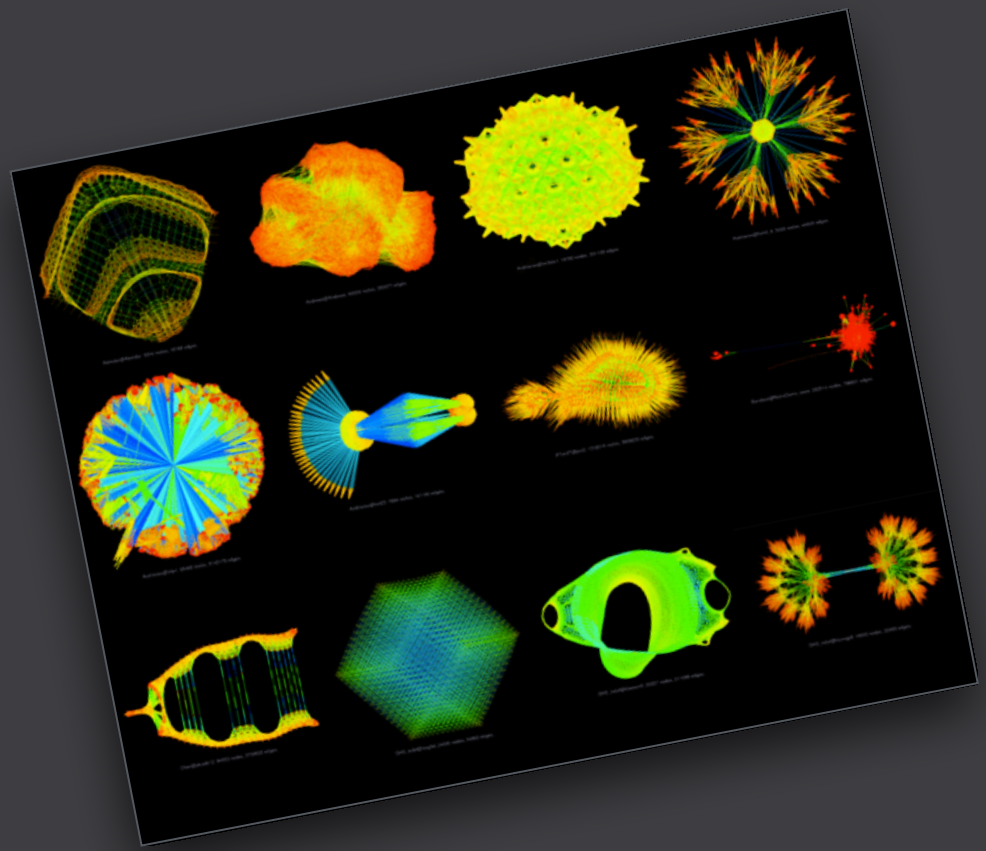
PyTextRank

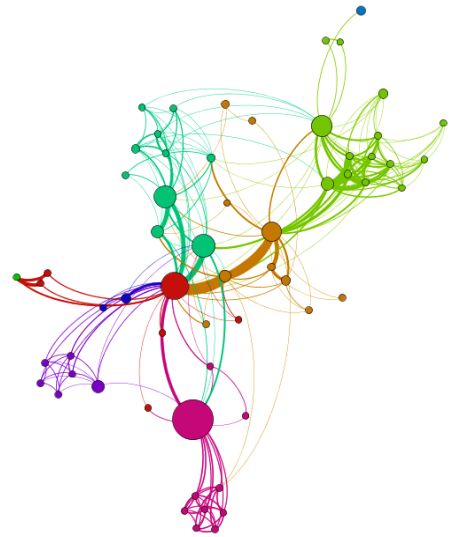
2017-02-08 SF Python Meetup

Paco Nathan, @pacoid
Dir, Learning Group @ O'Reilly Media

Just Enough Graph

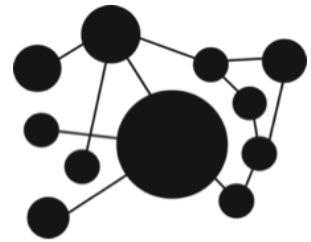


Graph Analytics: terminology

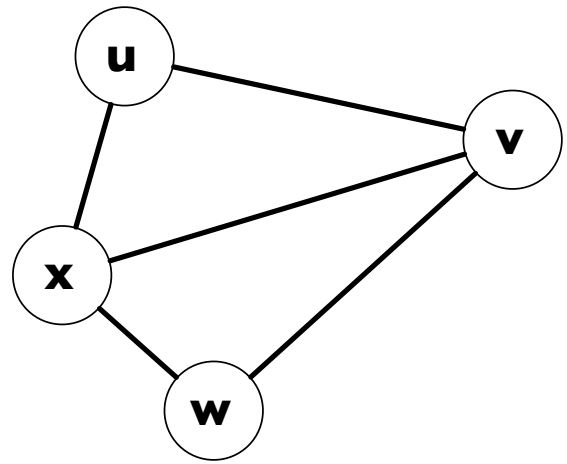


- many real-world problems are often represented as *graphs*
- graphs can generally be converted into *sparse matrices* (bridge to linear algebra)
- *eigenvectors* find the stable points in a system defined by matrices – which may be more efficient to compute
- beyond simpler graphs, complex data may require work with *tensors*

Graph Analytics: concepts



