constituent types

Constituent Types

- There may be many, but over the years a useful notion of a "head" has developed
  - Most units have a sub-unit which is "the most important" in some way.
  - E.g., noun phrases have nouns inside them; verb phrases have verbs
  - This allows us to generalize over many constituents

Constituent Types

- Noun phrases (NPs)

(DET NN)	*the ostrich*
(NNP)	*Kim*
(NN NN)	*container ship*
(DET JJ NN)	*A purple lawnmower*
(DET JJ NN)	*That darn cat*

- Verb phrases (VPs)

(VB)	*tango*
(VBD NP NP)	*gave the dog a bone*
(VBD NP PP)	*gave a bone to the dog*

Syntax

The set of rules governing permissible constructions in a language.

- Syntax constrains the ways in which words may be combined to form phrases, including sentences.
- Syntax forms one part of the description, or *grammar*, of a language.

Prescriptive and Descriptive Grammars

- Prescriptive
  - Rules against certain usages. Few if any rules for what *is* allowed.
    - Don't end sentences with a preposition. *(Where are you from?)*
    - Don't use "ain't", "irregardless", etc.
- Descriptive
  - Rules characterizing what people *do* say.
  - Goal is to characterize all and only what speakers find acceptable.
  - Based on the scientific method (observation, hypothesis, falsifiability)

Prescriptive Grammar

- Prescriptive rules are artificial
  - Often reflect social status of speakers of a dialect
  - African American English tends to bear the brunt of prescriptivist rules – power differential
  - Likewise for Southern American English
  - Even phenomena common to the standard dialect can be proscribed: e.g., singular “they” (dates to at least Chaucer)

Prescriptive Grammar

- *He/his, they/their,* or something else?
  - Everyone insisted that ___ were/was fine
  - Everyone drives ___ own car
  - Everyone was happy because ___ passed the test
  - Everyone passed, didn't ___ ?

(Taken from Bender, Saw, and Wasow)

Linguistic Ambiguity

1. Lexical ambiguity

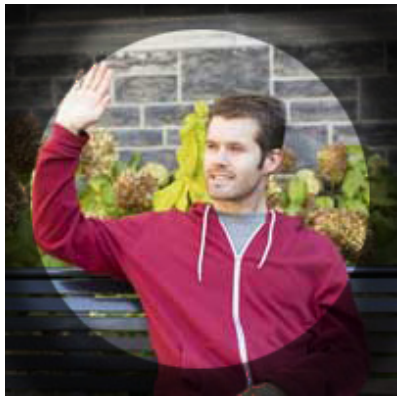
   The bank is crumbling.

?

Linguistic Ambiguity

2. Structural ambiguity.

   I saw a man with a telescope



?

Linguistic Ambiguity

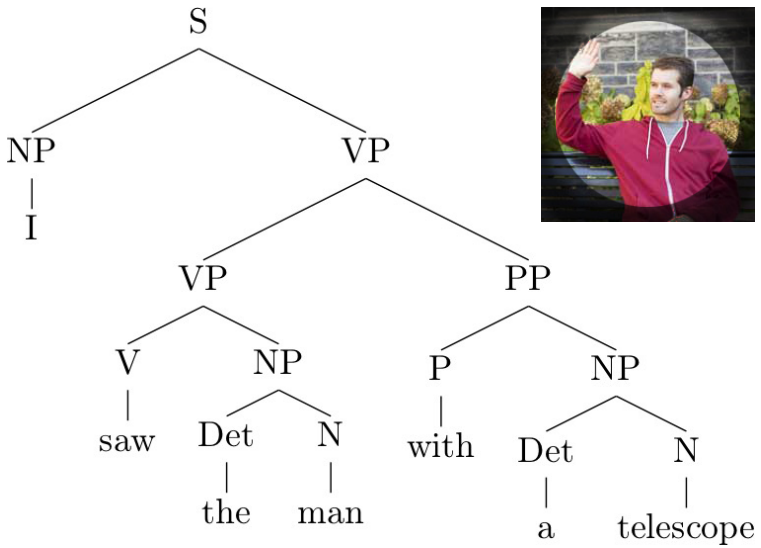
- And still more.

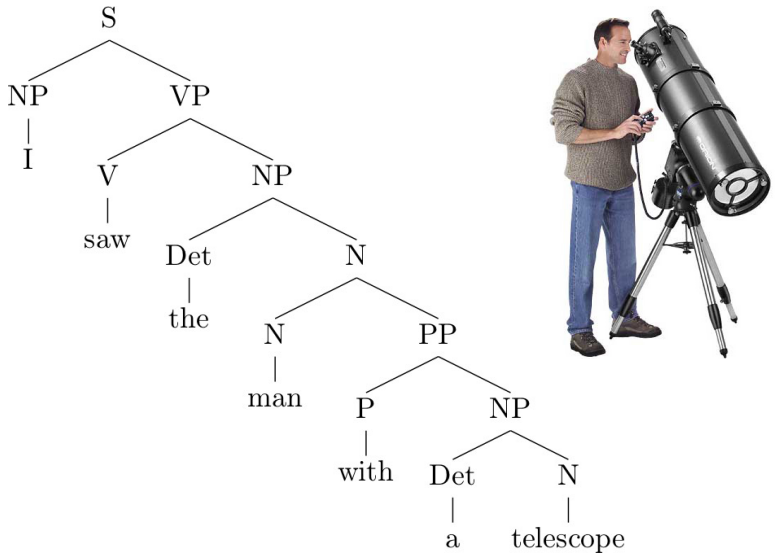
  I (often) saw a man

  with a telescope

Structural Ambiguity

- Constituents allow us to represent different interpretations of strings. (phrase structure)





Analytical NLP

- Language is extremely complex, despite our ease using it
- Ambiguity is a feature of language
- Coming up with all the rules is difficult
- Syntactic rules can easily be coded up, but there's a lot of them

Statistical NLP

- Large advances in practical applications starting in the 1990s
- Most visible example: MT
  - Google Translate
  - Bing Translate
- How? With clever math and very basic linguistic motivation.

