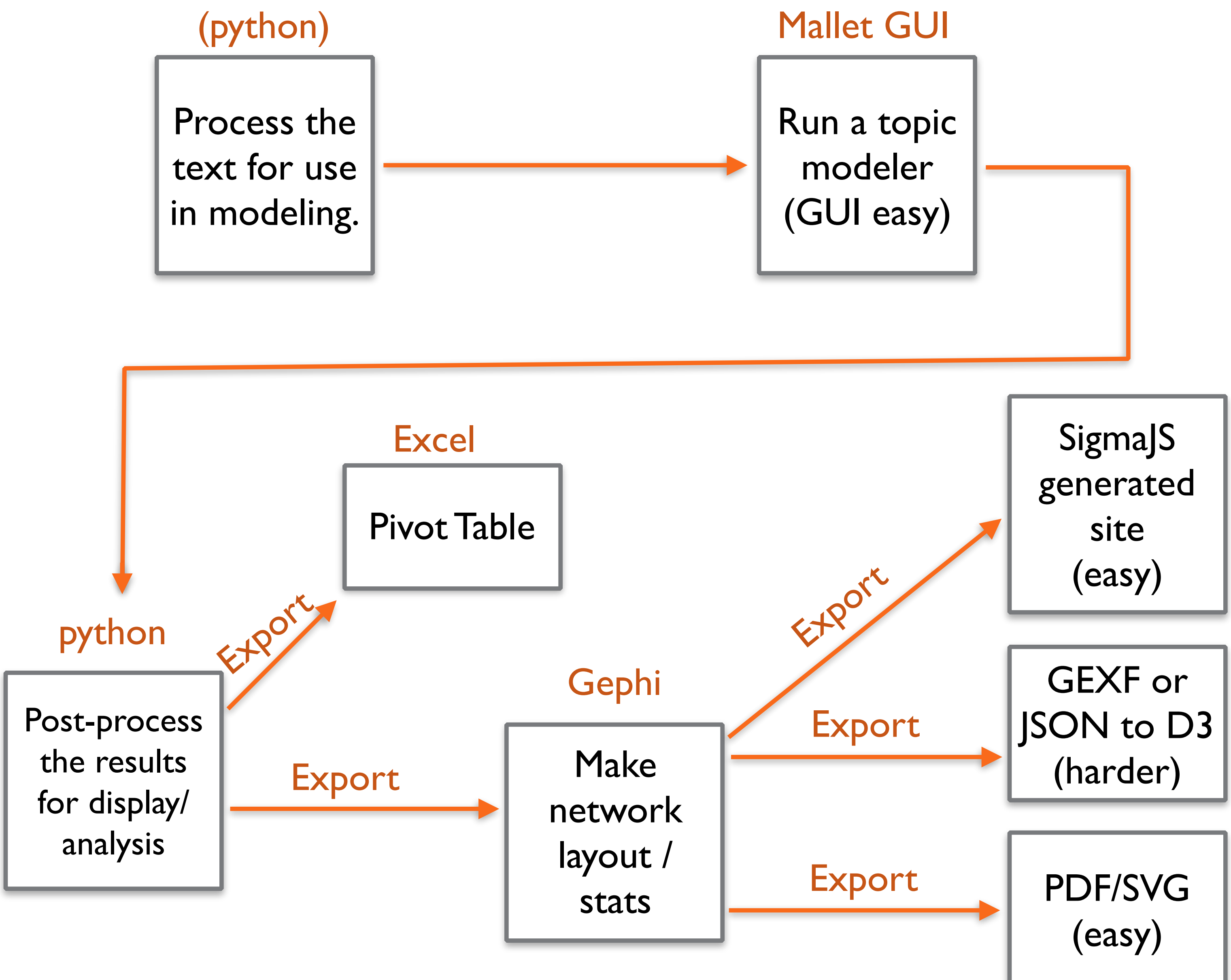
WHAT IS TOPIC ANALYSIS?

Problems We're Attacking

- Document collections are hard work to explore/manage manually
- Sometimes the contents are completely or mostly "unknown" (e.g., an email archive, or a collection of research papers)
- We'd like at least semi-automated methods to group them, annotate them, explore relationships

Workflow for Today



The Topic Problem

Text 1

We present a statistical parsing framework for sentence-level sentiment classification in this article. Different from previous work employing linguistic parsing results for sentiment analysis, we develop a statistical parser to directly analyze the sentiment structure of a sentence. We show that the complicated phenomena in sentiment analysis (e.g., negation, intensification, and contrast) can be elegantly handled the same as simple

Text 2

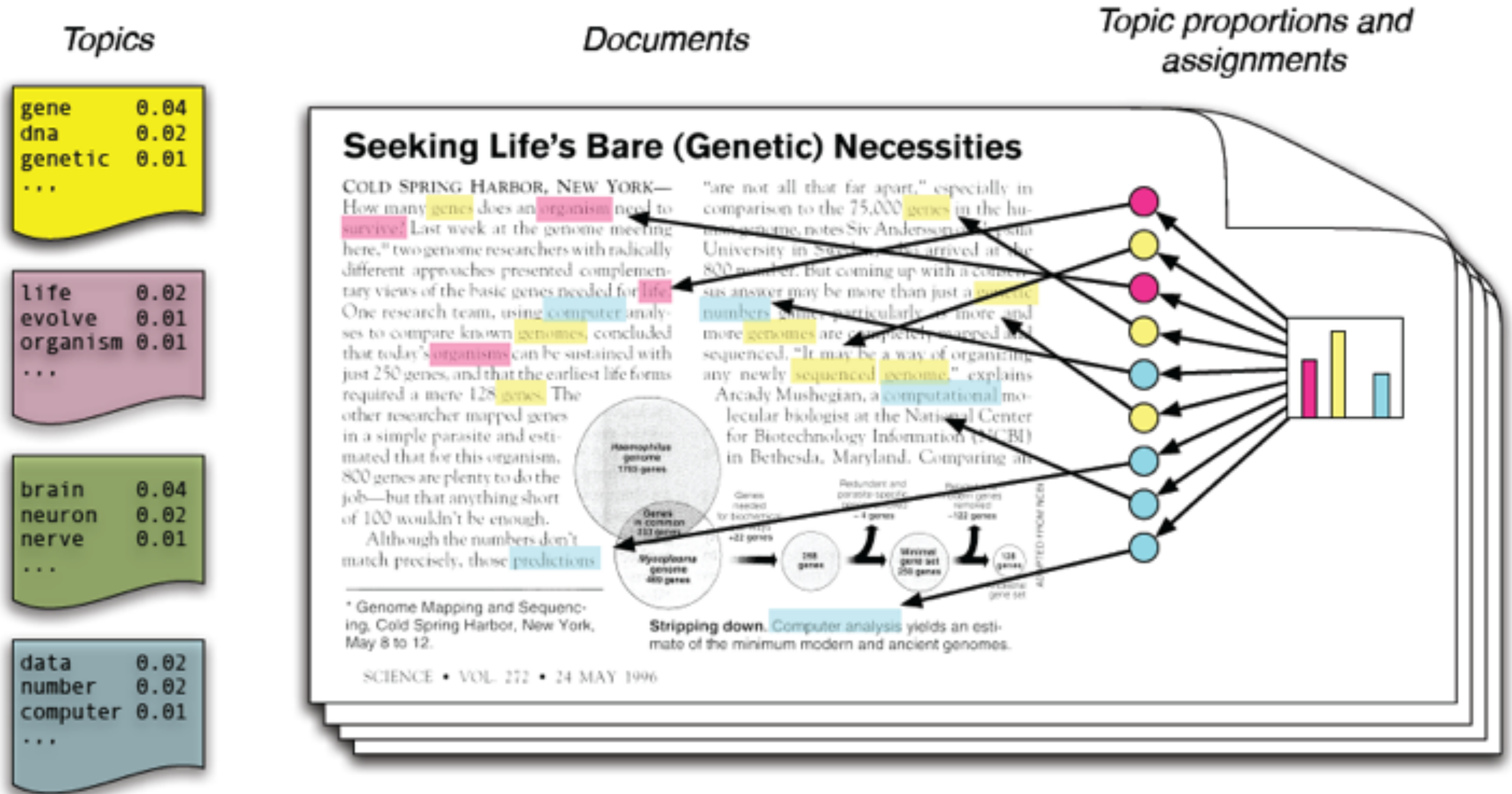
Sentiment analysis of Twitter data is performed. The researcher has made the following contributions via this paper: (1) an innovative method for deriving sentiment score dictionaries using an existing sentiment dictionary as seed words is explored, and (2) an analysis of clustered tweet sentiment scores based on tweet length is performed.

