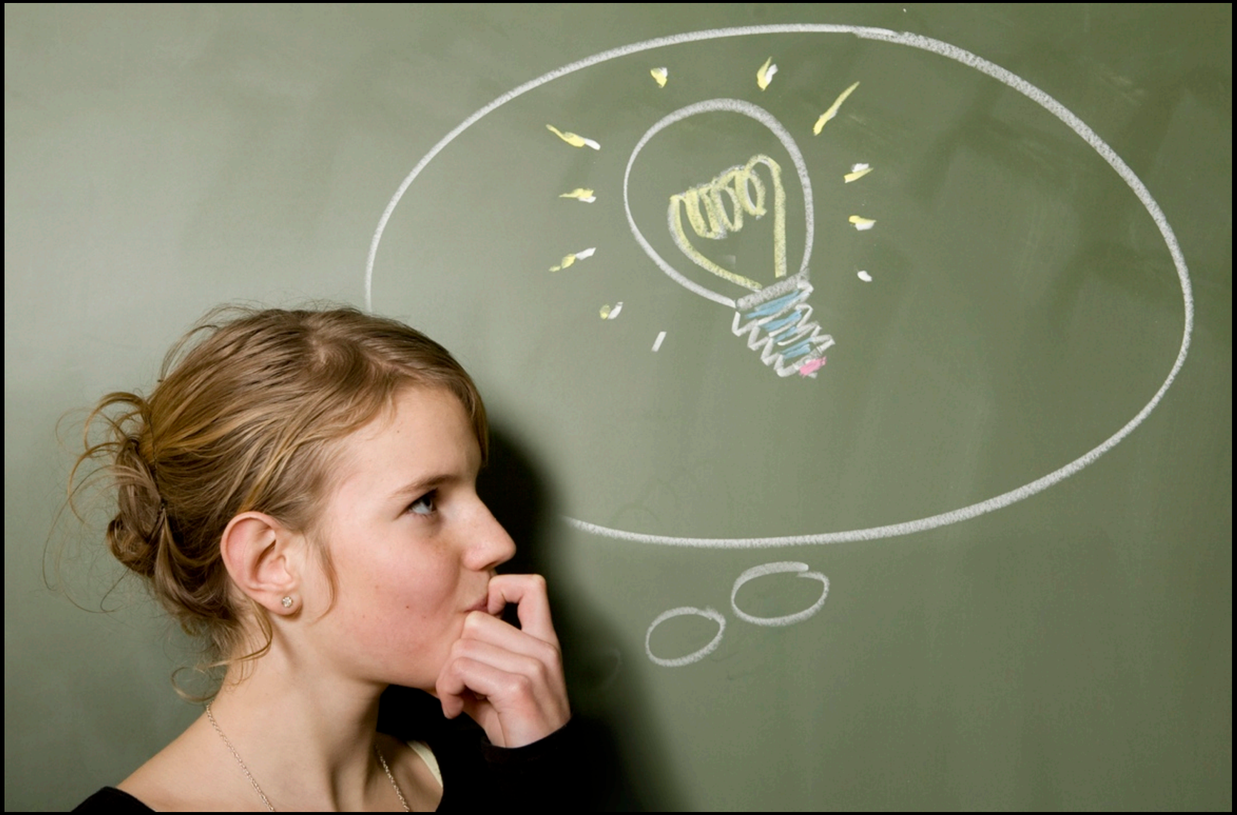
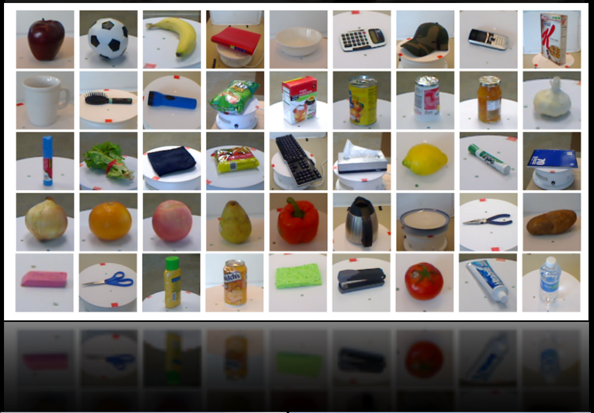
COMPUTER PROGRAMMING
ROCKS!

Comics day!

BALTIMORE COMIC-CON
America's Greatest Comic Book Convention!
Saturday and Sunday, September 8-9, 2012

In two years you will be an expert in automatic extraction of information from texts and images...



