PyTextRank: Graph algorithms for enhanced natural language processing

Paco Nathan @pacoid
Dir, Learning Group @ O’Reilly Media
2017-09-28 / slides goo.gl/mwMyk4
Background: helping people learn
Research questions

- How do we personalize learning experiences, across ebooks, videos, conferences, computable content, live online courses, case studies, expert AMAs, etc.

- How do we help experts (by definition, really busy people) share their knowledge with peers in industry?

- How do we manage the role of editors at human scale, while technology and delivery media evolve rapidly?

- How do we help organizations learn and transform continuously?
UX for content discovery:
- partly generated + curated by humans
- partly generated + curated by AI apps
O’Reilly Media produces ~20 conferences per year, each with ~200 hours of video content.

To review in entirety, an acquisitions editor would have a finger permanently on the fast-forward button 10 months out of the year.

Introduction of AR/VR for learning materials accelerates that disparity.

Plus, that’s only one among several major needs for applying AI in Media – to augment staff.
content which can represented as text can be parsed by NLP, then manipulated by available AI tooling

labeled images get *really* interesting

text or images within a context have inherent structure

representation of that kind of structure is rare in the Media vertical – so far
Working with text and NLP

- parsing
- named entity recognition
- smarter indexing (semantic search)
- summarization, especially for video
- query expansion
- augmenting ontology
- semantic similarity for recommendations
- quicker development of assessments
- complexity estimates for learning materials
let’s take a look at a few examples often when people are first learning about Docker they try and put it in one of a few existing categories sometimes people think it’s a virtualization tool like VMware or virtualbox also known as a hypervisor these are tools which are emulating hardware for virtual software
Which parts do people or machines do best?

human scale
primary structure
control points
testability

machine generated data products
~80% of the graph

team goal: maintain structural correspondence between the layers
big win for AI: inferences across the graph
Ontology

Deep learning (deep structured learning) is a type of machine learning based on a set of algorithms using multiple processing layers, with or without nonlinear transformations. Deep learning has been successfully applied to neural networks.

About: Deep learning

An Entity Of Type: ComputerArchitecture

- Deep learning (deep structured learning) is a type of machine learning based on a set of algorithms using multiple processing layers, with or without nonlinear transformations. Deep learning has been successfully applied to neural networks.

Properties:
- dbp:abstract

- Deep learning (deep structured learning) is a type of machine learning based on a set of algorithms using multiple processing layers, with or without nonlinear transformations. Deep learning has been successfully applied to neural networks.
Motivations:
a new generation of NLP
Long, long ago, the field of NLP used to be concerned about transformational grammars, linguistic theory by Chomsky, etc.

Machine learning techniques allowed simpler approaches, i.e., statistical parsing. With the rise of Big Data, those techniques grew more effective – Google won the NIST 2005 Machine Translation Evaluation and remains a leader.

More recently, deep learning helps extend performance (read: less errors) for statistical parsing and downstream uses.
ML applied

Models allow NLP packages to:

- determine the natural language being used
- segment texts into paragraphs and sentences
- annotate words with part-of-speech
- find roots for each word (lemmas)
- chunk the noun phrases
- identify named entities
- correct spelling and grammar errors
- estimate scores for sentiment, readability, complexity
- etc.
Common misconceptions

- **generational lag**: that keyword analysis ("bag of words"), stemming, co-occurrence, LDA, and other techniques from a prior generation remain current best practices.

- **vendor FUD**: that NLP requires Big Data frameworks to run at scale.

- **either/or fallacy**: that NLP results feeding into AI apps either must be fully automated or require lots of manual intervention.
What changed?

**Hardware**: multicore and large memory spaces, plus wide availability of cloud computing resources.

The same data + algorithms which required a large Hadoop cluster in 2007, run faster in Python on my laptop in 2017.

NLP benefited as hardware advances eliminated the need for computational shortcuts (e.g. stemming) and made more advanced approaches (e.g., graph algorithms) become feasible.