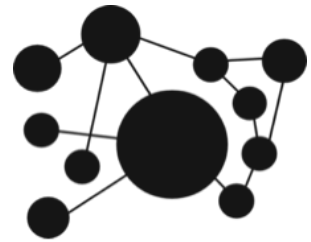
Suppose we have a graph as shown below:



We call **x** a *vertex* (sometimes called a *node*)

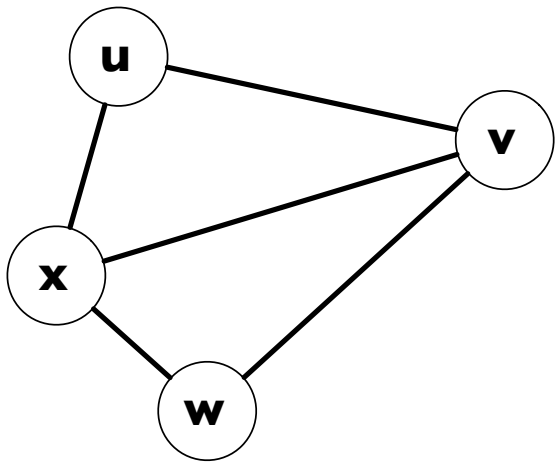
An *edge* (sometimes called an *arc*) is any line connecting two vertices

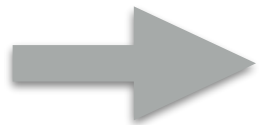
Graph Analytics: representation



We can represent this kind of graph as an *adjacency matrix*:

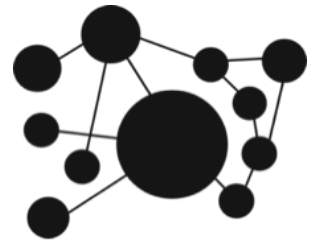
- label the rows and columns based on the vertices
- entries get a 1 if an edge connects the corresponding vertices, or 0 otherwise





	u	v	w	x
u	0	1	0	1
v	1	0	1	1
w	0	1	0	1
x	1	1	1	0

Algebraic Graph Theory



An *adjacency matrix* always has certain properties:

- it is *symmetric*, i.e., $\mathbf{A} = \mathbf{A}^T$
- it has real eigenvalues

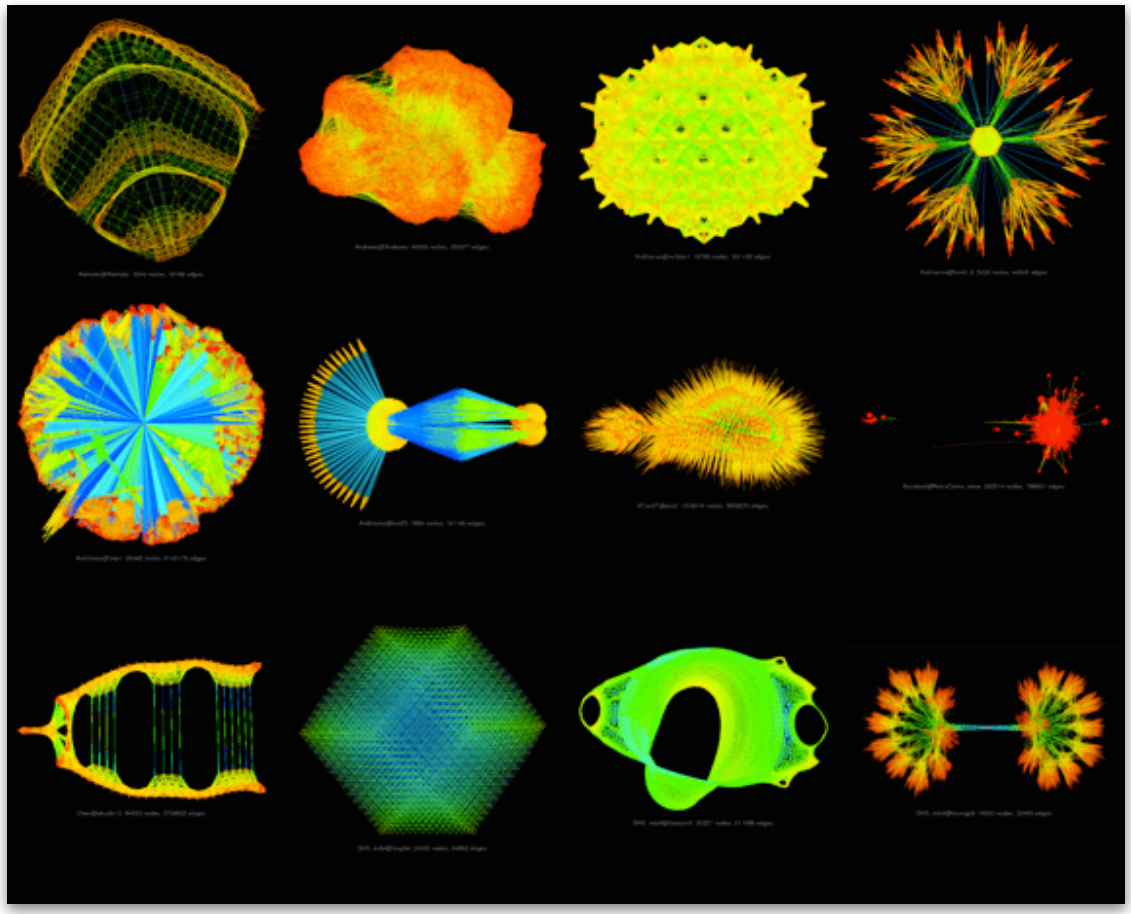
$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}$$

Therefore *algebraic graph theory* bridges between *linear algebra* and *graph theory*

Beauty in Sparsity

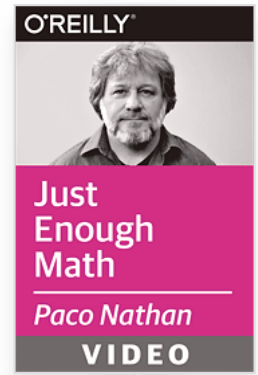
Sparse Matrix Collection… for when you **really** need a wide variety of sparse matrix examples, e.g., to evaluate new ML algorithms