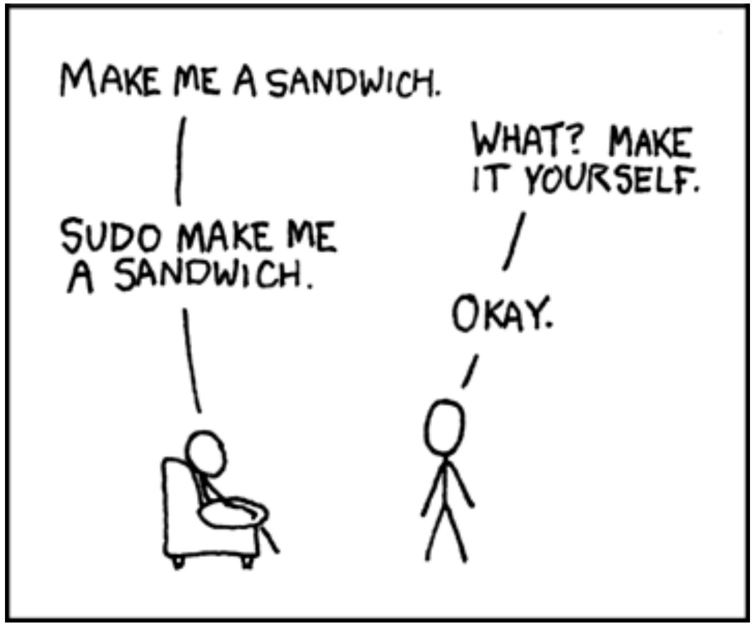


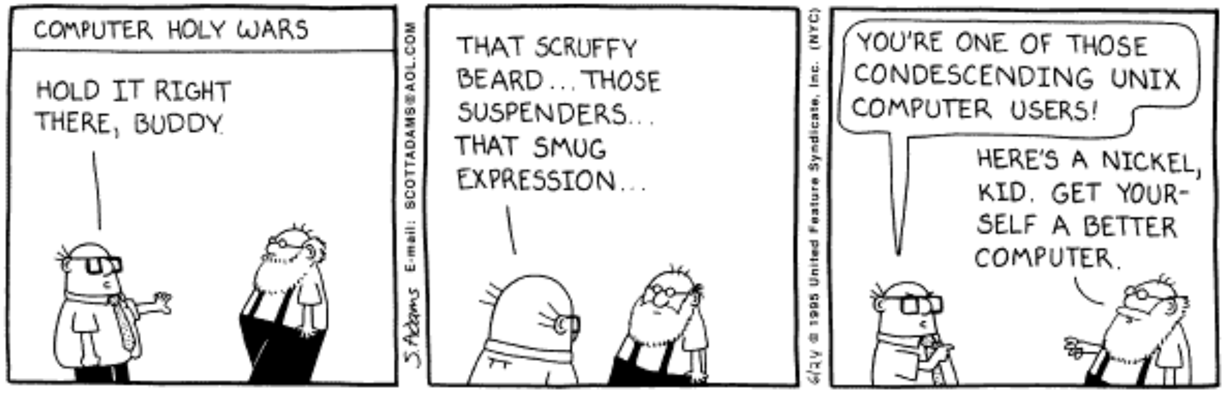
Check list for Day 1

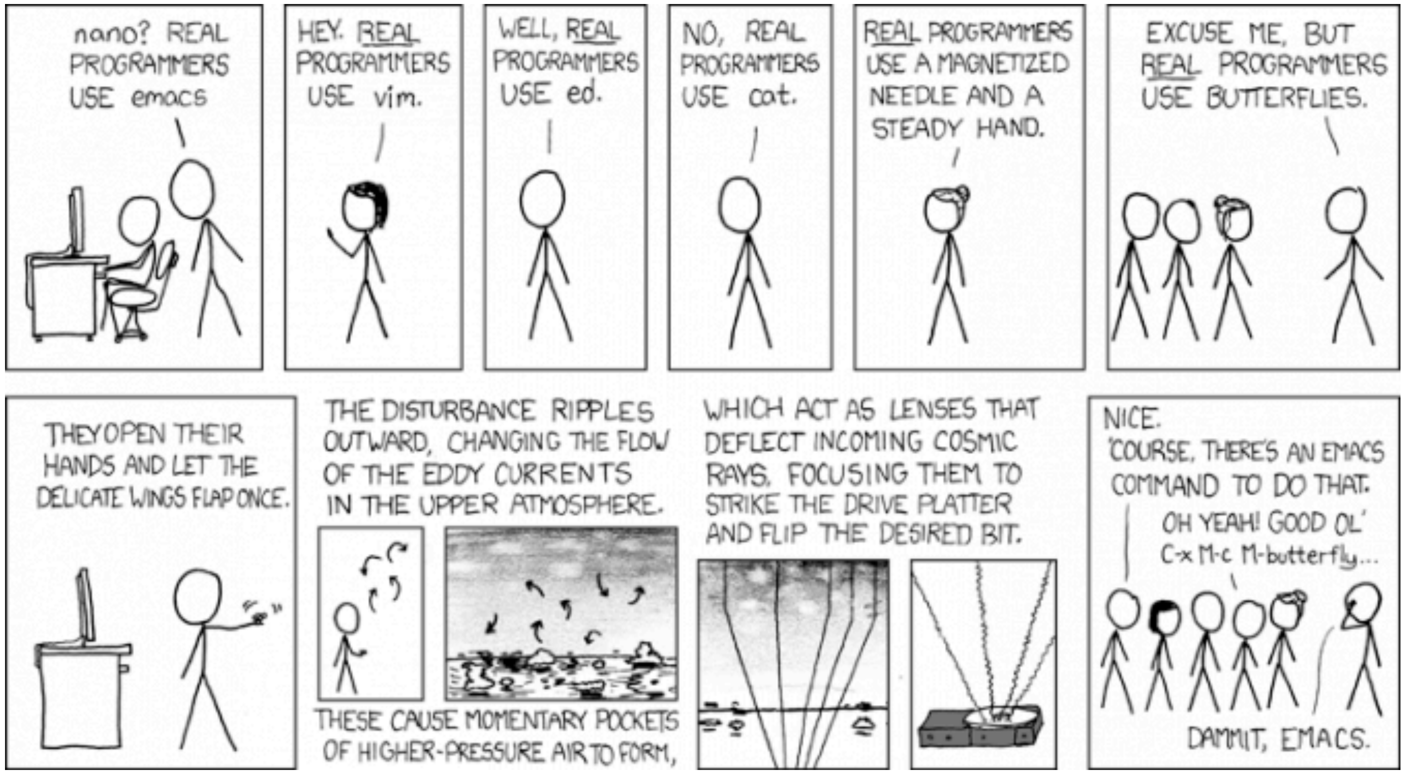
- open, readlines, strip, sys.argv (module: sys)
- str, int, float, list, dict (Try to use '+' and 'len' with different types)
- slices, .append(x), .insert(i,x), .count(x), .reverse(), max(), sum(),
- for, in, range, if, ==, else, Booleans (not, or, and),
- dictionaries (.iterkeys(),.itervalues(),.iteritems())
- <, >, +, -, *, /, **, math.log, math.sqrt (module: math)
- def, return

