The application life cycle

What are the steps to releasing a ML application?

Naive ML Pipeline

Better Pipeline

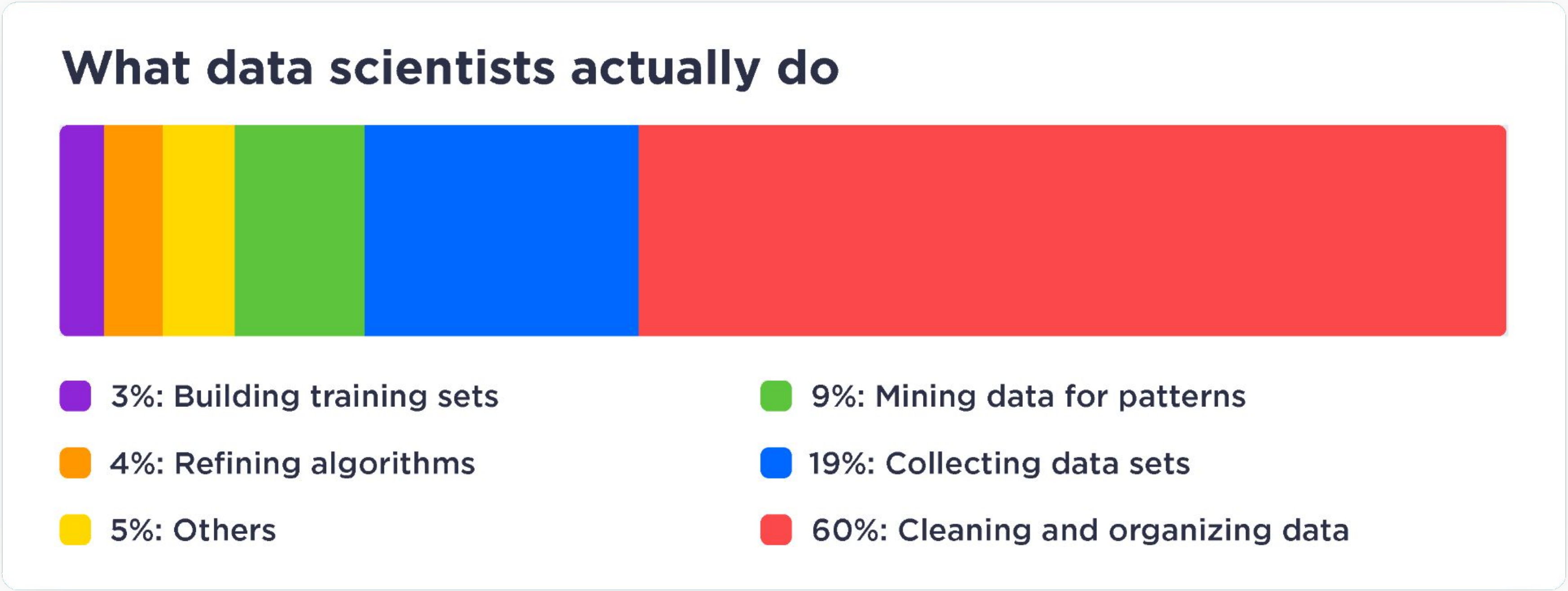
Collect Data
Collect and clean data
Annotate data if necessary

Modelling
Choose or design an appropriate model and objective function
Train model and monitor performance during training

Monitor in production
Monitor customer reaction, check for biases, distribution shift, unexpected behavior

Release to production
Set up data pipeline and compute resources, design user interface, communicate to users the capabilities and limitations of the system

Evaluate
Define evaluation metrics and check for biases
Conduct human evaluation



https://cldcvr.com/news-and-media/blog/clean-data-the-foundation-of-effective-machine-learning/


Who wants
clean data?
Who wants to
CLEAN the data?