Text 3

We perform a large-scale linguistic analysis of language diatopic variation using geotagged microblogging datasets. By collecting all Twitter messages written in Spanish over more than two years, we build a corpus from which a carefully selected list of concepts allows us to characterize Spanish varieties on a global scale. A cluster analysis proves the existence of well defined macroregions sharing common lexical

Intuitions

- Documents are composed of multiple words (“bag of words”)
- Documents may express multiple topics



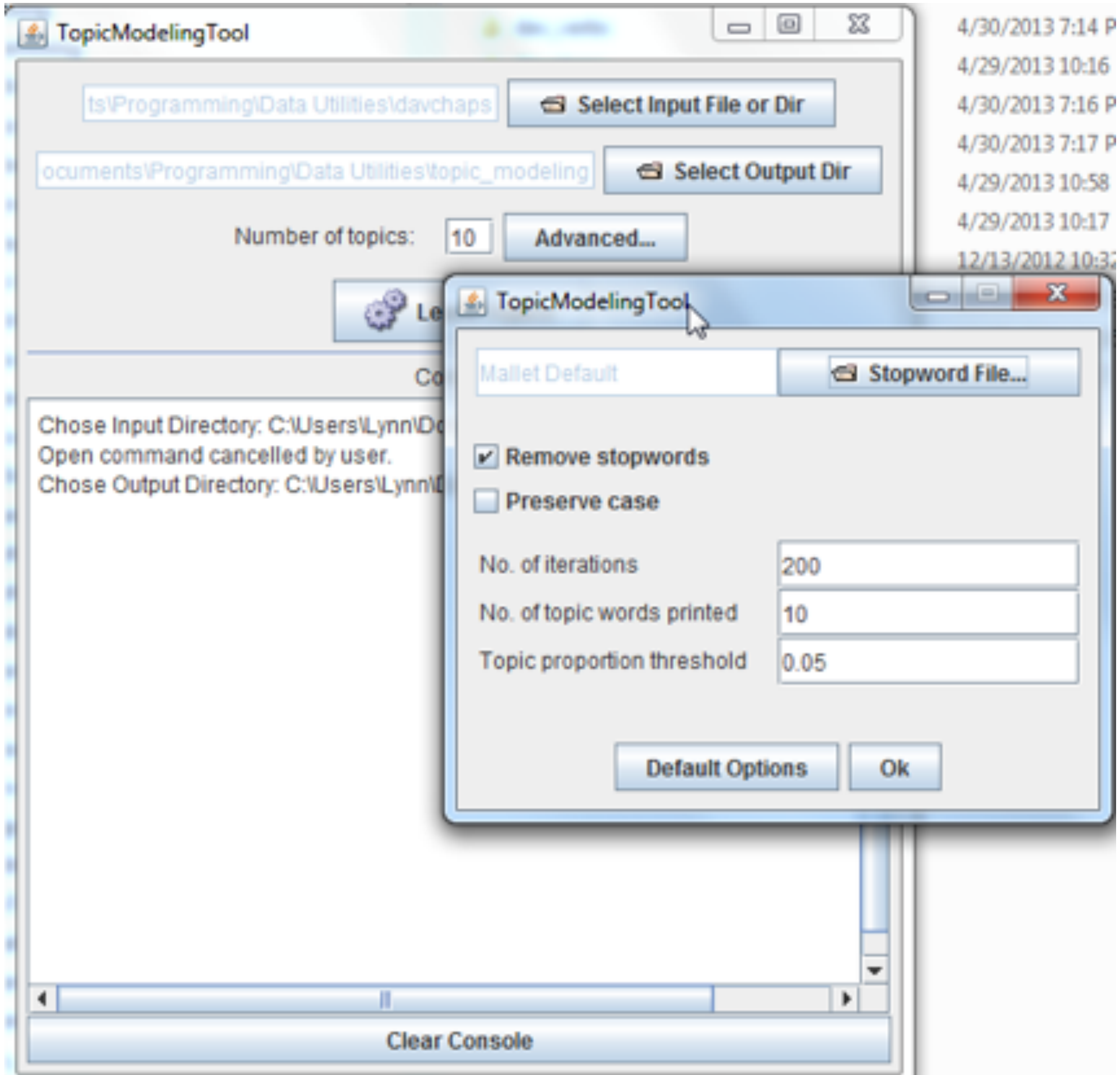
Topics
gene 0.04
dna 0.02
genetic 0.01
...
life 0.02
evolve 0.01
organism 0.01
...
brain 0.04
neuron 0.02
nerve 0.01
...
data 0.02
number 0.02
computer 0.01
...
Documents
Topic proportions and assignments
Seeking Life's Bare (Genetic) Necessities

Blei (2011)

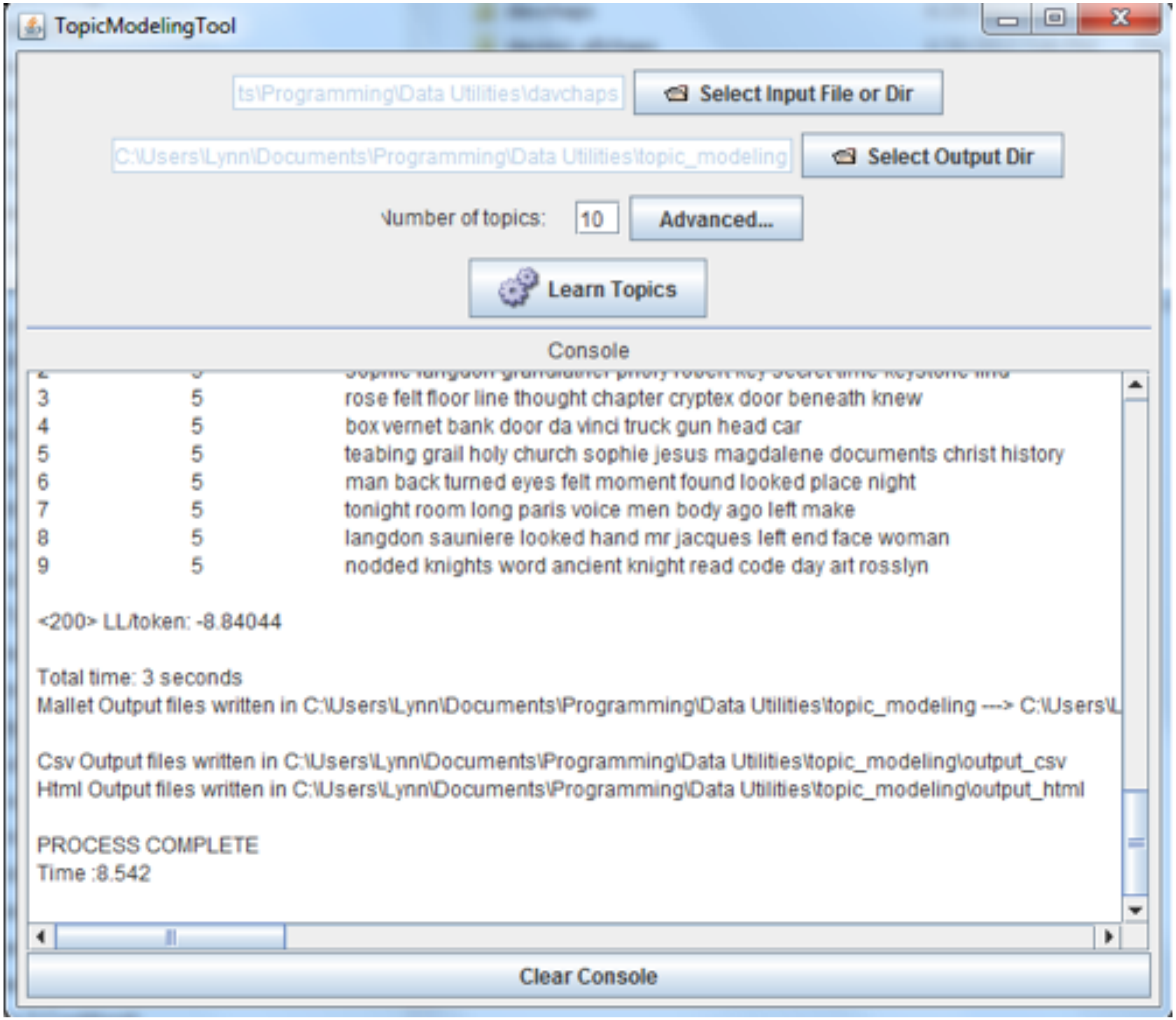
David Newman's Topic Modeling GUI

- Make a tool available for non-technical audiences! A GUI wrapper on the state-of-the-art mallet (java-based app by David Mimno).
- His overview slides:
- More of his work: http://www.ics.uci.edu/~newman/

Topic Modeling Tool (GUI)



Post run…


TopicModelingTool
ts\Programming\Data Utilities\davchaps
Select Input File or Dir
C:\Users\Lynn\Documents\Programming\Data Utilities\topic_modeling
Select Output Dir
Number of topics:
10
Advanced...
Learn Topics
Console
3 5 rose felt floor line thought chapter cryptex door beneath knew
4 5 box vernet bank door da vinci truck gun head car
5 5 teabing grail holy church sophie jesus magdalene documents christ history
6 5 man back turned eyes felt moment found looked place night
7 5 tonight room long paris voice men body ago left make
8 5 langdon sauniere looked hand mr jacques left end face woman
9 5 nodded knights word ancient knight read code day art rosslyn
<200> LL/token: -8.84044
Total time: 3 seconds
Mallet Output files written in C:\Users\Lynn\Documents\Programming\Data Utilities\topic_modeling ---> C:\Users\L
Csv Output files written in C:\Users\Lynn\Documents\Programming\Data Utilities\topic_modeling\output_csv
Html Output files written in C:\Users\Lynn\Documents\Programming\Data Utilities\topic_modeling\output_html
PROCESS COMPLETE
Time :8.542
Clear Console