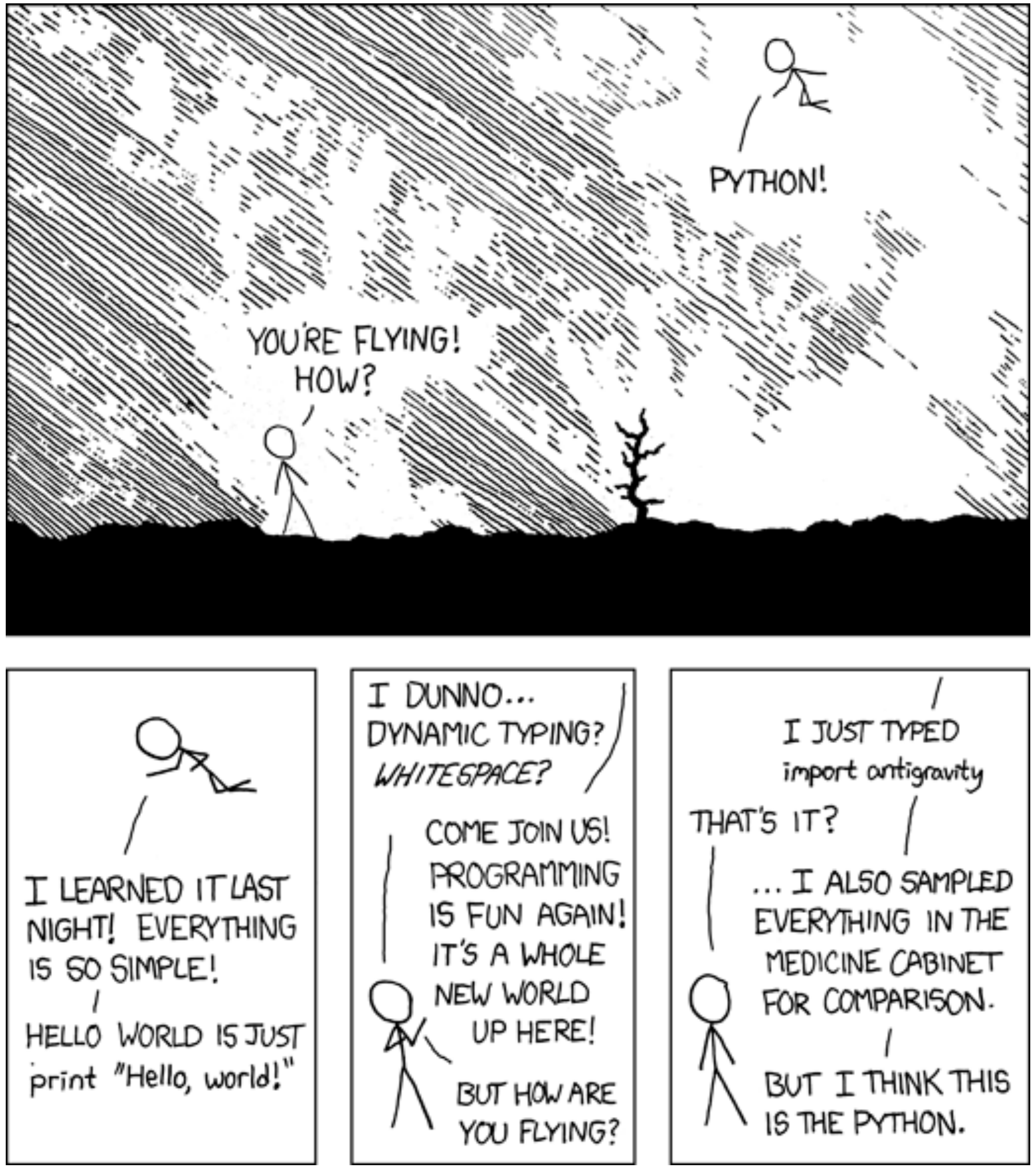
1. Toolbox: Functions. Topics: Data conversion, representing text, from folders to BOW. Modules: glob. Reading: PP9, PP10, PP13, and PP14. Exercise: Install NLTK and text2bigrams.py.
2. Toolbox: Classes. Topic: Handwritten digit recognition and BOWs in scikits. Modules: scikits. Reading: PP18 and PP19. Check this out: http://scikit-learn.org/stable/auto_examples/neighbors/plot_classification.html#example-neighbors-plot-classification-py and https://thenewcircle.com/s/post/1152/scikit-learn_machine_learning_in_python Exercise: Nearest neighbor using lists.
3. Toolbox: arrays, matrices. Topic: Nearest neighbor using scipy. Modules: numpy, scipy. Reading: http://www.engr.ucsb.edu/~shell/che210d/numpy.pdf Exercise: Implement and evaluate Rocchio on both texts and images, including error analysis (**exam**).
4. Topic: Implementing (averaged) perceptron. Reading: http://scikit-learn.org/stable/modules/linear_model.html Exercise: Implement stochastic gradient descent.
5. Topic: Evaluation against gold data. Modules: pylab. Reading: http://matplotlib.org/users/pyplot_tutorial.html https://thenewcircle.com/s/post/1133/advanced_matplotlib_tutorial_with_library_author_john_hunter Exercise: Learning curves on your own document classification dataset (**exam**).
6. Toolbox: scipy.stats, pandas. Topic: Naïve Bayes, distributional assumptions. Reading: http://www.youtube.com/watch?v=DXPwSiRTxYY http://pandas.pydata.org/ Exercise: Statistical analysis of features in 20 newsgroups (oral presentation). Implement feature selection.
7. Toolbox: PIL. Topic: Image Processing. Reading: CV1, CV2, CV9. Exercise: Rank images by brightness (**exam**).
8. Toolbox: scikits, pyfaces. Topic: Face recognition and clustering. Reading: CV6, http://pyfaces.blogspot.dk/ Exercise: Reading groups present a face recognition experiment in class. Implementation and dataset of your own choice.
9. Toolbox: scikits, theano. Topic: Dimensionality reduction. Reading: http://scikit-learn.org/stable/modules/decomposition.html Exercise: Train auto-encoder (neural network) to learn low-dimensional representations.