Let’s walk through an example

TLDR: Extreme Summarization of Scientific Documents

Isabel Cachola[†] **Kyle Lo**[†] **Arman Cohan**[†] **Daniel S. Weld**[†‡]

[†]Allen Institute for AI

[‡]Paul G. Allen School of Computer Science & Engineering, University of Washington

{isabelc,kylel,armanc,danw}@allenai.org

Abstract

We introduce TLDR generation, a new form of extreme summarization, for scientific papers. TLDR generation involves high source compression and requires expert background knowledge and understanding of complex domain-specific language. To facilitate study on this task, we introduce SciTLDR, a new multi-target dataset of 5.4K TLDRs over 3.2K papers. SciTLDR contains both author-written and expert-derived TLDRs, where the latter are collected using a novel annotation protocol that produces high-quality summaries while minimizing annotation burden. We propose CATTS, a simple yet effective learning strategy for generating TLDRs that exploits titles as an auxiliary training signal. CATTS improves upon strong baselines under both automated metrics and human evaluations. Data and code are publicly available at

Abstract While many approaches to make neural networks more fathomable have been proposed, they are restricted to interrogating the network with input data. [...] In this work, we propose neural persistence, a complexity measure for neural network architectures based on topological data analysis on weighted stratified graphs. [...]

Intro [...] In this work, we present the following contributions: We introduce neural persistence, a novel measure for characterizing the structural complexity of neural networks that can be efficiently computed. [...]

Conclusion [...] However, this did not yield an early stopping measure because it was never triggered, thereby suggesting that neural persistence captures salient information that would otherwise be hidden among all the weights of a network [...]

TLDR We develop a new topological complexity measure for deep neural networks and demonstrate that it captures their salient properties.

Figure 1: An example TLDR of a scientific paper. A TLDR is typically composed of salient information (indicated by colored spans) found in the abstract, intro, and conclusion sections of a paper.

Scientific TLDR Generation

Goal: Generate "TLDRs" or extremely short summaries for scientific papers

Input: Text of scientific papers

Output: One sentence or less summaries of papers

Why? Help scientists parse large numbers of papers quickly

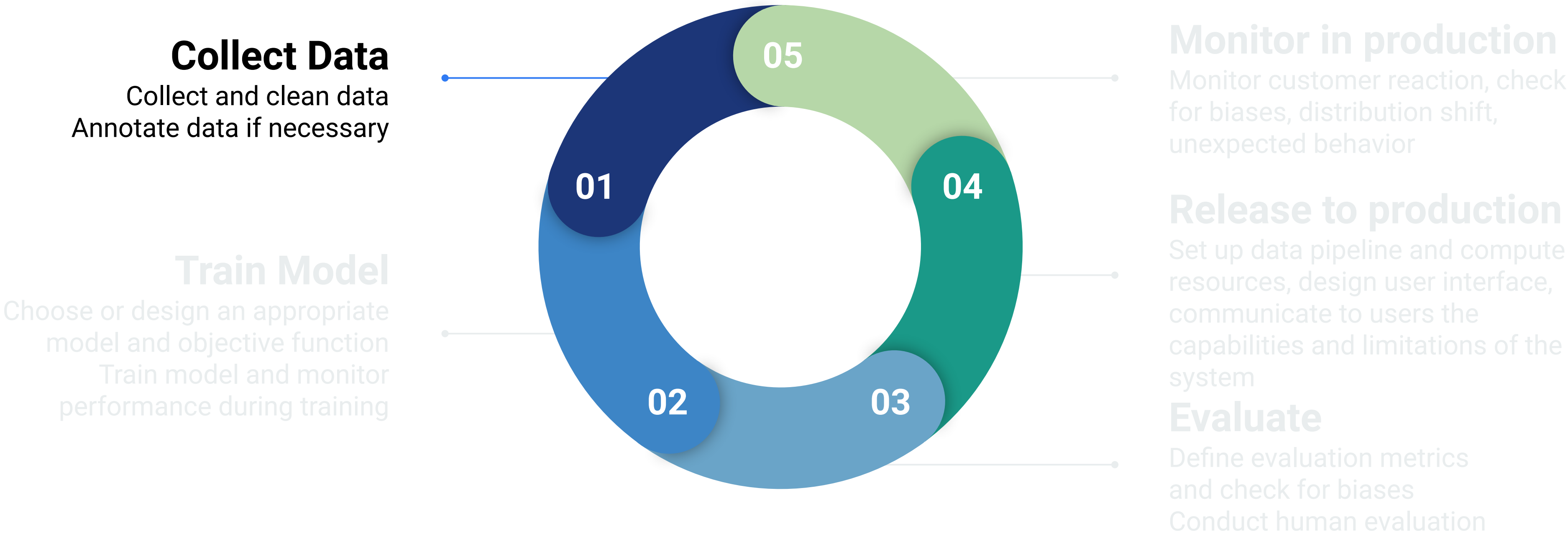
Collect Data
Collect and clean data
Annotate data if necessary