Also, subtle shifts as approximation algorithms became more integrated, e.g., substituting n-grams with **skip-grams**.
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", #makelclimategreatagain, "science"
Continued use of the outdated NLP approaches (bag-of-words, stemming, n-grams, plus the general class of Apache Solr) increases risk for “fake news bots” problems.
Advances: graph algorithms for NLP
TextRank

“TextRank: Bringing Order into Texts”
Rada Mihalcea, Paul Tarau

https://goo.gl/AJnA76

http://web.eecs.umich.edu/~mihalcea/
http://www.cse.unt.edu/~tarau/
TextRank

“In this paper, we introduced TextRank – a graph-based ranking model for text processing, and show how it can be successfully used for natural language applications. In particular, we proposed and evaluated two innovative unsupervised approaches for keyword and sentence extraction, and showed that the accuracy achieved by TextRank in these applications is competitive with that of previously proposed state-of-the-art algorithms. An important aspect of TextRank is that it does not require deep linguistic knowledge, nor domain or language specific annotated corpora, which makes it highly portable to other domains, genres, or languages.”
TextRank

1. segment document into paragraphs and sentences
2. parse each sentence, find PoS annotations, lemmas
3. construct a graph where nouns, adjectives, verbs are vertices
   - links based on skip-grams
   - links based on repeated instances of the same root
4. stochastic link analysis (e.g., PageRank) identifies noun phrases which have high probability of inbound references
TextRank leverages NLP and graph algorithms at scale, improving means to integrate Deep Learning plus Knowledge Graphs – for enhanced machine intelligence in text handling.
Graph terminology

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

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.

Keywords assigned by TextRank:
- linear constraints
- linear diophantine equations
- natural numbers
- nonstrict inequations
- strict inequations
- upper bounds

Keywords assigned by human annotators:
- linear constraints
- linear diophantine equations
- minimal generating sets
- non-strict inequations
- set of natural numbers
- strict inequations
- upper bounds
By the numbers

1: "Compatibility of systems of linear constraints"

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

3: compat
   ▼
    ▼
     system
     ▲
     linear
     ▲
     constraint
Important outcomes

Unsupervised methods which do not require subject domain info, but can use it when available

Extracting key phrases from a text, in lieu of keywords, provides significantly more information content.

For example, consider “deep learning” vs. “deep” and “learning”.

Also, automated summarization can reduce human (editorial) time dramatically.
Implementations: various trade-offs
Other implementations

Paco Nathan, 2008
Java, Apache Hadoop / English+Spanish
https://github.com/ceteri/textrank/

Radim Řehůřek, 2009
Python, NumPy, SciPy
https://radimrehurek.com/gensim/

Jeff Kubina, 2009
Perl / English
http://search.cpan.org/~kubina/Text-Categorize-Textrank-0.51/lib/Text/Categorize/Textrank/En.pm

Karin Christiansen, 2014
Java / Icelandic
https://github.com/karchr/icetextsum

Paco Nathan, 2014
Scala, Apache Spark / English
https://github.com/ceteri/spark-exercises

Got any others to add to this list??
PyTextRank

Paco Nathan, 2016

Python implementation atop spaCy, NetworkX, datasketch:

- https://pypi.python.org/pypi/pytextrank/
- https://github.com/ceteri/pytextrank/
PyTextRank: motivations

- **spaCy** is moving fast and benefits greatly by having an abstraction layer above it
- **NetworkX** allows use of multicore + large memory spaces for graph algorithms, obviating the need in many NLP use cases for Hadoop, Spark, etc.
- **datasketch** leverages approximation algorithms which make key features feasible, given the environments described above
- monolithic NLP frameworks often attempt to control the entire pipeline (instead of using modules or services) and are basically **tech debt waiting to happen**
PyTextRank: innovations

- fixed bug in the original paper’s pseudocode; see Java impl 2008 used in production by ShareThis, Rolling Stone, etc.
- included verbs into the graph (though not in keyphrase output)
- added text scrubbing, unicode normalization
- replaced stemming with lemmatization
- can leverage named entity recognition, noun phrase chunking
- provides extractive summarization
- makes use of approximation algorithms, where appropriate
- applies use of ontology: random walk of hypernyms/hyponyms helps enrich the graph, which in turn improves performance
- integrates vector embedding + other graph algorithms to help construct or augment ontology from a corpus
PyTextRank: summarization