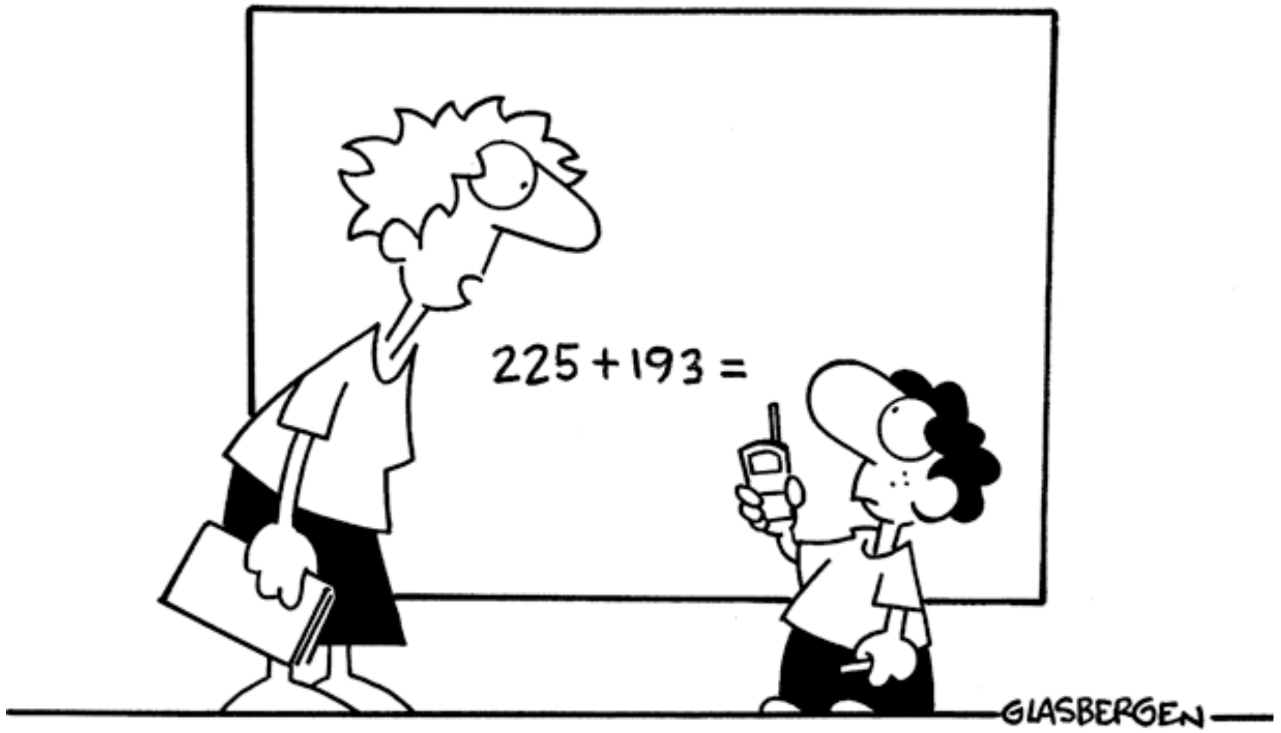
Unix cd, mkdir, ls, mv, rm, sudo, tail, more, grep




