

Machine Learning for Automated Keyword Tagging

Using Supervised Learning

Anthony Buonomo (JHUAPL), Brian Thomas (NASA), Justin Gosses (SAIC), Yulan Lin (Google)

Outline

1. Overview
2. Difficulties with manual tagging
3. Demo
4. Design and development
5. Value and applications
6. How to use it today

Overview

- Worked with Scientific and Technical Information (STI)
- Used NASA Technical Reports Server (NTRS) documents
- Designed to help content managers fill in blindspots



Salient
Mars
Volcano
Mars volcano

Implicit
Rock
Mars surface
Planetary geology

Manual tagging is difficult to scale

Problem: Apply 7,000 concept tags to 1 million documents.

Manual

10 people, 700 tags / 2 minutes

40 hr / wk, 52 wk / yr

16 years

\$8 million

Our Concept Tagging Tool

1 machine, 70,000 tags / sec

24 hr/day

2 days

\$5 in compute and storage

Manual tagging is difficult to standardize

Manual

- STI Thesaurus: 20k+ concepts
- Typical native English speaker's vocabulary between about 40k and 70k
- A lot for one person to handle and uniformly apply

Our Concept Tagging Tool

- Average the knowledge across content managers.
- More consistent, less prone to under-tagging

Tool Specs

- Trained on NTRS - 3.5 million abstracts
- 7,000+ concepts
- 11 topic domains
- Batch processing → tag 10 documents per second
- Could easily make this faster because parallelizable

Demo

<http://go.nasa.gov/concepttagger>

The workshop will be organized as a combination of keynote addresses, invited speakers, short talks and posters centered around specific themes and topics of interest. These are as follows:

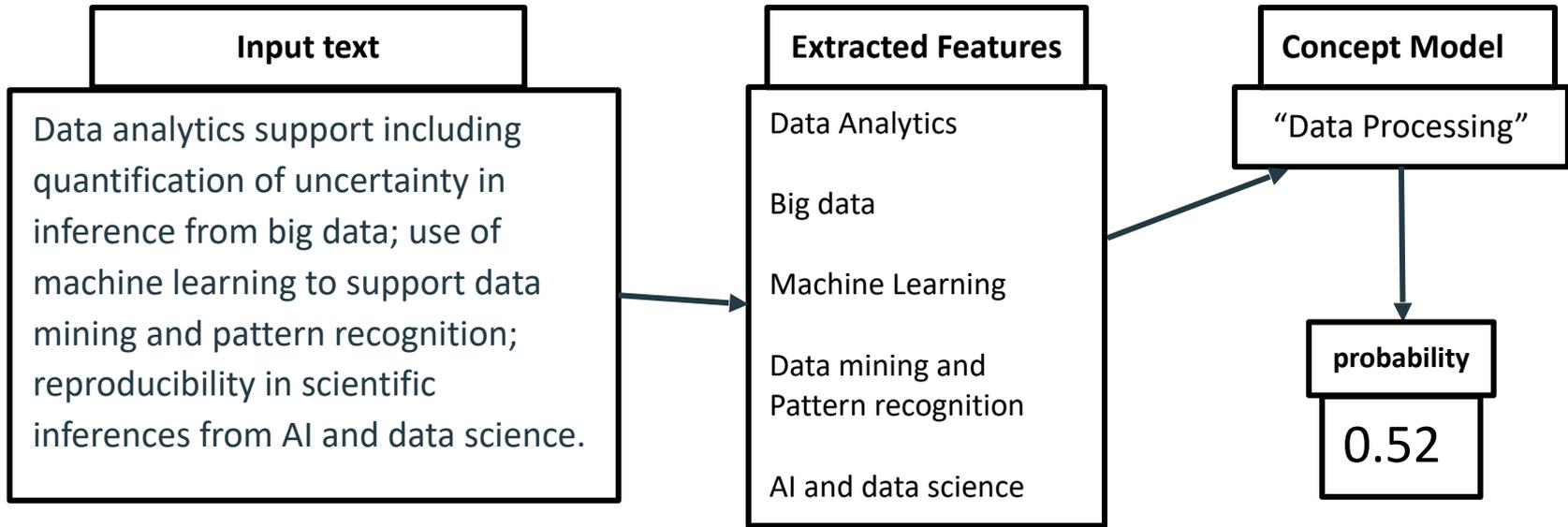
1. **Data-Driven Science** – Applications of AI and data science methodologies applied to enable science research at NASA. Support for working with observational data and model output. Data analytics support including i) quantification of uncertainty in inference from big data; ii) use of machine learning to support data mining and pattern recognition; iii) reproducibility in scientific inferences from AI and data science.
2. **AI in Engineering** – Applications of AI and data science methodologies applied to support NASA engineering applications across science, human exploration, and aeronautics. Use of AI methods for simulation, design, and operations.
3. **Autonomy** – Use of AI and data science to enable automated mission operations; onboard application of AI and data science to support autonomous missions including navigation, operations, and science.
4. **Cross-cutting AI and Data Science activities at NASA** – General AI methods and projects applicable to multiple NASA programs including machine learning, image analysis, natural language processing, etc.
5. **Cross-Agency AI and Data Science activities** – AI and data science collaborations across federal agencies
6. **Emerging Research Topics in AI; Collaborations with Academia** – AI and data science collaborations between NASA and academia. Training and educating next generation workforces in AI and data science. Research topics in AI and data science including trust, reproducibility, explainability, human AI interaction, etc.
7. **Real-World Applications of AI and Data Science** – NASA deployed applications in AI and data science.