SuiteSparse Matrix Collection
faculty.cse.tamu.edu/davis/matrices.html

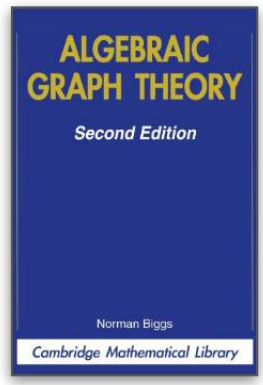


Resources

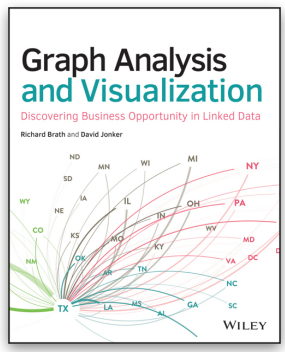
See examples in: **Just Enough Math**



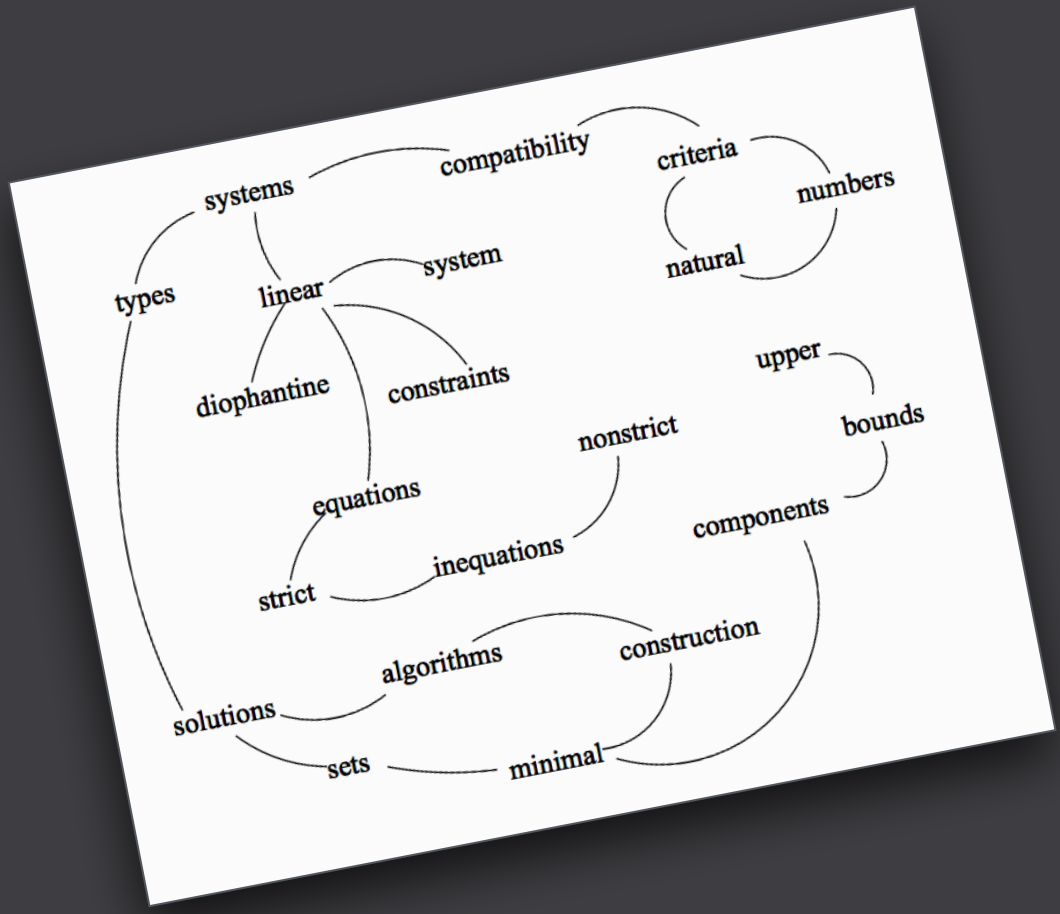
Algebraic Graph Theory
Norman Biggs
Cambridge (1974)
amazon.com/dp/0521458978



Graph Analysis and Visualization
Richard Brath, **David Jonker**
Wiley (2015)
shop.oreilly.com/product/9781118845844.do



TextRank


systems
compatibility
criteria
numbers
types
linear
system
natural
diophantine
constraints
upper
bounds
nonstrict
equations
components
inequations
strict
algorithms
construction
solutions
sets
minimal

TextRank: original paper

TextRank: Bringing Order into Texts

Rada Mihalcea*, *Paul Tarau

Conference on Empirical Methods in Natural Language Processing (July 2004)

https://goo.gl/AJnA76

http://web.eecs.umich.edu/~mihalcea/papers.html

http://www.cse.unt.edu/~tarau/

TextRank: other impl

Jeff Kubina (Perl / English):

http://search.cpan.org/~kubina/Text-Categorize-Textrank-0.51/lib/Text/Categorize/Textrank/En.pm

Paco Nathan (Hadoop / English+Spanish):