COMPUTER HOLY WARS
HOLD IT RIGHT THERE, BUDDY.
THAT SCRUFFY BEARD... THOSE SUSPENDERS... THAT SMUG EXPRESSION...
YOU'RE ONE OF THOSE CONDESCENDING UNIX COMPUTER USERS!
HERE'S A NICKEL, KID. GET YOURSELF A BETTER COMPUTER.
S. Adams E-mail: SCOTTADAMS@AOL.COM
6/24 © 1995 United Feature Syndicate, Inc. (NYC)

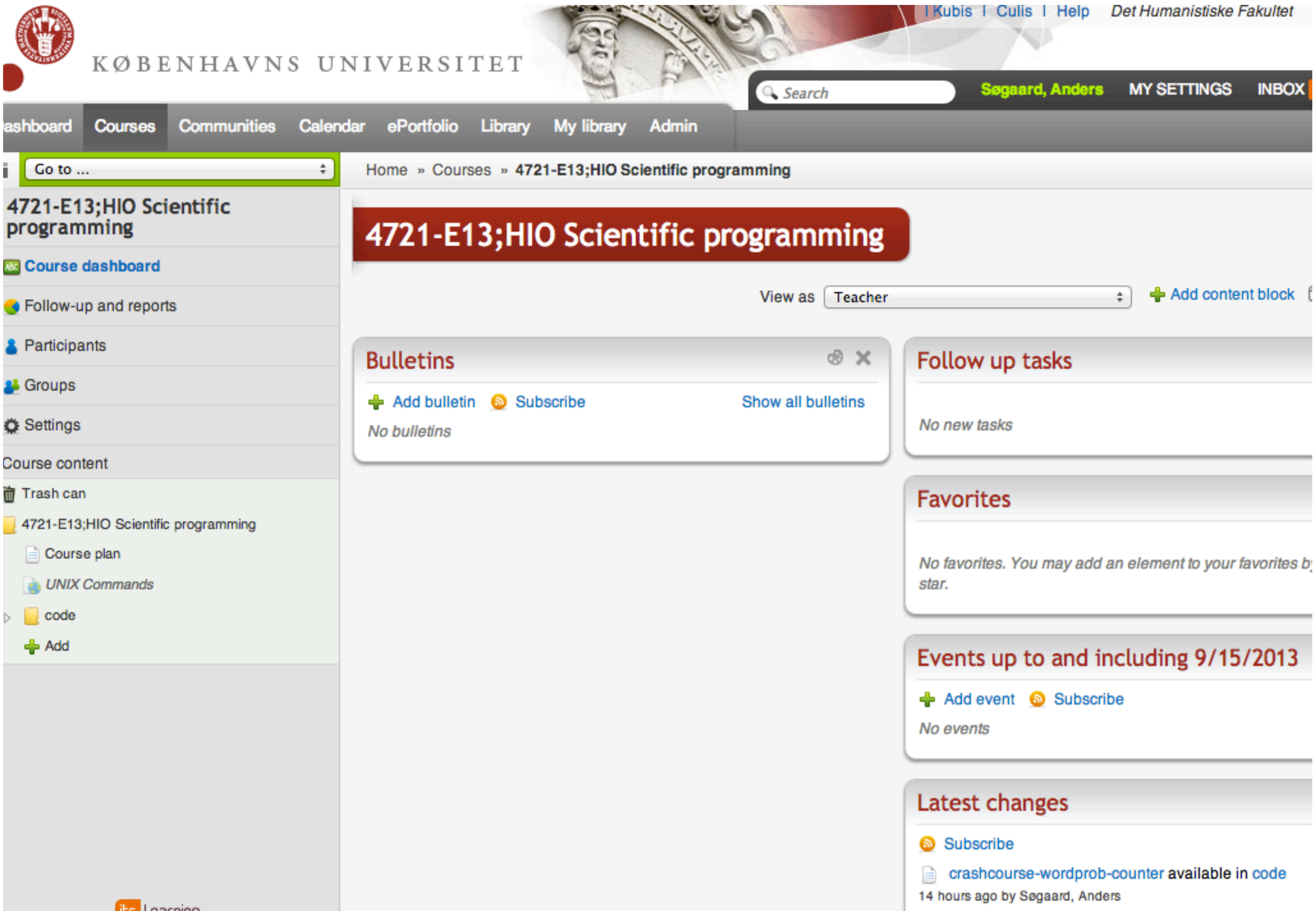

nano? REAL PROGRAMMERS USE emacs
HEY. REAL PROGRAMMERS USE vim.
WELL, REAL PROGRAMMERS USE ed.
NO, REAL PROGRAMMERS USE cat.
REAL PROGRAMMERS USE A MAGNETIZED NEEDLE AND A STEADY HAND.
EXCUSE ME, BUT REAL PROGRAMMERS USE BUTTERFLIES.
THEY OPEN THEIR HANDS AND LET THE DELICATE WINGS FLAP ONCE.
THE DISTURBANCE RIPPLES OUTWARD, CHANGING THE FLOW OF THE EDDY CURRENTS IN THE UPPER ATMOSPHERE.
THESE CAUSE MOMENTARY POCKETS OF HIGHER-PRESSURE AIR TO FORM,
WHICH ACT AS LENSES THAT DEFLECT INCOMING COSMIC RAYS, FOCUSING THEM TO STRIKE THE DRIVE PLATTER AND FLIP THE DESIRED BIT.
NICE.
'COURSE, THERE'S AN EMACS COMMAND TO DO THAT.
OH YEAH! GOOD OL' C-x M-c M-butterfly...
DAMMIT, EMACS.