1. parse a text to create its feature vector, i.e., the ranked keyphrases from TextRank
2. calculate semantic similarity between the feature vector and each sentence in the text
3. select the N top-ranked sentences, listed in order, to summarize

oreilly.com/ideas/the-current-state-of-machine-intelligence-3-0

Almost a year ago, we published our now-annual landscape of machine intelligence companies, and goodness have we seen a lot of activity since then. As has been the case for the last couple of years, our fund still obsesses over 'problem first' machine intelligence -- we've invested in 35 machine intelligence companies solving 35 meaningful problems in areas from security to recruiting to software development. At the same time, the hype around machine intelligence methods continues to grow: the words 'deep learning' now equally represent a series of meaningful breakthroughs (wonderful) but also a hyped phrase like 'big data' (not so good!). What's the biggest change in the last year?
NLP + AI, by the numbers

1. pull data across various silos by federating queries
2. scrub text; normalize unicode
3. synthesize missing tables; impute missing values for metadata; parse/repair TOCs
4. extract document text from DOM + metadata; segment sentences;
5. parse to get PoS tagging; lemmatize nouns and verbs; noun phrase chunking; named entity recognition
6. construct a graph for each document
7. enrich graph using ontology to add links among lemmas, named entities, etc.
8. build feature vectors with TextRank
9. use feature vectors + semantic similarity for extractive summarization
10. use feature vectors + vector embedding to expand queries and augment ontology
11. apply various graph algorithms as high-pass filters
12. build priors based on authors, publishers, titles, publication dates
13. disambiguate topic annotations using active learning
14. extract key features from annotation models for subsequent graph enrichment
15. use classifiers to predict language and complexity
16. build in-memory semantic indexing, auto-completion, content recommendations, gap analysis, etc.
17. run monte carlo methods to rank topics by usage
18. rinse, lather, repeat
Looking ahead
O’Reilly Media is actively working with other like-minded organizations to drive machine intelligence use cases in Media, through open source
PyTextRank: upcoming

Package installation is currently integrated with PyPi

2017-Q4: adding support for Anaconda

Also working on:

- more efficient integration with spaCy, e.g., spans
- more coding examples in the wiki, especially using Jupyter
- support for ReadTheDocs
- better handling for character encoding issues; see “Character Filtering” by Daniel Tunkelang, along with his entire series Query Understanding
- more integration into coursework; see Get Started with NLP in Python on O’Reilly
PyTextRank: upcoming

One compelling reason for using TextRank is to leverage graph algorithms.

Using an ontology to enrich the graph (prior to the “PageRank” step) has been shown to improve NLP results – thus leveraging graph algorithms.

A major upcoming feature in PyTextRank will take an ontology (e.g., SKOS) as an input parameter.

Working now to train/focus spaCy NER using an input ontology, as an automated preprocessing step for TextRank.
PyTextRank: upcoming

Disambiguating context is a hard problem in NLP, and relatively lacking in the available open source packages. For example: “ios operating system” – Apple or Cisco?

O’Reilly Media leverage a human-in-the-loop design pattern to build disambiguation pipelines through active learning, which in turn uses and helps extend our ontology work.
PyTextRank: upcoming

Working on improved summarization:

- currently provides *extractive summarization*
- working toward *abstractive summarization*; evaluating integration with *TensorFlow*

See also:

- [Text Summarization in Python](#), 2017-04-05
- [Text summarization, topic models and RNNs](#), 2016-09-25

Overall, we’re focused on improving work with video for O’Reilly Media use cases, e.g., parsing and repairing automated transcriptions
Strata Data
- SG, Dec 4-7
- SJ, Mar 5-8
- UK, May 21-24
- CN, Jul 12-15

The AI Conf
- CN Apr 10-13
- NY, Apr 29-May 2
- SF, Sep 4-7
- UK, Oct 8-11

JupyterCon
- NY, Aug 21-24

OSCON (returns!)
- PDX, Jul 16-19, 2018
updates, reviews, conference summaries...

liber118.com/pxn/ @pacoid