Understanding the Output

StackOverflow post: http://stackoverflow.com/questions/8447393/how-to-understand-the-output-of-topic-model-class-in-mallet

```
<1450> LL/token: -9.11846
<1460> LL/token: -9.11803
<1470> LL/token: -9.10896
<1480> LL/token: -9.11237
<1490> LL/token: -9.10845
```

Iteration number

Log Likelihood per word (we want this to increase as the algorithm runs)

Output files: Data (csv), web site browser

output_csv

List of T topics:

Topics_Words.csv

List of topics in each of D documents:

TopicsInDocs.csv

List of top-ranked documents in each of T topics

DocsInTopics.csv

output_html

all_topics.html

Mallet command line

- You could also run mallet from the command line:
  - http://programminghistorian.org/lessons/topic-modeling-and-mallet

- Or use a Python wrapper:
  - http://radimrehurek.com/2014/03/tutorial-on-mallet-in-python/
  - To do the rest of this workshop, you'll need to process the output files yourself (assume \t seps, not csv)

Pros/Cons vs CMD-Line Mallet

Pros of GUI

- Allows stopword file specifying
- Produces csv and html output in a near dir structure
- Has a GUI (simpler to just get going without code and help)

Cons of GUI

- Runs with defaults, so no optimize-interval or other cmd line options
- No diagnostic output (a command-line option)

2 of the 3 CSV Output files

	A	B
1	topicId	words..
2	1	silas aringarosa remy teacher church dei opus bishop tomb vatican
3	2	fache collet police message neveu agent phone captain plane sir
4	3	sophie langdon grandfather priory robert key secret keystone time find
5	4	rose floor felt line thought chapter cryptex door knew began
6	5	box vernet bank vinci door head da louvre truck gun
7	6	teabing grail holy church sophie jesus magdalene documents history ch
8	7	man back turned felt eyes moment found looked place night

DocsInTopics.csv

Topics_Words.csv

TopicsInDocs.csv

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	docId	filename	top topics	and contribution to doc ...														
2	1	C:\Users\I	9	0.21	4	0.188	8	0.147	7	0.13	5	0.13	6	0.063	1	0.055		
3	2	C:\Users\I	8	0.232	9	0.21	7	0.143	4	0.095	2	0.095	3	0.072	10	0.068	5	0.057
4	3	C:\Users\I	4	0.274	1	0.212	7	0.137	8	0.127	5	0.062	3	0.053				
5	4	C:\Users\I	1	0.442	7	0.106	8	0.097	4	0.091	6	0.085						
6	5	C:\Users\I	6	0.175	3	0.165	4	0.146	9	0.124	7	0.098	5	0.097	1	0.076	8	0.062
7	6	C:\Users\I	1	0.333	8	0.235	4	0.216	6	0.069								
8	7	C:\Users\I	1	0.222	2	0.197	7	0.191	8	0.098	6	0.095	9	0.065				
9	8	C:\Users\I	4	0.242	10	0.18	3	0.178	7	0.118	9	0.104	6	0.065	5	0.057		
10	9	C:\Users\I	9	0.252	3	0.187	6	0.134	4	0.097	7	0.087	10	0.08	5	0.055	8	0.054
11	10	C:\Users\I	4	0.353	8	0.155	7	0.124	9	0.114	10	0.075	5	0.072				
12	11	C:\Users\I	2	0.314	9	0.24	3	0.097	7	0.09	8	0.073	10	0.072				
13	12	C:\Users\I	2	0.261	9	0.229	3	0.15	8	0.132	7	0.083						
14	13	C:\Users\I	9	0.309	2	0.16	3	0.157	8	0.091	7	0.091	6	0.058				
15	14	C:\Users\I	2	0.459	8	0.176	3	0.102	7	0.083	9	0.059						
16	15	C:\Users\I	1	0.25	4	0.182	7	0.12	8	0.104	5	0.089	3	0.089	6	0.057		
17	16	C:\Users\I	3	0.347	8	0.18	2	0.117	7	0.107	9	0.1	10	0.05	4	0.05		
18	17	C:\Users\I	2	0.354	8	0.173	3	0.116	9	0.104	10	0.065	7	0.057	5	0.051		
19	18	C:\Users\I	8	0.292	2	0.223	5	0.185	3	0.085	7	0.071	9	0.064	4	0.06		
20	19	C:\Users\I	8	0.244	1	0.199	4	0.124	3	0.11	9	0.09	5	0.076	7	0.07	6	0.059
21	20	C:\Users\I	1	0.312	8	0.157	3	0.114	6	0.107	4	0.1	7	0.084	5	0.066		
22	21	C:\Users\I	10	0.368	9	0.284	3	0.066	7	0.06	5	0.06	2	0.053				
23	22	C:\Users\I	3	0.256	9	0.216	5	0.165	8	0.09	7	0.079	2	0.069	4	0.062		
24	23	C:\Users\I	4	0.461	1	0.145	8	0.07	7	0.068	9	0.066	10	0.062				
25	24	C:\Users\I	3	0.368	5	0.131	8	0.111	4	0.107	9	0.099	7	0.087				
26	25	C:\Users\I	4	0.336	8	0.185	1	0.185	3	0.126	6	0.059						
27	26	C:\Users\I	2	0.457	8	0.104	9	0.098	7	0.098	10	0.069	3	0.069	4	0.064		
28	27	C:\Users\I	5	0.34	9	0.261	7	0.084	4	0.07	3	0.07	10	0.059				

Notice a horrible thing here:

docId	filename	top topics	and contribution to doc ...		
1	C:\Users\Lynn\Documents\Programming\Data Utilities\davchaps\chap_0.txt	9	0.21	4	0.188
2	C:\Users\Lynn\Documents\Programming\Data Utilities\davchaps\chap_1.txt	8	0.232	9	0.21
3	C:\Users\Lynn\Documents\Programming\Data Utilities\davchaps\chap_10.txt	4	0.274	1	0.212
4	C:\Users\Lynn\Documents\Programming\Data Utilities\davchaps\chap_100.txt	1	0.442	7	0.106
5	C:\Users\Lynn\Documents\Programming\Data Utilities\davchaps\chap_101.txt	6	0.175	3	0.165
6	C:\Users\Lynn\Documents\Programming\Data Utilities\davchaps\chap_102.txt	1	0.333	8	0.235
7	C:\Users\Lynn\Documents\Programming\Data Utilities\davchaps\chap_103.txt	1	0.222	2	0.197
8	C:\Users\Lynn\Documents\Programming\Data Utilities\davchaps\chap_104.txt	4	0.242	10	0.18
9	C:\Users\Lynn\Documents\Programming\Data Utilities\davchaps\chap_105.txt	9	0.252	3	0.187
10	C:\Users\Lynn\Documents\Programming\Data Utilities\davchaps\chap_106.txt	4	0.353	8	0.155
11	C:\Users\Lynn\Documents\Programming\Data Utilities\davchaps\chap_11.txt	2	0.314	9	0.24

This workshop has lots of code to process these files… see the .ipynb files and make_gephi_file.py

```
def read_doctopics(filename):
    from collections import defaultdict
    """ Takes topic_docs and outputs a dict style of topic assignment """
    docs = defaultdict(list)   # set because you
    with open(filename, 'rb') as csvfile:
        spamreader = csv.reader(csvfile, delimiter=',', quotechar='"')
        for row in spamreader:
            if spamreader.line_num > 1:
                docid = stripdir(row[1])  # beware not to use docid from the file in row[0]
                topics = row[2:]
                topics_dict = dict(zip(topics[::2],topics[1::2]))
                print docid, topics_dict
                docs[docid] = (topics_dict)
    return docs
```

```
chap_10 {'1': '0.212', '3': '0.053', '5': '0.062', '4': '0.274', '7': '0.137', '8': '0.127'}
chap_100 {'1': '0.442', '8': '0.097', '4': '0.091', '7': '0.106', '6': '0.085'}
chap_101 {'1': '0.076', '3': '0.165', '5': '0.097', '4': '0.146', '7': '0.098', '6': '0.175', '9': '0.124
chap_102 {'1': '0.333', '8': '0.235', '4': '0.216', '6': '0.069'}
chap_103 {'1': '0.222', '2': '0.197', '7': '0.191', '6': '0.095', '9': '0.065', '8': '0.098'}
chap_104 {'10': '0.180', '3': '0.178', '5': '0.057', '4': '0.242', '7': '0.118', '6': '0.065', '9': '0.10
chap_105 {'10': '0.080', '3': '0.187', '5': '0.055', '4': '0.097', '7': '0.087', '6': '0.134', '9': '0.25
chap_106 {'10': '0.075', '5': '0.072', '4': '0.353', '7': '0.124', '9': '0.114', '8': '0.155'}
chap_11 {'10': '0.072', '3': '0.097', '2': '0.314', '7': '0.090', '9': '0.240', '8': '0.073'}
chap_12 {'9': '0.229', '8': '0.132', '3': '0.150', '2': '0.261', '7': '0.083'}
chap_13 {'3': '0.157', '2': '0.160', '7': '0.091', '6': '0.058', '9': '0.309', '8': '0.091'}
chap_14 {'9': '0.059', '8': '0.176', '3': '0.102', '2': '0.459', '7': '0.083'}
```