PYTHON!
YOU'RE FLYING! HOW?
I LEARNED IT LAST NIGHT! EVERYTHING IS SO SIMPLE!
HELLO WORLD IS JUST print "Hello, world!"
I DUNNO... DYNAMIC TYPING? WHITESPACE?
COME JOIN US! PROGRAMMING IS FUN AGAIN! IT'S A WHOLE NEW WORLD UP HERE!
BUT HOW ARE YOU FLYING?
I JUST TYPED import antigravity
THAT'S IT?
... I ALSO SAMPLED EVERYTHING IN THE MEDICINE CABINET FOR COMPARISON.
BUT I THINK THIS IS THE PYTHON.



"You have to solve this problem by yourself. You can't call tech support."

Exercise 1 Write a line-to-BOW script (like crashcourse-bow.py) taking only frequent words into account; e.g, words with more than *m* occurrences.



4721-E13;HIO Scientific programming

Course dashboard
Follow-up and reports
Participants
Groups
Settings
Course content
Trash can
4721-E13;HIO Scientific programming
Course plan
UNIX Commands
code
Add

4721-E13;HIO Scientific programming

View as Teacher
Add content block

Bulletins

Add bulletin Subscribe Show all bulletins

No bulletins

Follow up tasks

No new tasks

Favorites

No favorites. You may add an element to your favorites b
star.