Train Model
Choose or design an appropriate model and objective function
Train model and monitor performance during training

Evaluate
Define evaluation metrics and check for biases
Conduct human evaluation

Release to production
Set up data pipeline and compute resources, design user interface, communicate to users the capabilities and limitations of the system

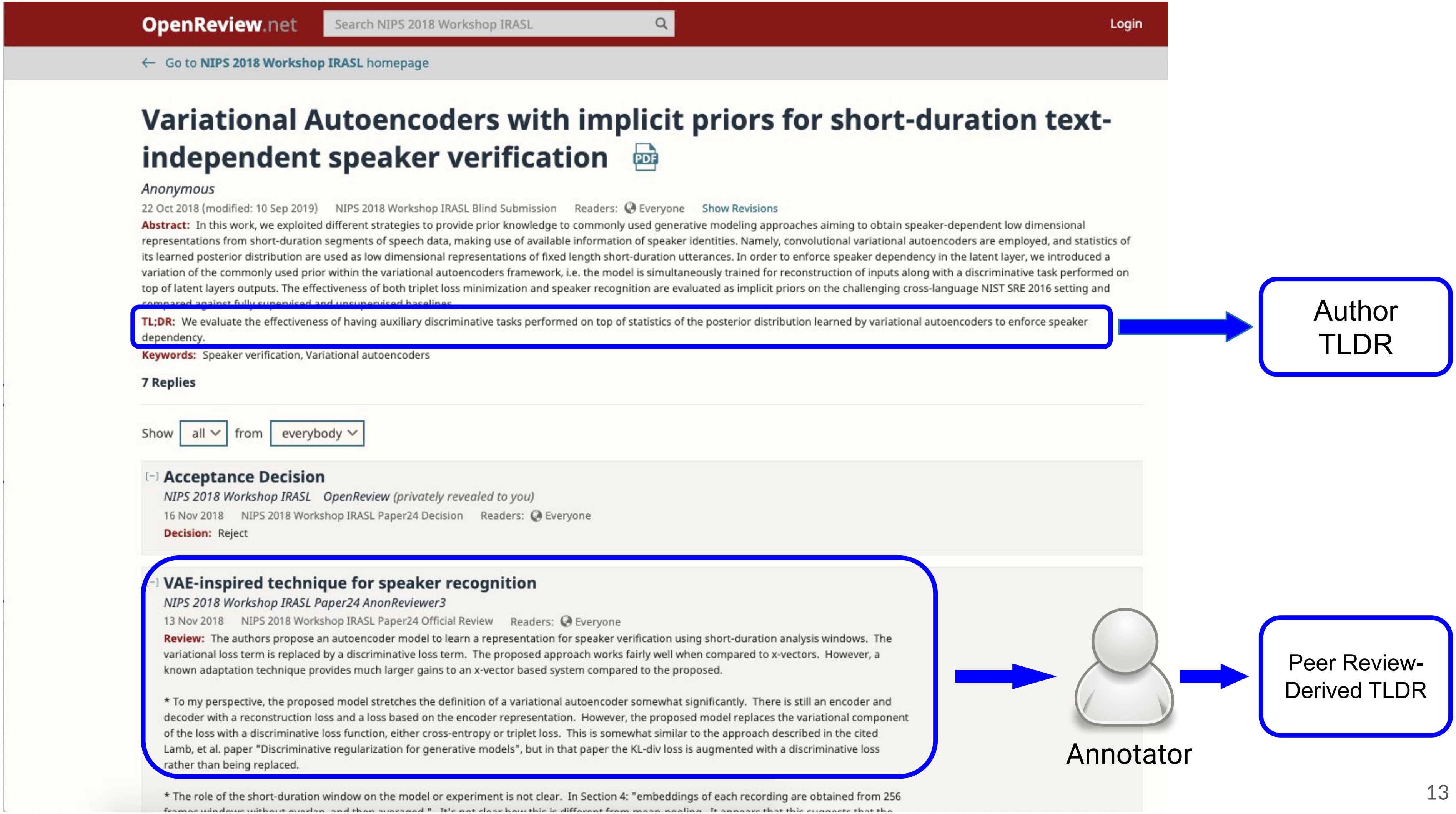
Monitor in production
Monitor customer reaction, check for biases, distribution shift, unexpected behavior



