AI within O’Reilly Media

Paco Nathan  @pacoid
Director, Learning Group @ O’Reilly Media

Strata SG, Singapore  2017-12-07
AI: why now?
Big Picture: Three Factors

- **Big Data**: *machine data* (1997-ish)
- **Big Compute**: *cloud computing* (2006-ish)
- **Big Models**: *deep learning, etc.* (2009-ish)

The confluence of three factors created a business environment where AI could become mainstream
Benchmark: achieving human parity

2016-10-12: Microsoft researchers reach human parity in conversational speech recognition

Achieving Human Parity in Conversational Speech Recognition
W. Xiong, et al. Microsoft
“AI is the new electricity”

Andrew Ng, AI SF 2017 keynote


(especially beginning at 11:38)

- creating virtuous cycles for acquiring data at scale
- monetizing that data through other business lines
Flywheel Effect, circa 2017

AI drives features in products and services ... which in turn drives cloud consumption ... which in turn acquires even more data ... particularly with mobile products

Consequently a small set of firms now leads in AI, cloud, mobile: Google, Amazon, Microsoft, Baidu, etc.
AI: resources
AI: Trends in 2017

Ben Lorica
Chief Data Scientist, O’Reilly Media

How companies can navigate the age of machine learning

oreilly.com/ideas/how-companies-can-navigate-the-age-of-machine-learning

To become a “machine learning company,” you need tools and processes to overcome challenges in data, engineering, and models
Factors in Adoption

- How automatable?
- Strong/weak ICT base
- ROI / payback period
- Labor costs / margin pressure
- Competition
Consensus among major consulting firms:

Enormous upside from AI, across verticals. However, to be in the game, an organization must already have Big Data infrastructure and related practices in place:

1. cloud and SRE
2. eliminating data silos
3. cleaning data / repairing metadata
4. embracing contemporary data science

Those are prerequisites, there are no short cuts to AI. Plus, there’s an ongoing talent crunch.

– Strata 2017 Exec Briefings
2017 highlights from leading teams

- **TensorFlow, Machine Learning, and Learning to Learn**
  Sherry Moore  Google

- **Distributed deep learning on AWS using MXNet**
  Anima Anandkumar  Amazon

- **Scalable operationalization of trained CNTK and TensorFlow DNNs**
  Mary Wahl  Microsoft
2017 highlights

- scale up to solve complex problems (big models); then reduce to deploy consumer products (low power)
- neuroevolution as a dramatically faster path for training
- many use cases for reinforcement learning
- adversarial machine learning
- custom hardware (TPUs, IPUs, etc.) confronting the power limits (~300W)
- inverse reward design: making AI blend in with human norms for collaboration
- AI in manufacturing
- a need to leverage fitness functions in lieu of objective functions
NLP Renaissance

- Previous generation of NLP now superseded, because of ML + cloud
- NLP underpinnings for current innovation – current work on spaCy, Stanford Parser, etc.
- vector embedding use cases
- auto-summarization use cases
- natural language understanding
- natural language generation
Systems that learn in uncertain domains

Peter Norvig  Google – Software engineering for AI at scale:

- difficult to debug, revise incrementally, or verify
- less transparency into algorithms
- components are hard to isolate, for debugging
- automated integration introduces unusual risks
- tech debt accumulates more readily

Related:
Machine Learning: The High Interest Credit Card of Technical Debt
D. Sculley, et al.  Google
Systems that learn in uncertain domains
Why are AI Programs Different?

Peter Norvig  Google – [Ai in the software engineering workflow](#)

- Content: Models not programs
- Process: Training not debugging
- Release: Retraining not patching
- Uncertainty: of objective
- Uncertainty: of action/recommendation
- Uncertainty: propagates through model
AI: hands-on coding tutorials

Getting Started with Deep Learning using Keras and Python
Mike Williams

Generative Adversarial Networks for Beginners
Jon Bruner

Hello, TensorFlow!
Aaron Schumacher

Probabilistic data structures in Python
Paco Nathan

Genetic Algorithms: A Gentle Introduction
Safia Abdalla

Geolocated clustering and prediction services with scikit-learn
Natalino Busa

Probabilistic Programming from Scratch
Mike Williams
AI: resources
AI: events

The AI Conf:
Beijing, New York, San Francisco, London

A Practical Introduction to Machine Learning
Matt Kirk

Get Started in NLP in Python
Paco Nathan

Deep Learning for NLP
John Krohn

Introduction to TensorFlow
Michael Li, Data Mastropole, Robert Schroll
Background: helping people learn
Safari is how you 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 knowledge with their 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
AI in Media

- 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
Transcript: 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

Confidence: 0.973419129848
O’Reilly Media produces ~20 conferences per year, each with ~200 hours of video content.

To review those video products, an editor must have a finger permanently on the fast-forward button for 10 months out of the year.

Introduction of AR/VR for learning materials heightens that need.