Events up to and including 9/15/2013

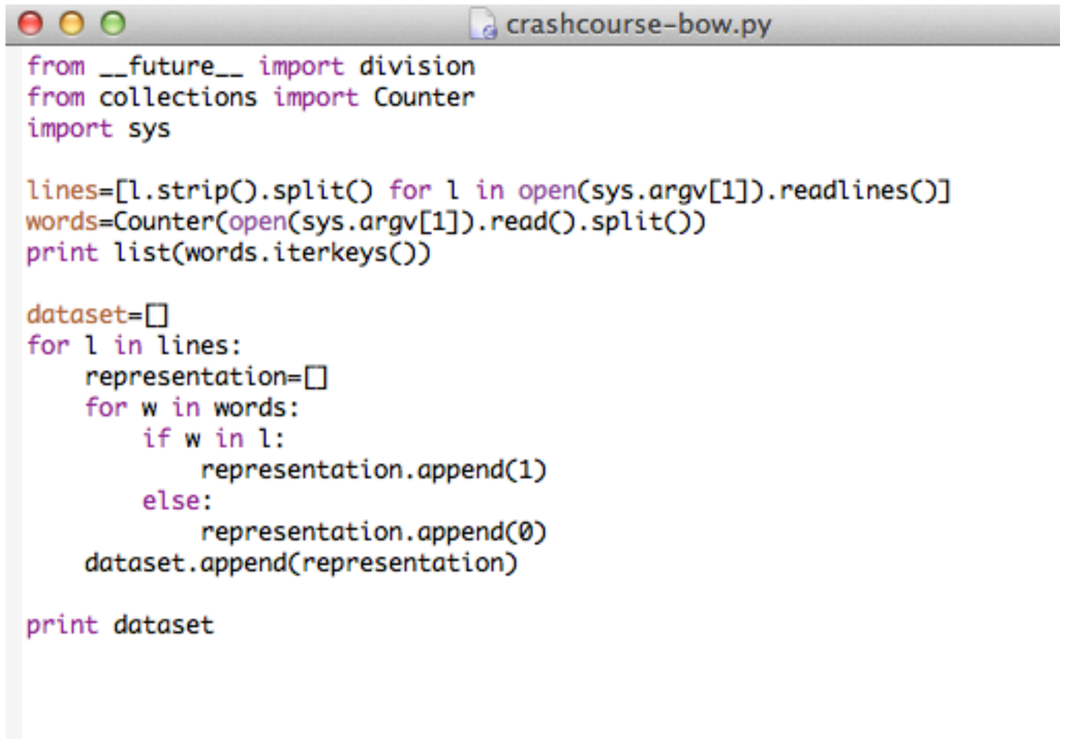
Add event Subscribe

No events

Latest changes

Subscribe

crashcourse-wordprob-counter available in code
14 hours ago by Søgaard, Anders

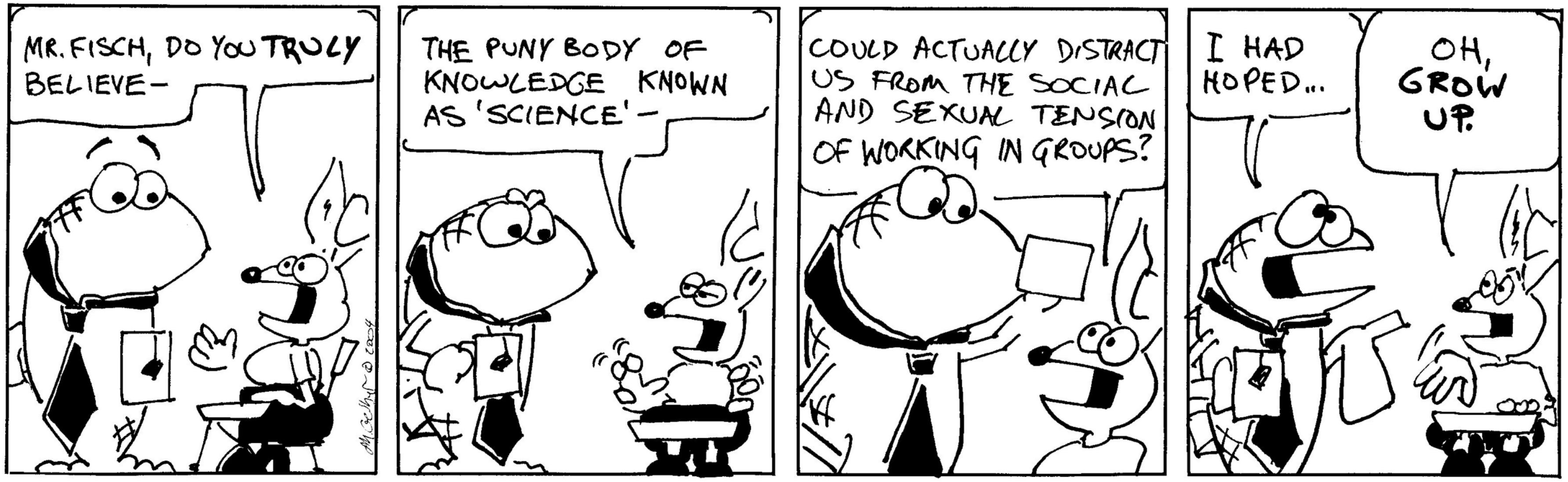

crashcourse-bow.py

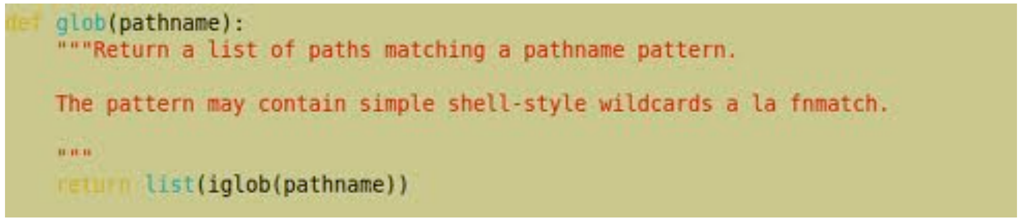
```
from __future__ import division
from collections import Counter
import sys

lines=[l.strip().split() for l in open(sys.argv[1]).readlines()]
words=Counter(open(sys.argv[1]).read().split())
print list(words.iterkeys())

dataset=[]
for l in lines:
    representation=[]
    for w in words:
        if w in l:
            representation.append(1)
        else:
            representation.append(0)
    dataset.append(representation)

print dataset
```

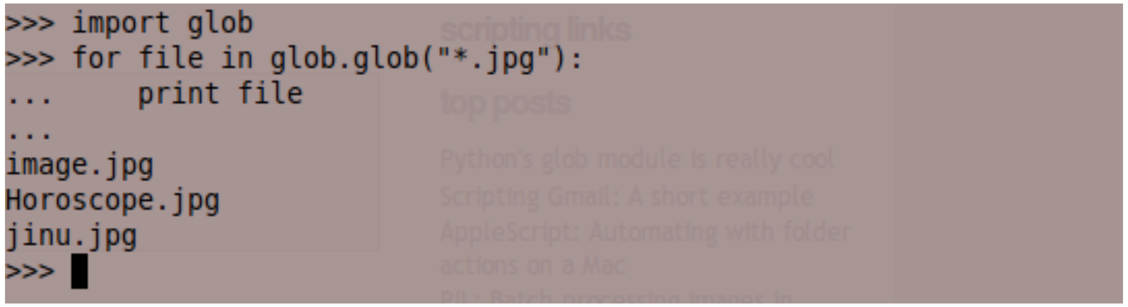

MR. FISCH, DO YOU TRULY BELIEVE–
THE PUNY BODY OF KNOWLEDGE KNOWN AS 'SCIENCE'–
COULD ACTUALLY DISTRACT US FROM THE SOCIAL AND SEXUAL TENSION OF WORKING IN GROUPS?
I HAD HOPED...
OH, GROW UP.



```
def glob(pathname):
    """Return a list of paths matching a pathname pattern.

    The pattern may contain simple shell-style wildcards a la fnmatch.

    """
    return list(iglob(pathname))
```



```
>>> import glob
>>> for file in glob.glob("*.jpg"):
...     print file
...
image.jpg
Horoscope.jpg
jinu.jpg
>>> 
```