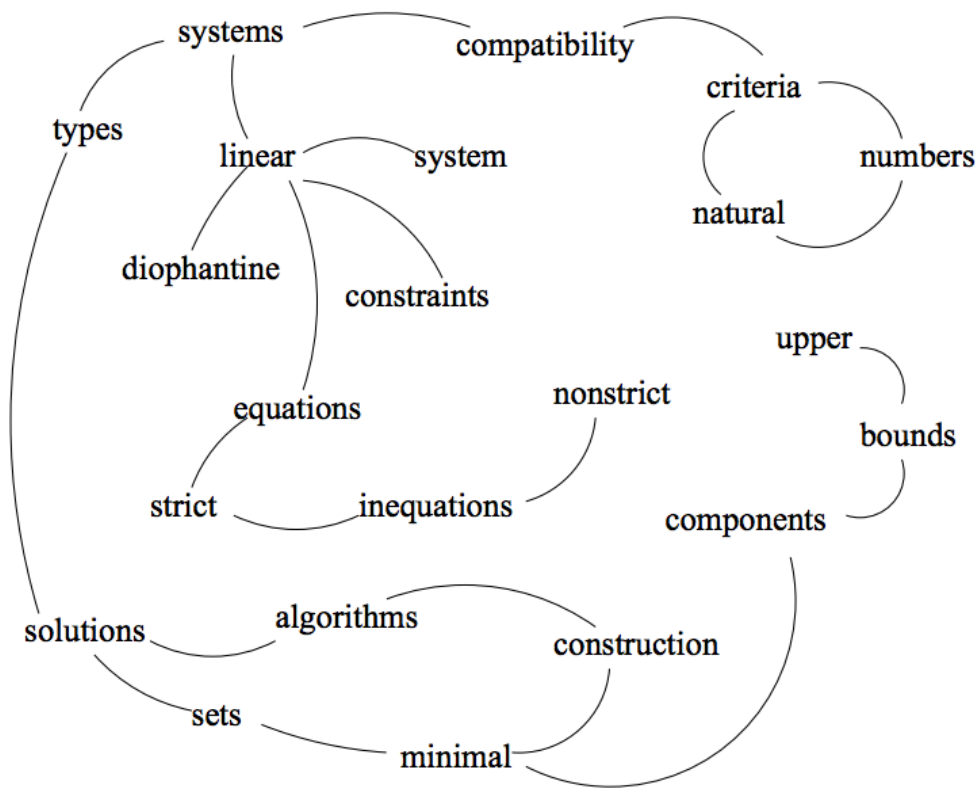
https://github.com/ceteri/textrank/

Karin Christiasen (Java / Icelandic):

https://github.com/karchr/icetextsum

TextRank: raw text input

Compatibility of systems of linear constraints over the set of natural numbers. Criteria of compatibility of a system of linear Diophantine equations, strict inequations, and nonstrict inequations are considered. Upper bounds for components of a minimal set of solutions and algorithms of construction of minimal generating sets of solutions for all types of systems are given. These criteria and the corresponding algorithms for constructing a minimal supporting set of solutions can be used in solving all the considered types systems and systems of mixed types.



TextRank: data results

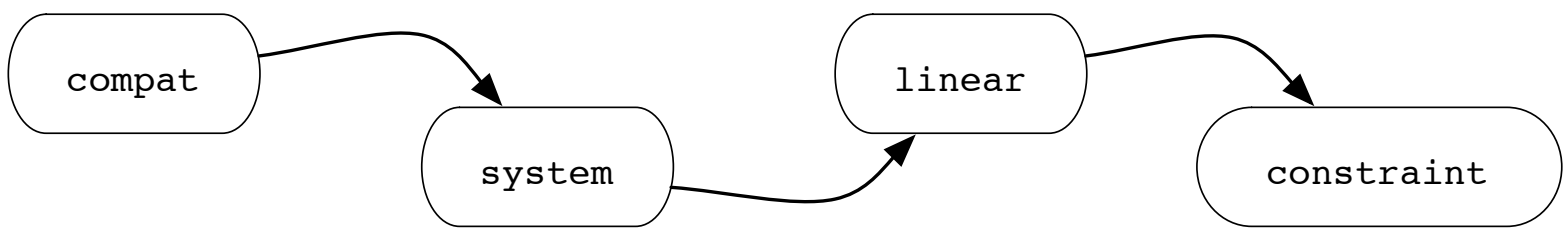
1:

```
"Compatibility of systems of linear constraints"
```

2:

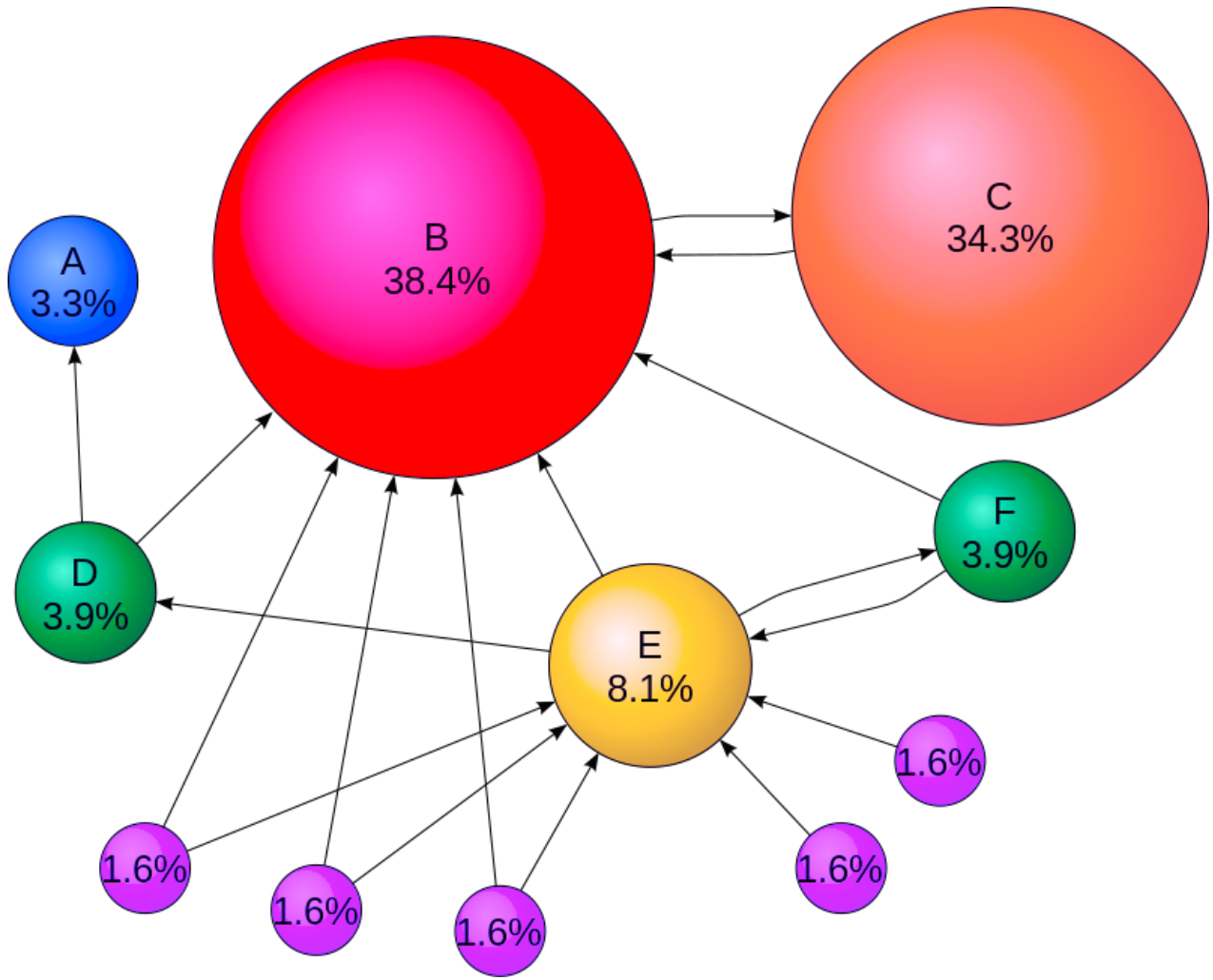
```
[{'index': 0, 'stem': 'compat', 'tag': 'NNP','word': 'compatibility'},
 {'index': 1, 'stem': 'of', 'tag': 'IN', 'word': 'of'},
 {'index': 2, 'stem': 'system', 'tag': 'NNS', 'word': 'systems'},
 {'index': 3, 'stem': 'of', 'tag': 'IN', 'word': 'of'},
 {'index': 4, 'stem': 'linear', 'tag': 'JJ', 'word': 'linear'},
 {'index': 5, 'stem': 'constraint', 'tag': 'NNS','word': 'constraints'}]
```

3:

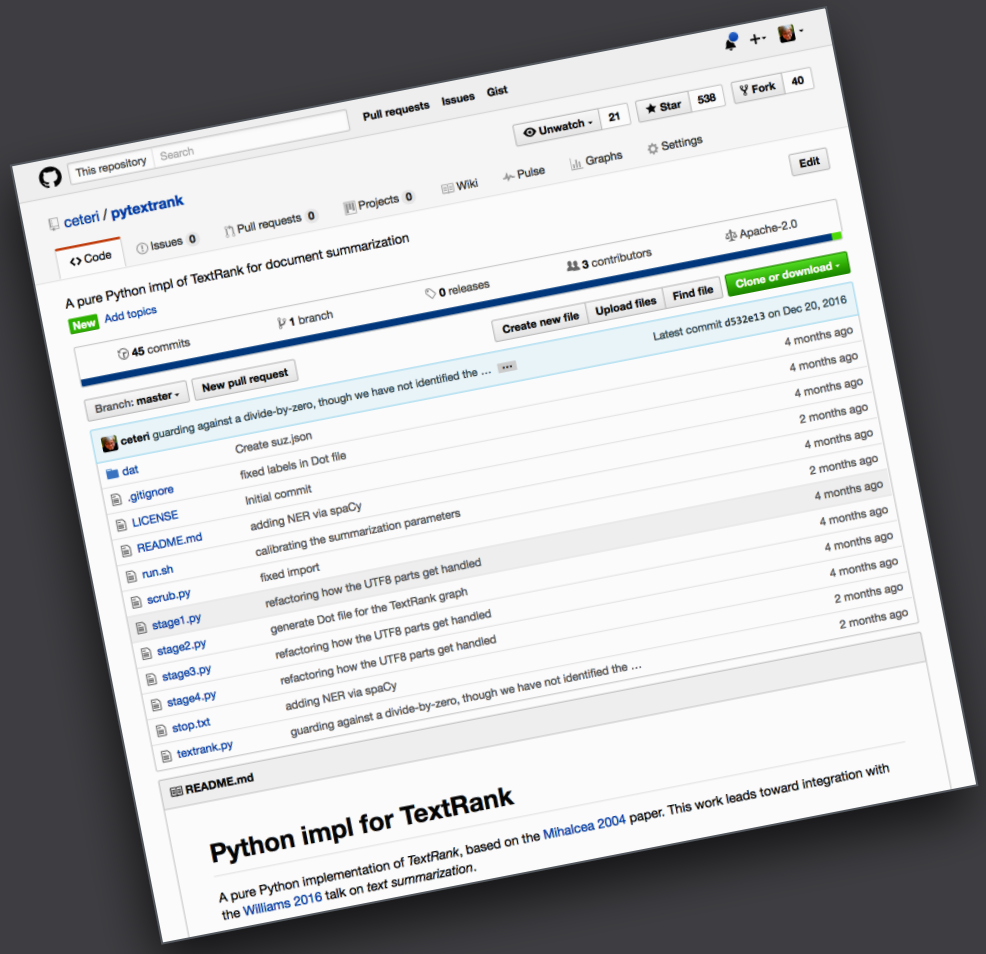


TextRank: how it works

https://en.wikipedia.org/wiki/PageRank



PyTextRank


ceteri / pytextrank
A pure Python impl of TextRank for document summarization
Python impl for TextRank
A pure Python implementation of *TextRank*, based on the Mihalcea 2004 paper. This work leads toward integration with the Williams 2016 talk on *text summarization*.

Motivations

A pure Python implementation of TextRank,
based on *Mihalcea 2004*

Because **monolithic NLP frameworks** which
control the entire pipeline (instead of APIs/services)
are wretched

Because **keywords**…
(note that *ngrams* aren't much better)

Using keywords leads to surprises!

The Journal of Alternative Facts 01 (2017) 01–20

JAF
THE JOURNAL OF alternative FACTS

The Journal of Alternative Facts

We Have All the Best Climates, Really, They're Great

Iwas A. Scientistonce *

* and now I have all my research approved by a public relations office

Abstract

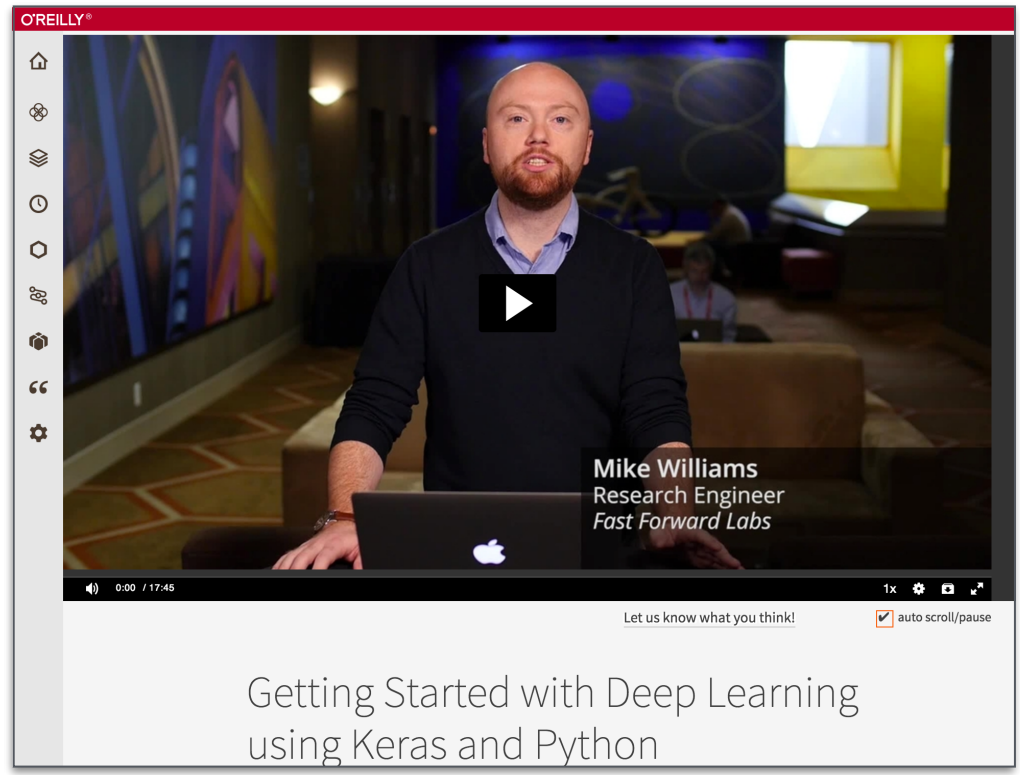
The research presented in this paper is really the best research that you will ever see. We have methods, the best methods, and we used them to study climate. As you may already know, the Earth, led by America, has all the best climates. In this paper we refute prior work by out-of-touch scientists who insist that the climate is changing – why would it change, when it's so great already? It is not getting warmer. In fact, our findings show that you were cold at least one day last year. Our (really fantastic) data also reveals that America has all the best CO2 levels, really great levels. In our discussion, we reveal that there is no reason to believe a bunch of scientists who spent all their time learning and studying "facts" instead of being out in the real world making jobs. Our alternative facts definitively prove that scientists are losers. Finally, we had peer reviews, by all the best people, our people, because politicians know the most about science, the very best things about science.