Collect Data
Collect and clean data
Annotate data if necessary
Train Model
Choose or design an appropriate model and objective function
Train model and monitor performance during training
01
02
03
04
05
Monitor in production
Monitor customer reaction, check for biases, distribution shift, unexpected behavior
Release to production
Set up data pipeline and compute resources, design user interface, communicate to users the capabilities and limitations of the system
Evaluate
Define evaluation metrics and check for biases
Conduct human evaluation

Where do we get data?

- No existing dataset ➡ Have to collect data
- Ideal: Collect papers, have scientists read them and then write summaries
  - Problem: Expensive, time consuming
- Is there anywhere we can get naturally occurring data?



OpenReview.net

Search NIPS 2018 Workshop IRASL

Login

← Go to **NIPS 2018 Workshop IRASL** homepage

Variational Autoencoders with implicit priors for short-duration text-independent speaker verification

Anonymous

22 Oct 2018 (modified: 10 Sep 2019) NIPS 2018 Workshop IRASL Blind Submission Readers: Everyone Show Revisions

Abstract: In this work, we exploited different strategies to provide prior knowledge to commonly used generative modeling approaches aiming to obtain speaker-dependent low dimensional representations from short-duration segments of speech data, making use of available information of speaker identities. Namely, convolutional variational autoencoders are employed, and statistics of its learned posterior distribution are used as low dimensional representations of fixed length short-duration utterances. In order to enforce speaker dependency in the latent layer, we introduced a variation of the commonly used prior within the variational autoencoders framework, i.e. the model is simultaneously trained for reconstruction of inputs along with a discriminative task performed on top of latent layers outputs. The effectiveness of both triplet loss minimization and speaker recognition are evaluated as implicit priors on the challenging cross-language NIST SRE 2016 setting and compared against fully supervised and unsupervised baselines.

TL;DR: We evaluate the effectiveness of having auxiliary discriminative tasks performed on top of statistics of the posterior distribution learned by variational autoencoders to enforce speaker dependency.

Keywords: Speaker verification, Variational autoencoders

7 Replies

Show all from everybody

Acceptance Decision

NIPS 2018 Workshop IRASL OpenReview (privately revealed to you)

16 Nov 2018 NIPS 2018 Workshop IRASL Paper24 Decision Readers: Everyone

Decision: Reject

VAE-inspired technique for speaker recognition

NIPS 2018 Workshop IRASL Paper24 AnonReviewer3

13 Nov 2018 NIPS 2018 Workshop IRASL Paper24 Official Review Readers: Everyone

Review: The authors propose an autoencoder model to learn a representation for speaker verification using short-duration analysis windows. The variational loss term is replaced by a discriminative loss term. The proposed approach works fairly well when compared to x-vectors. However, a known adaptation technique provides much larger gains to an x-vector based system compared to the proposed.

* To my perspective, the proposed model stretches the definition of a variational autoencoder somewhat significantly. There is still an encoder and decoder with a reconstruction loss and a loss based on the encoder representation. However, the proposed model replaces the variational component of the loss with a discriminative loss function, either cross-entropy or triplet loss. This is somewhat similar to the approach described in the cited Lamb, et al. paper "Discriminative regularization for generative models", but in that paper the KL-div loss is augmented with a discriminative loss rather than being replaced.