Statistical NLP

- "Anytime a linguist leaves the group the recognition rate goes up" – Fredrick Jelinek, 1998(?)
- Computing power was increasing
- NLP accuracy was not
- Idea: Maybe the rules behind language are too hard to fully program.
- No more coding specific rules, try to find likely patterns and correlations in existing data using algorithms.

Analytical + Stochastic Models

- Hybrid systems can (if carefully designed) mitigate the ills of both approaches.
  - Hybrid machine translation (Oepen et al 2007) http://www.mt-archive.info/TMI-2007-Oepen.pdf
  - Parse ranking in the English Resource Grammar
  - Unsupervised Rule Learning (Poon and Domingos 2009) http://www.aclweb.org/anthology-new/D/D09/D09-1001.pdf

Corpus Linguistics

- A corpus is a collection of text (or data generally)
- Create the rules (by hand) or derive statistical models (algorithmically) according to a corpus of language data.
- The rules or the model are created and judged based on their performance against the corpus.

Treebanks

- A treebank is a corpus with annotated syntactic structure for each sentence.
  - Trees may be human-generated
  - or generated by precision grammar and curated
- Penn Treebank (PTB) (Marcus et al 1994)
  http://www.cis.upenn.edu/~treebank/
- Linguistic Data Consortium
  http://www.ldc.upenn.edu/

Penn Treebank

- Heavily used because it is so large
- Annotations look like:

```
( (S (NP (NNP John))
     (VP (VBZ loves)
         (NP (NNP Mary)))
   (. .)) )
```

Penn Treebank Tags

CC	Coordinating conjunction
CD	Cardinal number
DT	Determiner
EX	Existential there
FW	Foreign word
IN	Preposition or subordinating conjunction
JJ	Adjective
JJR	Adjective, comparative
JJS	Adjective, superlative
LS	List item marker
MD	Modal
NN	Noun, singular or mass
NNS	Noun, plural
NNP	Proper noun, singular
NNPS	Proper noun, plural
PDT	Predeterminer
POS	Possessive ending
PRP	Personal pronoun
PRP	Possessive pronoun
RB	Adverb
RBR	Adverb, comparative
RBS	Adverb, superlative
RP	Particle
SYM	Symbol
TO	to
UH	Interjection
VB	Verb, base form
VBD	Verb, past tense
VBG	Verb, gerund or present participle
VBN	Verb, past participle
VBP	Verb, non 3rd person singular present
VBZ	Verb, 3rd person singular present
WDT	Wh-determiner
WP	Wh-pronoun
WP	Possessive wh-pronoun
WRB	Wh-adverb

Counting

- Corpus linguistics is based on counting collections and sequences (of words or characters)
- Elements are collected together into sets
- It's important to keep track of what we are counting. We can count:
  - The number of elements in a collection.
    - This is called the "cardinality" of the collection. It is also a "token count."
  - The number of occurrences of a particular element in a collection
    - We can distinguish this by calling it a "tally." It is also a "type count."
  - The number of collections of a certain type that are found in a corpus
  - The number of collections of a certain type that could possibly be formed from another collection

Collections

- A collection of *n* elements can be ordered or unordered, distinct (all elements are unique) or non-distinct.

	Ordered	Unordered
Distinct	(i.e. MRU cache)	vocabulary, alphabet, set
Not-Distinct	sentence, string, vector, sequence, word	bag, multiset

Distinctness

- Distinct, no repetition (set, vocabulary)

  `{ banana, apple, kumquat, dragon fruit }`

  – we can count: the number of distinct elements (“cardinality”, “vocabulary size”)
- Non-distinct (bag, multiset)

  `{ banana, apple, banana, banana }`

  – we can count: the number of distinct elements
  the number of times each element appears (“tally”, “multiplicity”)
- ‘Set’ means unordered and distinct.
- We use parentheses for order-sensitive collections (tuples), braces for order-independent collections { sets }.

Ordering

- Ordered, non-distinct (string, sequence, n-tuple)

```
( I, saw, a, man, with, a, telescope )
!= ( a, a, I, man, saw, telescope, with )
```

- Unordered, non-distinct ("bag")

```
{ I, saw, a, man, with, a, telescope }
 = { a, a, I, man, saw, telescope, with }
```

- Unordered, distinct ("vocabulary")

```
{ a, I, man, saw, telescope, with }
```

- Ordered and distinct is rare. You see this in caches, where the order is recentness.

Vocabulary

The distinct set of elements which appear in some other set.

- The words used in a document or in a sentence.
- All the words of a language.
- The characters used in a word
- The English alphabet is a “vocabulary” of symbols used in writing.

Tallying

The count (number of occurrences) for each distinct element of a non-distinct set

- Each element becomes associated with its count
- Tallying creates a distinct set of tuples (item, count) from a non-distinct set (e.g., a corpus)
- Tallying is extremely common in corpus linguistics.

Tallying

- "I saw a man with a telescope"

```
(a, 2)
(I, 1)
(saw, 1)
(man, 1)
(with, 1)
(telescope, 1)
```

- Once you have a count, you can divide by the total tallies to obtain *frequencies*.

```
(a, 0.285)
(I, 0.142)
(saw, 0.142)
(man, 0.142)
(with, 0.142)
(telescope, 0.142)
```

Next Time

- Counting, Linux, using Patas, regular expressions