Keywords: climate, "data", "facts", #makeclimategreatagain, "science"

Motivations

Better to have **keyphrases** and **summaries**

Leads toward integration with the Williams 2016 talk on text summarization with deep learning: **http://mike.place/2016/summarization/**



Enhancements to TextRank

Modifications to the original Mihalcea algorithm include:

- fixed bug in the original paper's pseudocode;
  see **Java impl 2008** used by ShareThis, etc.
- uses **lemmatization** instead of stemming
- verbs included in graph (not in key phrase output)
- integrates **named entity resolution**
- keyphrase ranks used in **MinHash** to approximate **semantic similarity**, which produces summarizations
- allow use of **ontology**, i.e., AI knowledge graph to augment the parsing

Lemmatization vs. Stemming

Use of *stemming*, e.g., with **Porter Stemmer**, has long been a standard way to "normalize" text data: a computationally efficient approach to reduce words to their "stems" by removing inflections.

A better approach is to *lemmatize*, i.e., use part-of-speech tags to lookup the root for a word in WordNet – related to looking up its *synsets*, *hypernyms*, *hyponyms*, etc.

Lexeme	PoS	Stem	Lemma
interact	VB	interact	interact
comments	NNS	comment	comment
provides	VBZ	provid	provide

PytTextRank: repo + dependencies

https://github.com/ceteri/pytextrank

- **TextBlob** – a Python 2/3 library that provides a consistent API for leveraging other resources
- **WordNet** – think of it as somewhere between a large thesaurus and a database
- **spaCy**
- **NetworkX**
- **datasketch**
- **graphviz**

The Beyond

BTW, some really cool stuff to leverage, once you have ranked keyphrases as **feature vectors**:

- **Happening** – *semantic search*
  http://www.happening.technology/

- **DataRefiner** – *topological data analysis w/ DL*
  https://datarefiner.com/

O'Reilly Media conferences + training:

NLP in Python

repeated live online courses

Strata

SJ Mar 13-16

Deep Learning sessions, 2-day training

Strata+ Hadoop WORLD

Artificial Intelligence

NY Jun 26-29, SF Sep 17-20

SF CFP opens soon, follow @OreillyAI for updates

O'REILLY Artificial Intelligence CONFERENCE

JupyterCon

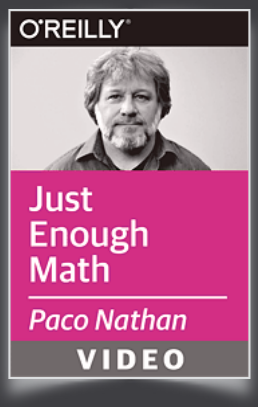
NY Aug 22-25

jupytercon

speaker:

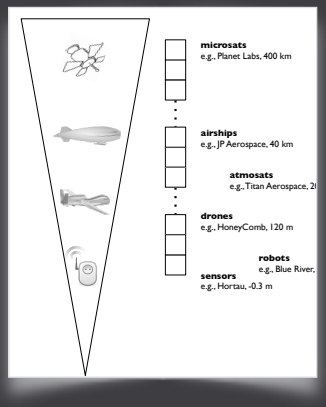
periodic newsletter for updates,
events, conf summaries, etc.:

liber118.com/pxn/
@pacoid

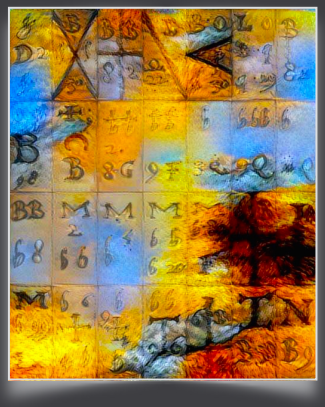


Just Enough Math

Building Data Science Teams



Ag + Data



Hylbert-Speys

How Do You Learn?