We look forward to you joining us!

**System finds *concepts*
which need not appear in the text.**

Found concept	probability
Pattern recognition	0.99
education	0.95
Machine learning	0.95
Artificial intelligence	0.89
Data mining	0.78
Data processing	0.76
simulation	0.65
Computerized simulation	0.64
Image processing	0.63
Image analysis	0.63
Data simulation	0.61
Research management	0.54

Example with one concept



Execute with 7,000 + concepts in parallel

Thing we tried	Problem	Where we landed
Pre-trained language models for topic domains	Prediction execution too slow, marginal ROC-AUC increase	Logistic Regression with Stochastic Gradient Descent
Multilabel training	Slow training (could not parallelize) and did not help accuracy	Binary models
Evolution algorithms for hyper-parameter tuning	Pointed us toward the right parameters, but not used in final implementation because training too slow	A Constrained Grid Search over parameters
NLTK part of speech feature extraction	Slow feature extraction	spaCy Part of Speech and Named Entity Recognition

Tool Value

- Tags more documents faster
- Captures the knowledge of the content managers
 - Need humans because semantic drift, edge cases, new concepts
- Scalable for consistent tagging across many document collections
- Cost effective to accurately tagging docs

Applications of the Concept Tagging Tool

- STI tagging workflow
- <https://code.nasa.gov>
- Insight data management platform
- NASA Lessons Learned System
- Wordpress extension

How can I start using it now?

- Test it here: <http://go.nasa.gov/concepttagger>
- Run your own instance by:
 - Getting code from here: <https://github.com/nasa/concept-tagging-api>
 - Getting models from here:
 - https://data.nasa.gov/docs/datasets/public/concept_tagging_models/10_23_2019.zip
 - Each request has overhead cost. Want speed? Use batches of ~500.
- Train your own models with code here: <https://github.com/nasa/concept-tagging-training>

Just google “NASA concept tagging api”
and you should see the github repo.

Links

- API: <http://go.nasa.gov/concepttagger>
- Training code: <https://github.com/nasa/concept-tagging-training>
- API code: <https://github.com/nasa/concept-tagging-api>
- Federal Data Strategy Proofpoint: <https://strategy.data.gov/proof-points/2019/05/28/improving-data-access-and-data-management-artificial-intelligence-generated-metadata-tags-at-nasa/>

Contact

Anthony Buonomo - anthony.buonomo@jhuapl.edu

Justin Gosses - justin.c.gosses@nasa.gov

Brian Thomas - brian.a.thomas@nasa.gov

Thank you!

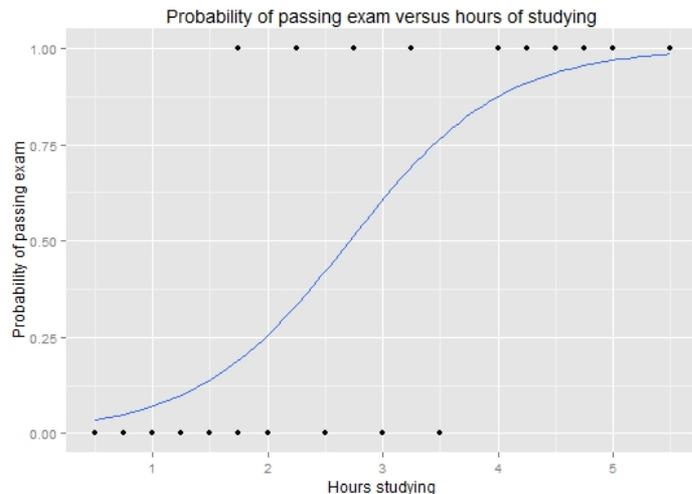
Backup Slides

Design: feature selection

- abstract and keywords min occurrence: 100 time
- [spaCy](#) language model max occurrence: <60% of docs
 - noun
 - noun phrases
 - acronyms Stricter thresholds
 - entities → smaller, faster models

Design: model training

- In 500+ documents in corpus
- Independent binary classifiers
- Logistic regression
 - stochastic gradient descent
- Train quickly and in parallel



Uses

- Rigorous standards? → fill in blindspots
 - Content manager assistant. Feedback to improve models.
- No capacity for manual → automatically tag
- Packaging some of the experts' knowledge
- Downstream NLP tasks

Design: challenges

- Lots of data → ~3.5 million abstracts
- Lots of keywords → ~7,000
- Imbalanced data → base rate of 1/10,000 or 1/1,000
- Many domains, some overlapping

Design: model training - parameters

- Large parameter space
 - Grid search → too time consuming
 - Evolutionary algorithms → still time consuming
 - Constrained grid search → good balance
- Imbalanced → add weight to minority class

Design: model training - domains

- Train topic classifiers
- Train keywords w.r.t different topics
- Choose district domain or no