The default HTML output is a little lacking…

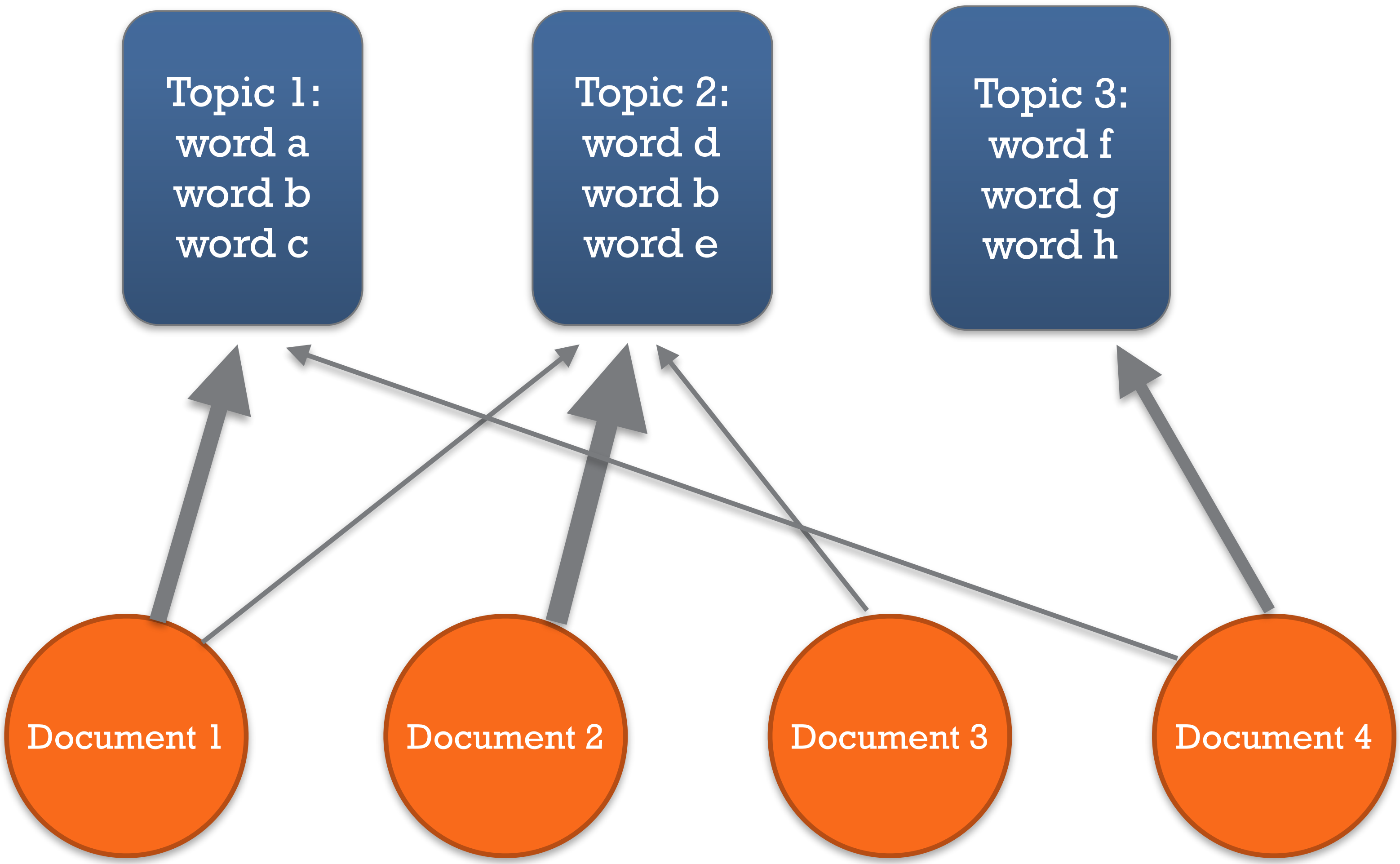
TOPIC : man back turned felt eyes moment found looked place night ...

top-ranked docs in this topic (#words in doc assigned to this topic)

2. (219) chap_67.txt
3. (193) chap_104.txt
4. (180) chap_84.txt
5. (179) chap_99.txt
6. (160) chap_51.txt
7. (153) chap_32.txt
8. (145) chap_81.txt

A bipartite graph of chapters and topics is an obvious vis method….

The results of topic modeling



Going to D3

- Raw nodes-edges json
  - export json from gephi
  - or post-process and create the json
- Export gexf and use Elijah Meeks' code to process and display it:
  - http://bl.ocks.org/emeeks/9357371

Network JSON for D3.js, this format:

```
nodes[0:10]
```

```
[{'name': 'chap_28', 'other': '1', 'type': 'doc'},
 {'name': 'chap_29', 'other': '1', 'type': 'doc'},
 {'name': 'chap_20', 'other': '0.714285714', 'type': 'doc'},
 {'name': 'chap_21', 'other': '1.1', 'type': 'doc'},
 {'name': 'chap_22', 'other': '0.333333333', 'type': 'doc'},
 {'name': 'chap_23', 'other': '1', 'type': 'doc'},
 {'name': 'chap_24', 'other': '0.5', 'type': 'doc'},
 {'name': 'chap_25', 'other': '0.5', 'type': 'doc'},
 {'name': 'chap_26', 'other': '0.75', 'type': 'doc'},
 {'name': 'chap_27', 'other': '1.5', 'type': 'doc'}]
```

```
links[0:10]
```

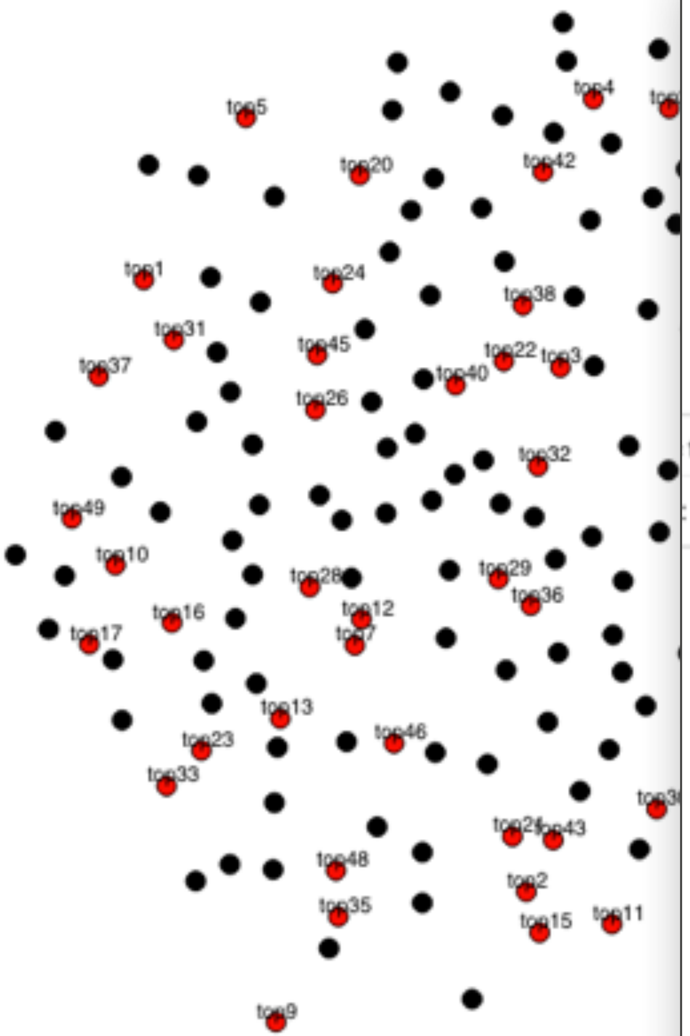
```
[{'source': 'chap_28', 'strength': '0.073', 'target': 'top10'},
 {'source': 'chap_28', 'strength': '0.108', 'target': 'top3'},
 {'source': 'chap_28', 'strength': '0.077', 'target': 'top5'},
 {'source': 'chap_28', 'strength': '0.083', 'target': 'top4'},
 {'source': 'chap_28', 'strength': '0.071', 'target': 'top7'},
 {'source': 'chap_28', 'strength': '0.183', 'target': 'top6'},
 {'source': 'chap_28', 'strength': '0.270', 'target': 'top9'},
 {'source': 'chap_28', 'strength': '0.073', 'target': 'top8'},
 {'source': 'chap_29', 'strength': '0.090', 'target': 'top10'},
 {'source': 'chap_29', 'strength': '0.110', 'target': 'top1'}]
```