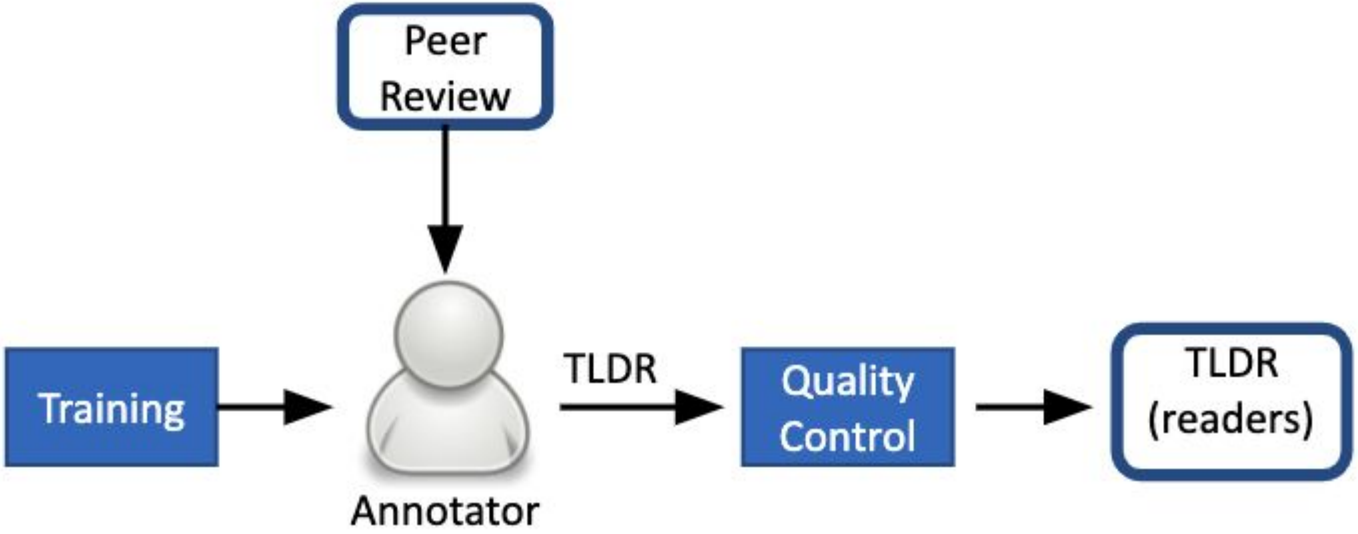
* The role of the short-duration window on the model or experiment is not clear. In Section 4: "embeddings of each recording are obtained from 256

Good work

The authors propose a method for learning node representations which, like previous work (e.g. node2vec, DeepWalk), is based on the skip-gram model. However, unlike previous work, they use the concept of shared neighborhood to define context rather than applying random walks on the graph. The paper is well-written and it is quite easy to follow along with the discussion. This work is most similar, in my opinion, to node2vec. In particular, when node2vec has its restart probability set pretty high, the random walks tend to stay within the local neighborhood (near the starting node). The main difference is in the sentence construction strategy. Whereas node2vec may sample walks that have context windows containing the same node, the proposed method does not as it uses a random permutation of...

A method for learning node representations using the concept of shared neighborhood to define context

We've found that this task is doable by CS undergrads after 2 half hour training sessions


Peer
Review
Training
Annotator
TLDR
Quality
Control
TLDR
(readers)

Data

Author's point of view

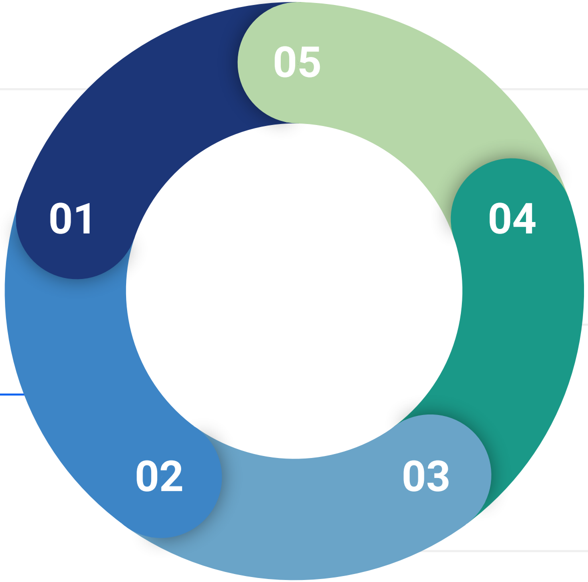
- From OpenReview, author written TLDRs
- ∼3.2k TLDRs (1 per paper)