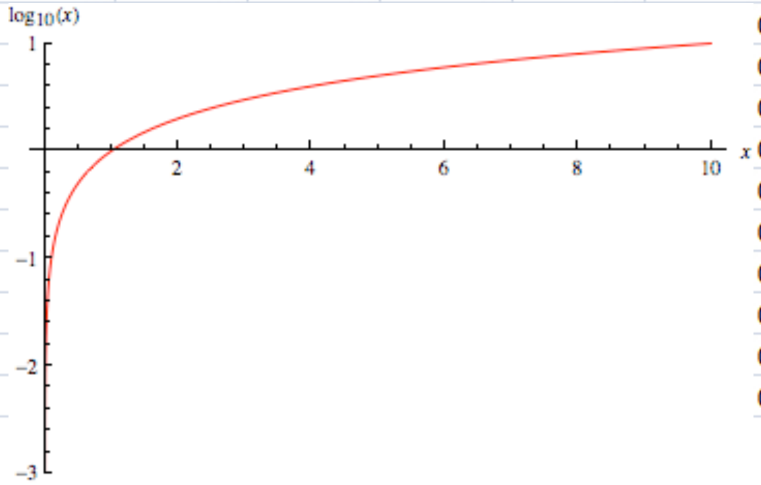
scripting links
top posts
Python's glob module is really cool
Scripting Gmail: A short example
AppleScript: Automating with folder actions on a Mac

Exercise II Write a file-to-BOW script using *import glob*.

Exercise III Write a file-to-BOW script with TF-IDF scores.

id	men	entered	bank	charlotte	missiles	masks	aryan	guns	witnesses	reported	silver	suv	august
seg1.txt	0.239441	0	0.153457	0.195243	0	0.237029	0	0.195243	0.237029	0.140004	0.195243	0.237029	0
seg13.txt	0	0	0	0	0	0	0	0	0	0	0	0	0
seg14.txt	0	0.192197	0	0	0	0	0	0	0	0	0	0	0.172681
seg15.txt	0	0	0	0	0	0	0	0	0	0	0	0	0.149652
seg16.txt	0	0	0	0	0	0	0	0	0	0	0	0	0
seg17.txt	0	0	0	0	0	0	0	0	0	0	0	0	0
seg18.txt	0	0.158432	0	0	0	0	0	0	0	0	0	0	0
seg19.txt	0	0	0	0.19725	[illegible]	[illegible]	[illegible]	[illegible]	0	0.141447	0	0	0.155038
seg2.txt	0	0	0	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0	0	0	0	0
seg20.txt	0	0.234323	0	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0	0	0	0	0
seg21.txt	0	0	0	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0	0	0	0	0
seg22.txt	0	0	0	0	0.135025	0	0.127589	0	0	0	0	0	0
seg23.txt	0	0	0	0	0	0	0	0	0	0.180656	0	0	0
seg24.txt	0	0	0	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0	0.117966	0	0	0
seg25.txt	0	0	0	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0	0	0	0	0
seg26.txt	0	0	0	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0	0	0	0	0
seg27.txt	0	0	0.235418	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0	0	0	0	0
seg28.txt	0	0	0	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0	0	0	0	0
seg29.txt	0	0	0	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0	0	0	0	0.142329
seg3.txt	0	0	0	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0	0	0	0	0
seg30.txt	0.078262	0	0	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0	0	0	0	0
seg31.txt	0	0	0.213409	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0	0	0	0	0
seg32.txt	0	0	0	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0	0	0	0	0

$$w_{i,j} = tf_{i,j} \times \log\left(\frac{N}{df_i}\right)$$



Term frequency		Document frequency		Normalization	
n (natural)	$\mathrm{tf}_{t,d}$	n (no)	1	n (none)	1
l (logarithm)	$1 + \log(\mathrm{tf}_{t,d})$	t (idf)	$\log \frac{N}{\mathrm{df}_t}$	c (cosine)	$\frac{1}{\sqrt{w_1^2+w_2^2+\ldots+w_M^2}}$
a (augmented)	$0.5 + \frac{0.5\times\mathrm{tf}_{t,d}}{\max_t(\mathrm{tf}_{t,d})}$	p (prob idf)	$\max\{0, \log\frac{N-\mathrm{df}_t}{\mathrm{df}_t}\}$	u (pivoted unique)	$1/u$ (Section 6.4.4)
b (boolean)	$\begin{cases} 1 & \text{if } \mathrm{tf}_{t,d} > 0 \\ 0 & \text{otherwise} \end{cases}$			b (byte size)	$1/\mathit{CharLength}^{\alpha}$, $\alpha < 1$
L (log ave)	$\frac{1+\log(\mathrm{tf}_{t,d})}{1+\log(\mathrm{ave}_{t\in d}(\mathrm{tf}_{t,d}))}$				

$$\mathrm{tf}(t,d) = 0.5 + \frac{0.5 \times \mathrm{f}(t,d)}{\max\{\mathrm{f}(w,d) : w \in d\}}$$

Exercise IV Problems where BOW don't cut it?

it was... ah, words can't describe!
are you sure? that's what they do

Exercise V and VI From documents to bigram representations, and BOW using external dictionaries...