That’s only one among several major needs for applying AI in Media, to augment staff.
AI in Media: tools, projects, use cases
Knowledge Graph

- used to construct an ontology about technology, based on learning materials from 200+ publishers
- uses SKOS as a foundation, ties into US Library of Congress and DBpedia as upper ontologies
- primary structure is “human scale”, used as control points
- majority (>90%) of the graph comes from machine generated data products
Knowledge Graph

About: Deep learning

Deep learning is a type of machine learning based on a set of algorithms using multiple processing layers, with the goal of using transformations. Deep learning has been used in neural networks.

- **Property**: Deep learning
- **Value**: Machine learning using multiple processing layers, with the goal of using transformations.

Deep learning algorithms learn representations that improve accuracy by using intensity values to represent complex concepts. These representations are often more efficient than traditional expression recognition algorithms, which rely on features with efficient extraction. Research in deep learning representations has been driven by neuroscience and the idea of a brain-like system and associated learning techniques. Deep neural networks are used in computer vision, natural language processing, bioinformatics, and deep learning has been highly influential in these areas.
Knowledge Graph

- ontology provides context which Deep Learning lacks
- aka, “knowledge graph” – a computable thesaurus
- maps the semantics of business relationships
- S/V/O: “nouns”, some “verbs”, a few “adjectives”
- difficult work, a relatively expensive investment, potentially high ROI
- conversational interfaces (e.g., Google APIs) improve UX by importing controlled vocabularies
Which parts do people or machines do best?

**human scale**
primary structure
control points
testability

**team goal:** maintain structural correspondence between the layers

**big win for AI:** inferences across the graph

>90% of the graph
Video transcription

- AI services in the cloud transcribe audio
- ontology provides domain-specific “hints”
- results are comparable with human transcription services, at least for use in NLP
- speeds development of *formative assessment*, and other text-based use of video content
- costs pennies on the dollar, fast turnaround
- otherwise, video is too time-intensive for editorial staff
Semantic similarity

Semantic similarity between two texts can be measured using a Jaccard metric, approximated using MinHash and LSH.

Used to construct a graph of the overall content, i.e., showing structure.

Probabilistic Data Structures in Python
Components

- **rdflib + NetworkX**: ontology graph represented as N3 “turtle”
- **PyTextRank**: NLP parsing, feature vectors, summarization
- **Jupyter + nbtransom**: human-in-the-loop ML pipelines
- **Apache Spark**: sort, partitioning, task management
- **scikit-learn**: machine learning models
- **gensim**: vector embedding
- **datasketch**: approximation algorithms
- **TensorFlow**: abstractive summarization
- **Flask, React**: microservices, UI web components
- **Redis**: in-memory indexing, full-text search
PyTextRank

TextRank ([R Mihalcea, P Tarau, 2004](#)) a graph algorithm that extracts key phrases and summarizes texts – for NLP which is improved over use of keywords, n-grams, etc.

- construct a graph from a paragraph of text
- run link analysis on that graph
- extract the highly ranked phrases

Python implementation atop spaCy, NetworkX, datasketch:
- [https://pypi.python.org/pypi/pytextrank/](https://pypi.python.org/pypi/pytextrank/)
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 of systems and systems of mixed types.
Working with text and NLP

- parsing
- named entity recognition
- smarter indexing (semantic search)
- summarization (especially video)
- vector embedding
- query expansion
- semantic similarity for recommendations
- speeds the development of assessments
- content complexity estimates
- amending ontology
Disambiguating contexts

Suppose someone publishes a book which uses the term `IOS`: are they talking about an operating system for an Apple iPhone, or about an operating system for a Cisco router?

We handle lots of content about both. Disambiguating those contexts is important for good UX in **personalized learning**.

In other words, how do machines help people distinguish that content within search?

Potentially a good case for deep learning, **except for the lack of labeled data at scale.**
Active learning through Jupyter

Jupyter notebooks are used to manage ML pipelines for disambiguation, where machines and people collaborate:

- ML based on examples – most all of the feature engineering, model parameters, etc., has been automated
- [https://github.com/ceteri/nbtransom](https://github.com/ceteri/nbtransom)
- based on use of [nbformat], [pandas], [scikit-learn]
Active learning through Jupyter

Jupyter notebooks are used to manage ML pipelines and people collaborate:

- ML based on examples – most all of the feature engineering, model parameters, etc., has been automated
- https://github.com/ceteri/nbtransom
- based on use of Jupyter notebooks

Jupyter notebook as...

- one part configuration file
- one part data sample
- one part structured log
- one part data visualization tool

plus, subsequent data mining of these notebooks helps augment our ontology
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
“Transfer Learning” via graph inference
Summary:
people and machines at work
Summary

The “game” is not to replace people – instead it is about leveraging AI to augment staff, so that organizations can retain people with valuable domain expertise, making their contributions and experience even more vital.

This is a personal opinion, which does not necessarily reflect the views of my employer.

However, the views of my employer...
Why we’ll never run out of jobs
Acknowledgements

Many thanks to all of the contributors on this work:

John Allwine, Martin Bravo, Niki Gitinabard, Chris Guidry, Asif Hasan, Taylor Martin, Colin Megill, Scott Murray, Andrew Odewahn, Eugene Panaitov, Eszti Schoell, Ningyu Zhang
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
- PDX, Jul 16-19, 2018
updates, reviews, conference summaries...

liber118.com/pxn/@pacoid