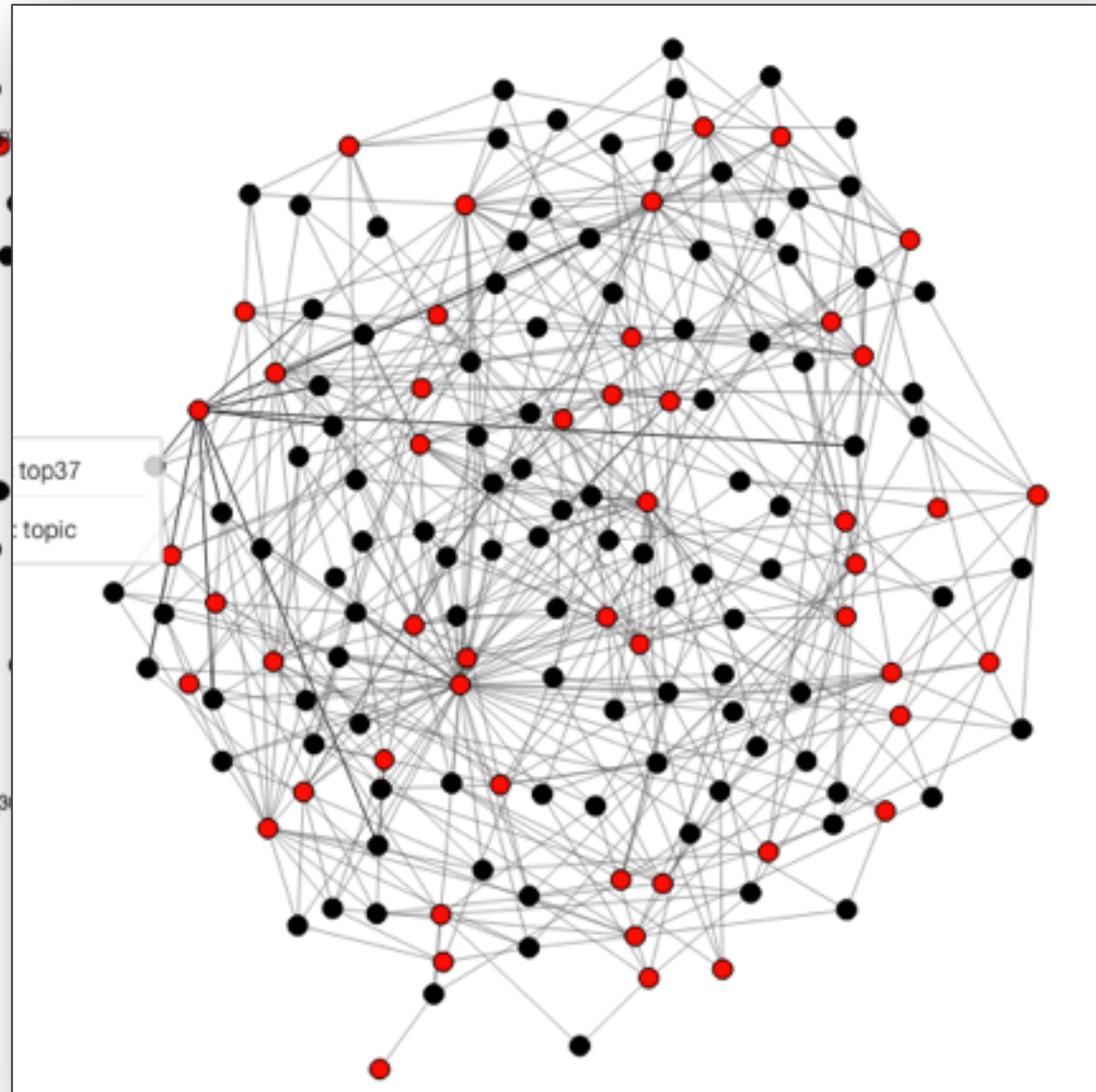
Making the objects:

Make objects of nodes, links, and any extra data values on each that you want…

```
def make_nodes_links(results, otherdata=None):

    """ Write out dicts for later creating a json file of nodes/links from a results array of 'doc, topic, strength' values.
    Must make sure the 'otherdata' dict has the same keys as the chapter names here."""

    chapters = set()
    topics = set()

    for x in results:
        chapters.add(x[0])
        topics.add(x[1])

    chapters = list(chapters)
    topics = list(topics)

    nodes = []

    for chapter in chapters:
        if otherdata:
            nodes.append({"name": chapter, "type": "doc", "other": otherdata[chapter]})
        else:
            nodes.append({"name": chapter, "type": "doc"})

    for topic in topics:
        nodes.append({"name": "top" + topic, "type": "topic"})

    links = []
    for x in results:
        doc = x[0]
        topic = x[1]
        strength = x[2]
        links.append({"source": doc, "target": "top" + topic, "strength": strength})

    return nodes, links
```

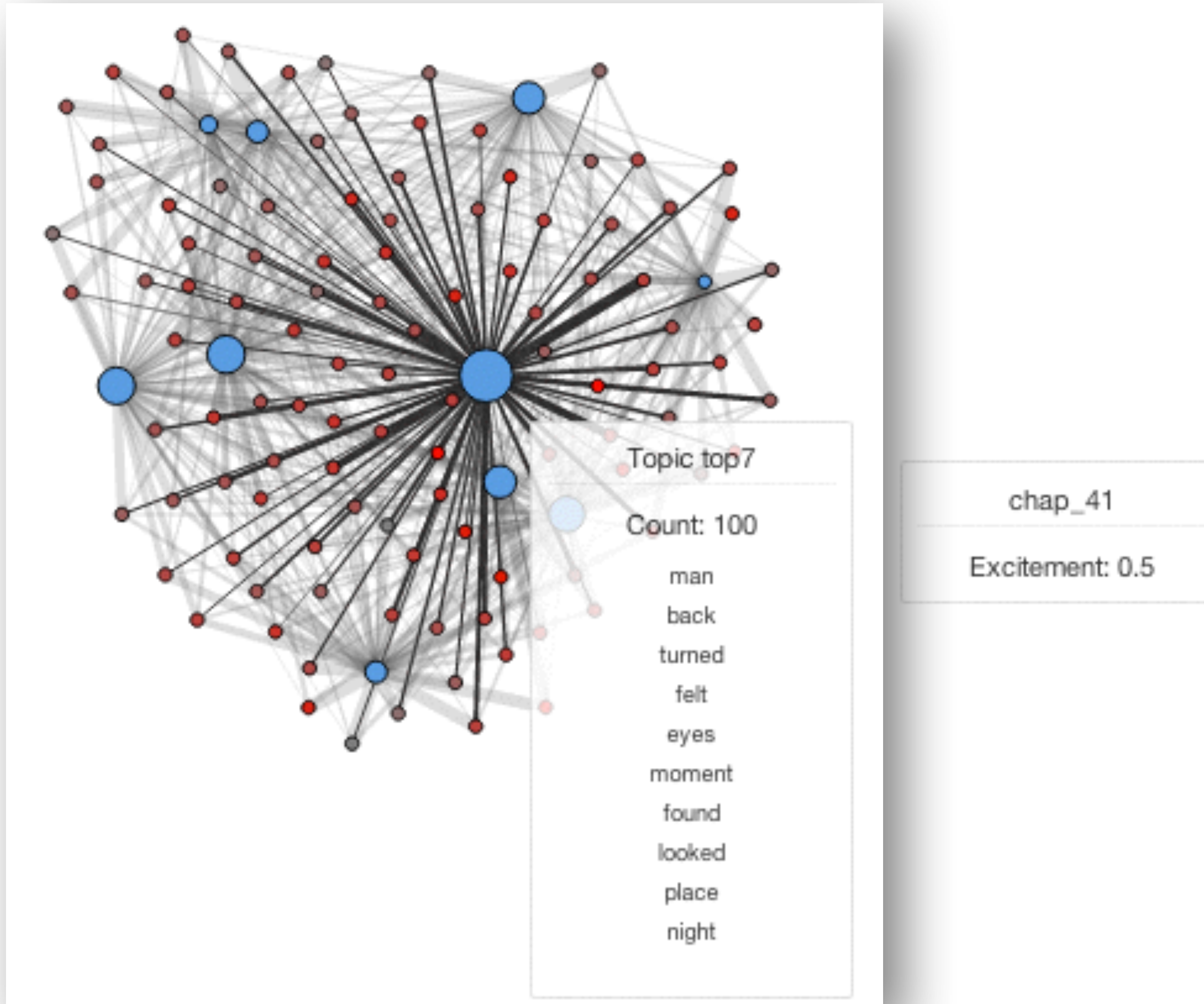
Let's try a hairball!





Improving the network's UI…

Demo:
http://www.ghostweather.com/essays/talks/openvisconf/topic_docs_network/index_better.html

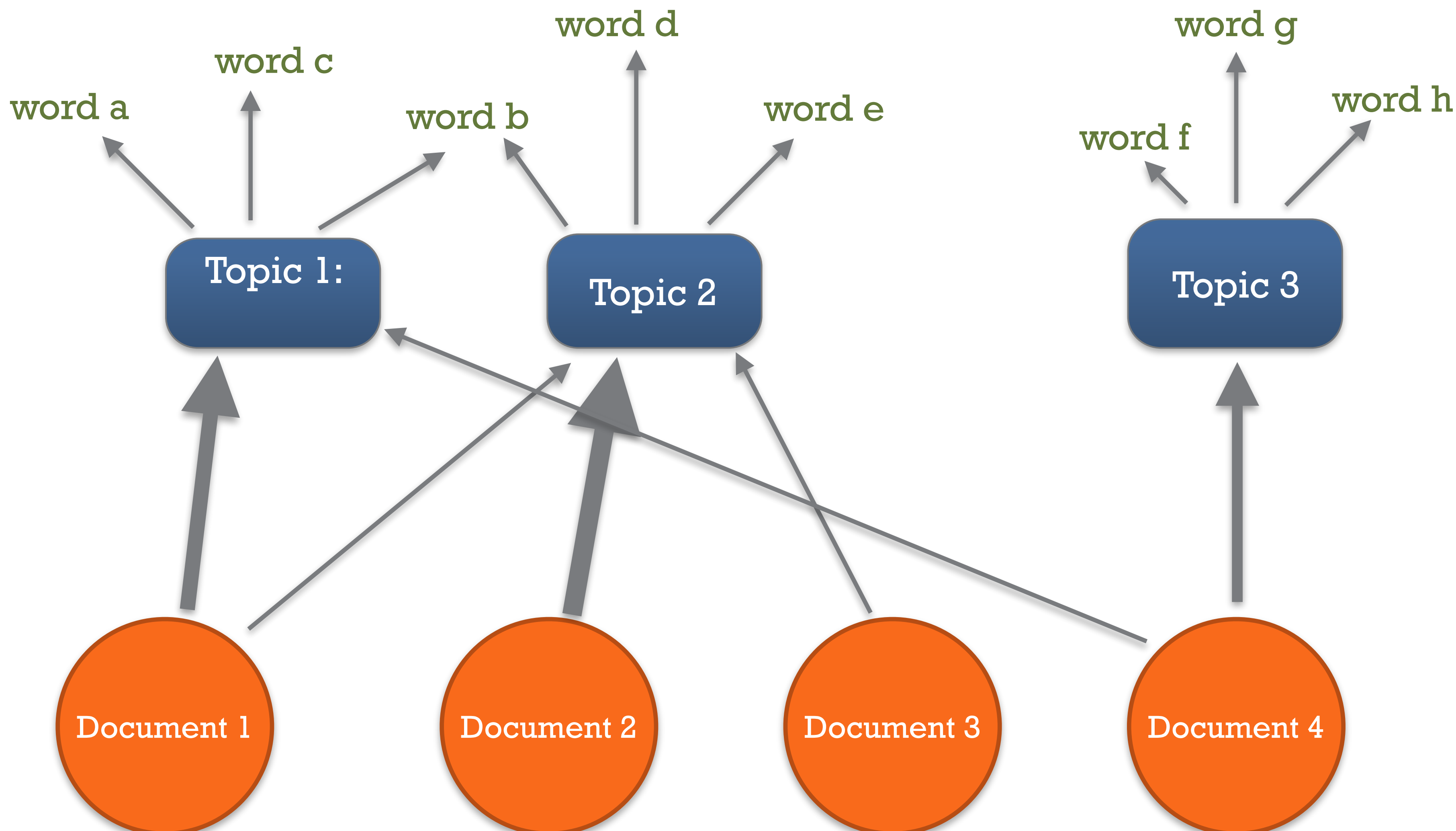


Adding strength, highlight effect, another variable, and informative tooltips.

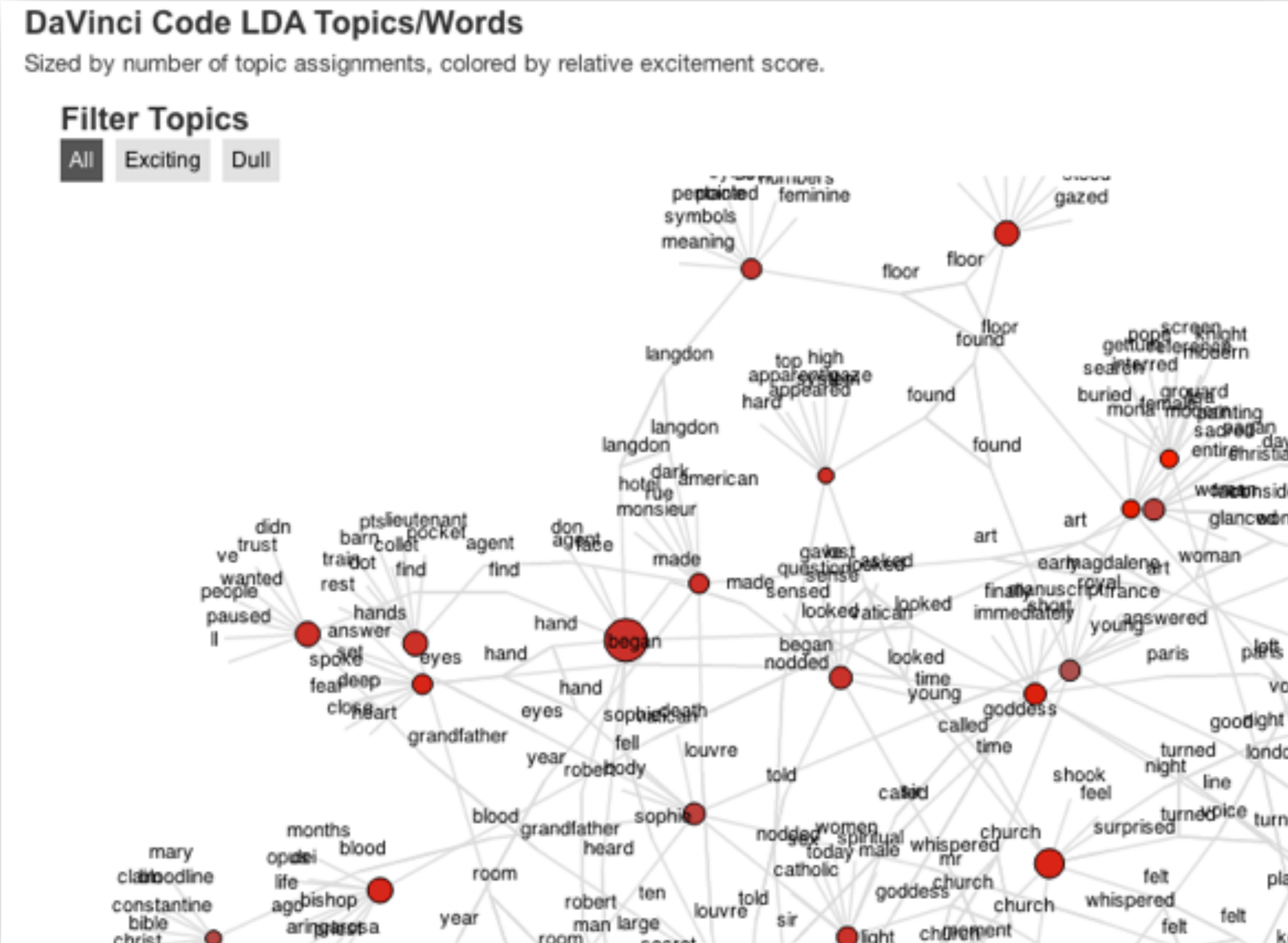
Tricks in D3 – scales for sizing relatively:

```
countExtent = d3.extent(data.topics, (d) -> +d.counts)
circleRadius = d3.scale.linear().range([5, 15]).domain(countExtent)

countStrength = d3.extent(data.links, (d) -> +d.strength)
linkSize = d3.scale.linear().range([.5,6]).domain(countStrength)
```

A further level of network you could draw....



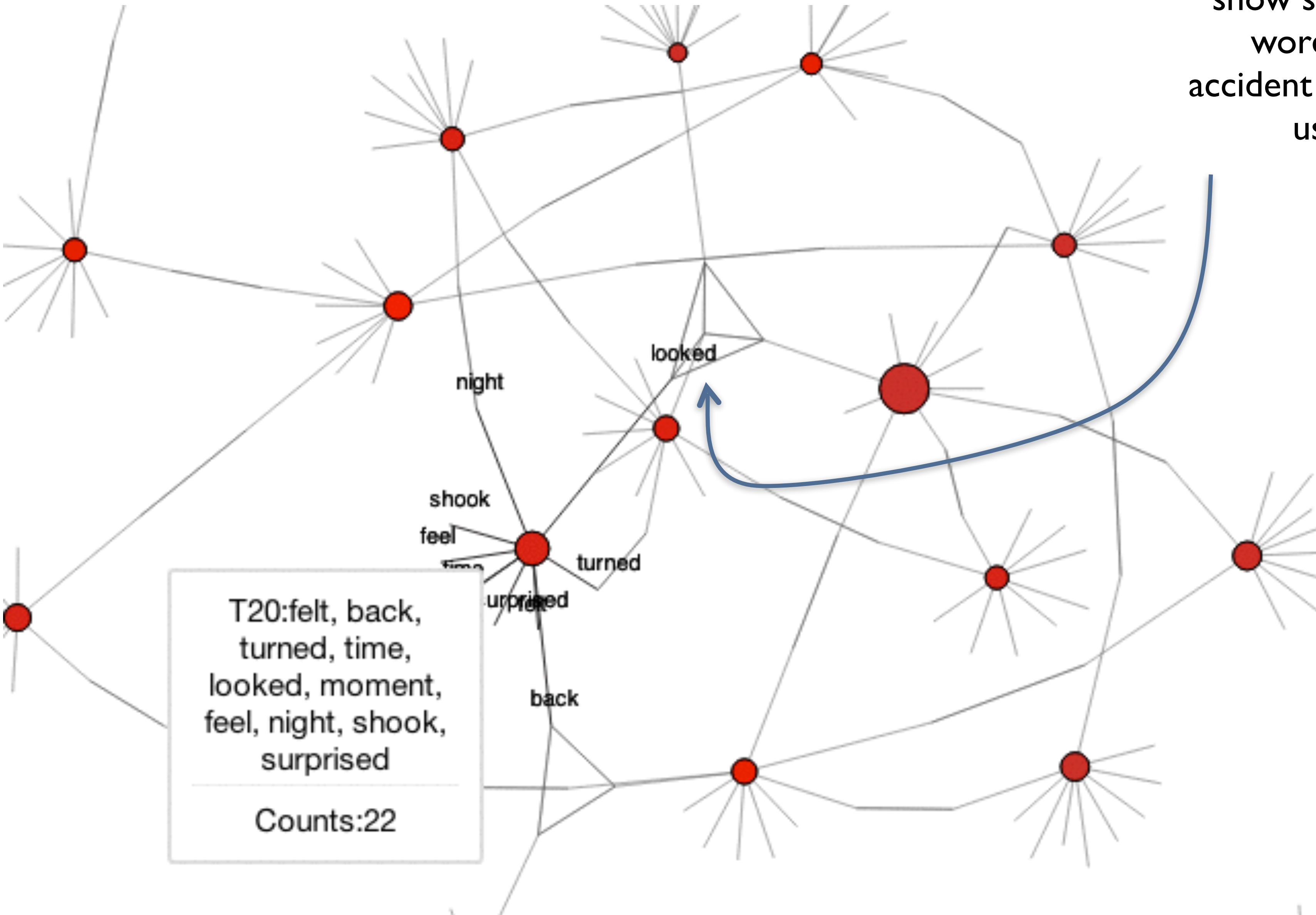
Maybe I need One More Tool. Any word relations of interest?
Let's try another hairball…



Demo: http://www.ghostweather.com/essays/talks/openvisconf/topic_words_network/index.html

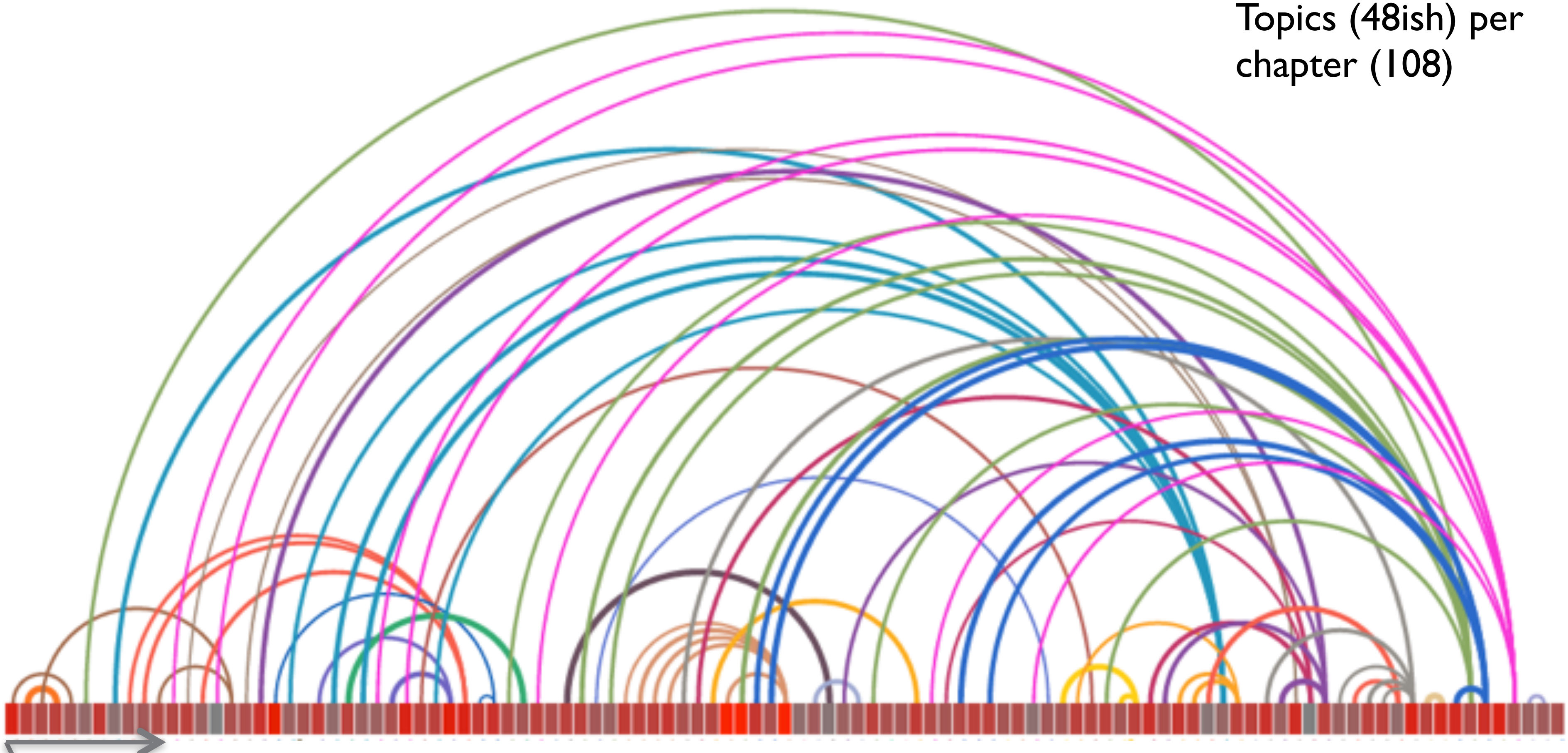
Filtered to only the "exciting" nodes…

Small "constellations" show shared words (an accident that's useful!)



Another tool: Sequential documents, with topic arcs.
DaVinci Code topics to chapters mapping

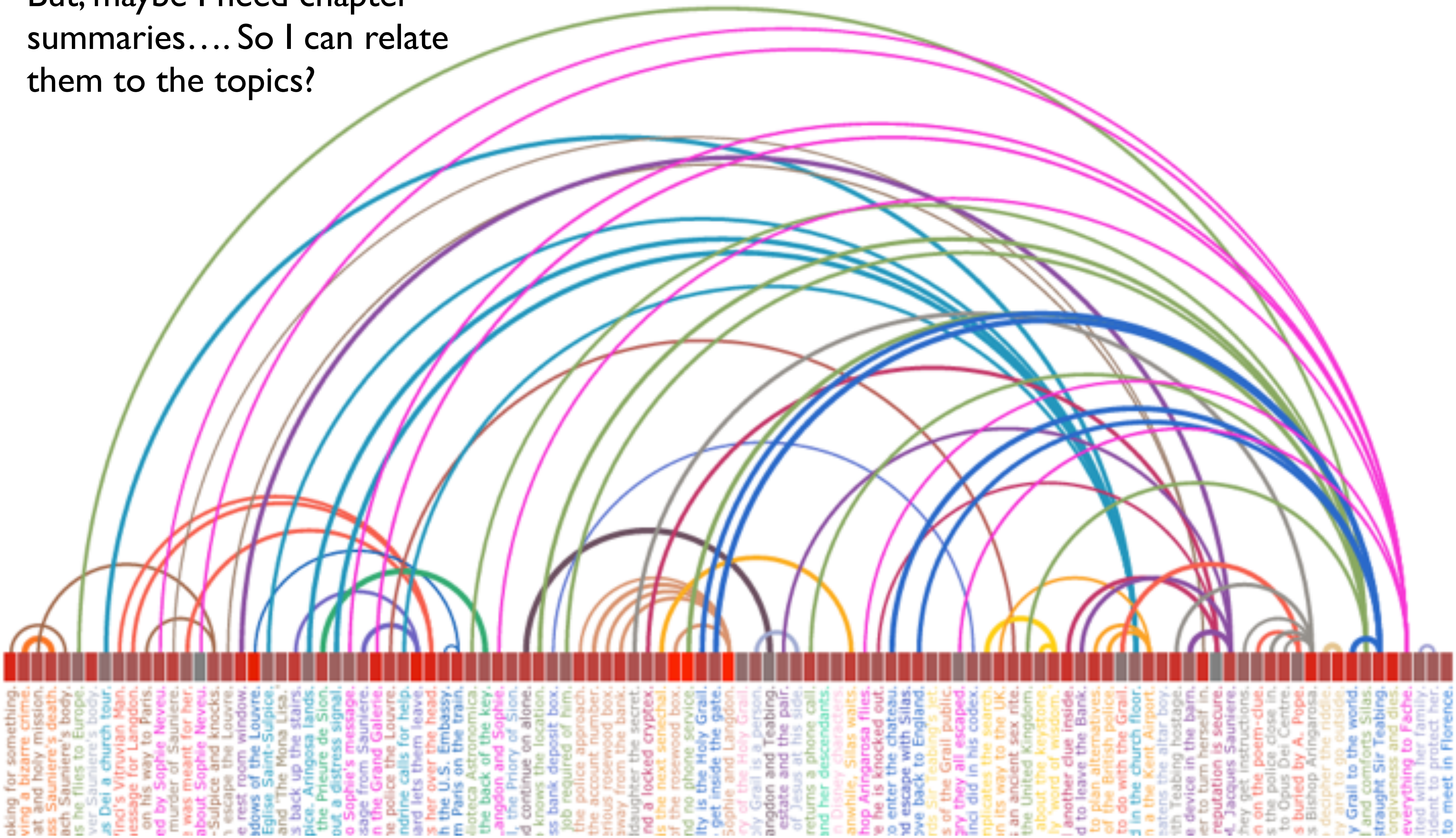
Topics (48ish) per chapter (108)



Chapter 1… to Chapter 108

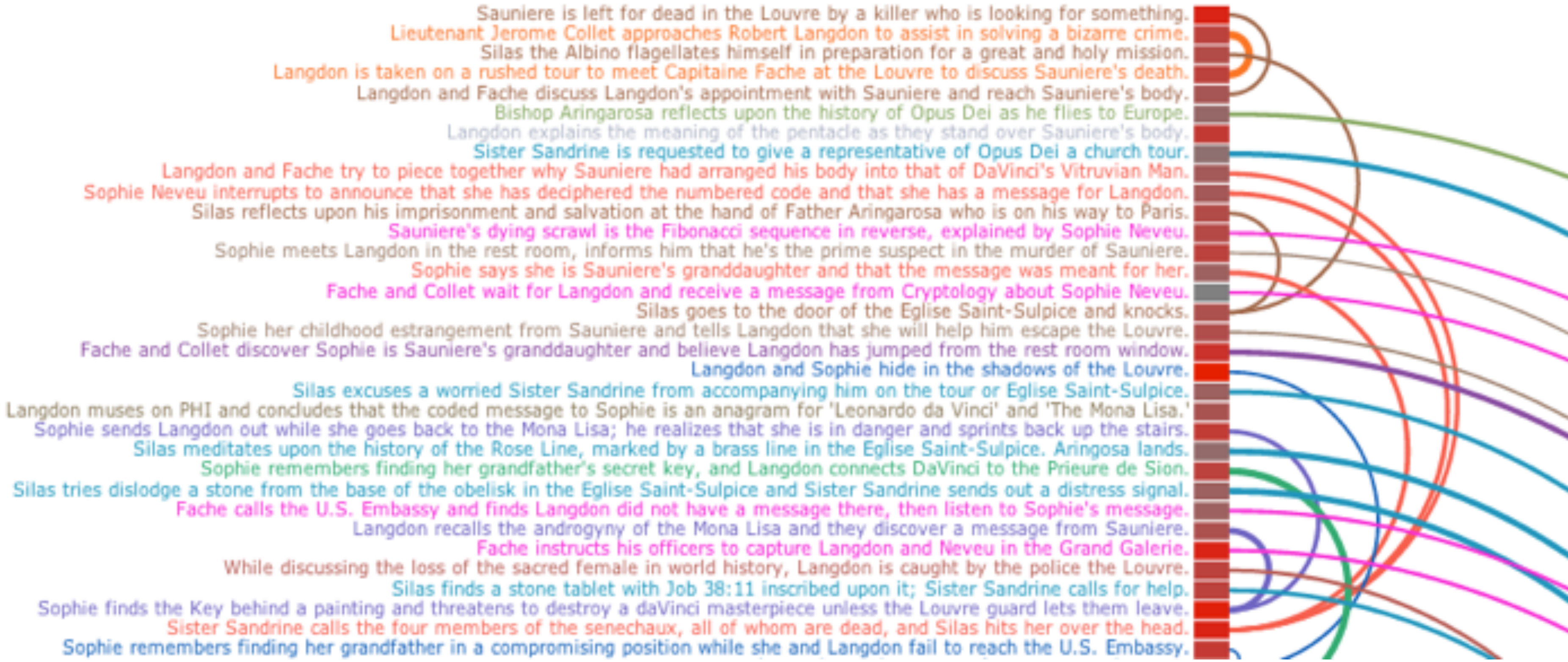
"Excitement" rating color scale avg by chapter, ordered (obviously)

But, maybe I need chapter summaries…. So I can relate them to the topics?



Ah, but since it's svg/d3…

```
var chart = chart.append("g").attr("translate","0," + y).attr("transform","rotate(90 600 600)");
```

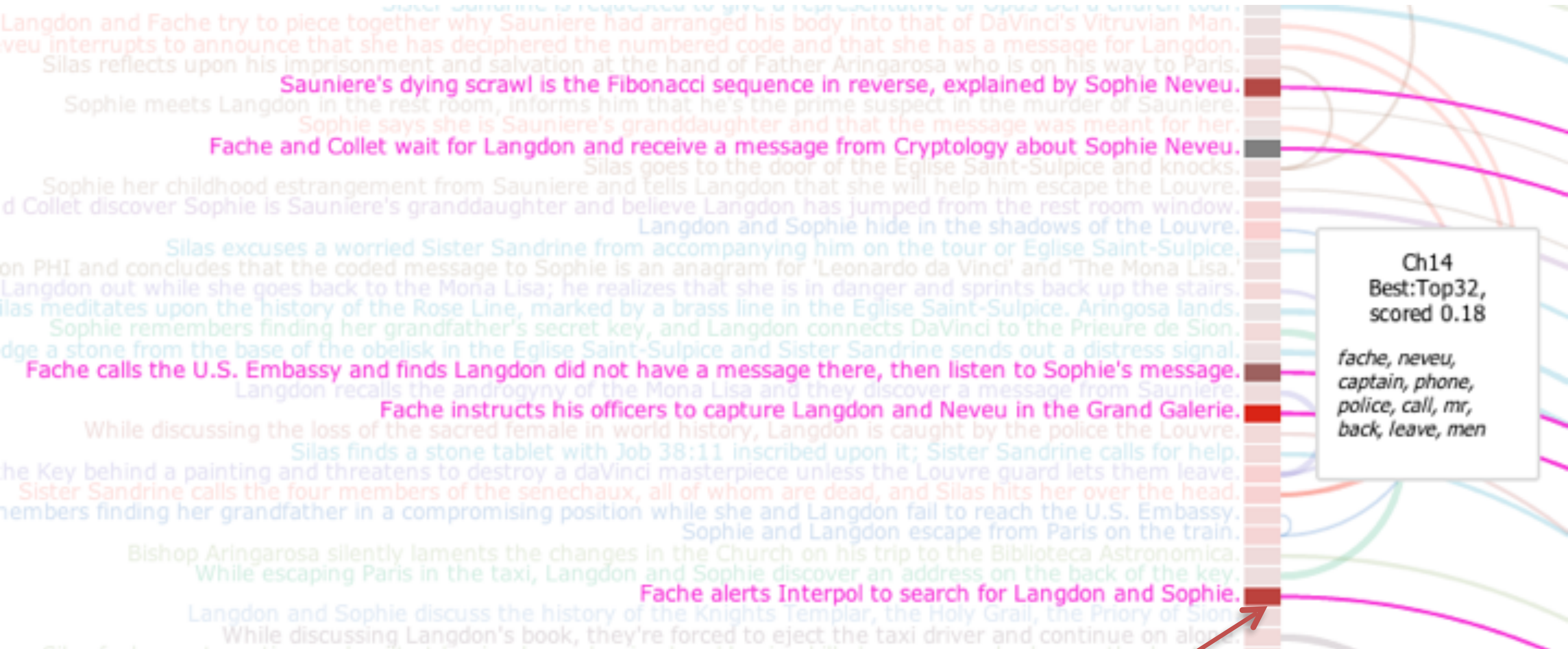


Add some topic-tooltips and fade-outs….

Demo: http://www.ghostweather.com/essays/talks/openvisconf/topic_arc_diagram/TopicArc.html

But what did this show?

Some topics are just neither exciting nor dull – topic clustering (as I did it) had little to do with action scenes. It's slightly helpful for topics, though ☺



These nodes are shaded from gray (dull) to red (exciting)

How can you improve on the results?

Topic modeling depends on the input:

pre-process the documents differently:

- only verbs / nouns?
  - Read in a document, parse it, save out a new "document" of the POS you want
- use stop words tuned for your data set (don't want proper nouns? or only proper nouns?)
- iterate on the number of topics you output

Improve the Results Display

- Visualize differently or more…
- Look for the topic words "in context" - find sentences with them and use those as part of your topic description
- Construct phrases from your topic words to make them "better" for descriptors
- Use only the interesting output words for a topic
- Don't use the result immediately — use as input to other methods (it's a data reduction technique like principal components analysis)

A Few More References

- Scott Weingart's nice overview of LDA Topic Modeling in Digital Humanities: http://www.scottbot.net/HIAL/?p=221
- Elijah Meeks' lovely set of articles on LDA & Digital Humanties vis: https://dhs.stanford.edu/comprehending-the-digital-humanities/
- Some pure python (and C) implementations (toy code, primarily) are listed on Blei's website: http://www.cs.princeton.edu/~blei/topicmodeling.html