Reader's point of view

- Reviewer comments rewritten as TLDRs
- ∼2.2k TLDRs

Collect Data
Collect and clean data
Annotate data if necessary

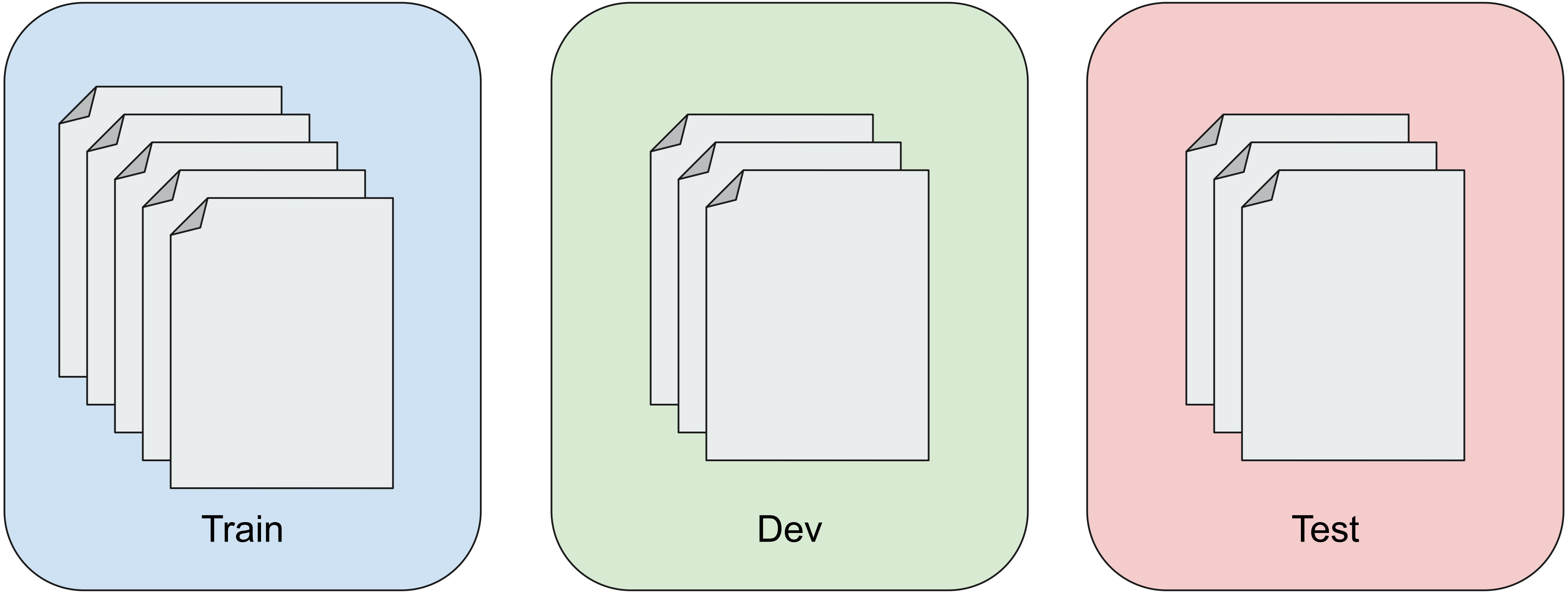
Train Model
Choose or design an appropriate model and objective function
Train model and monitor performance during training



Monitor in production
Monitor customer reaction, check for biases, distribution shift, unexpected behavior

Release to production
Set up data pipeline and compute resources, design user interface, communicate to users the capabilities and limitations of the system

Evaluate
Define evaluation metrics and check for biases
Conduct human evaluation



Randomly split data into 60/20/20 sets

Model choices

- **Extractive:** Choose the best sentence from the input as a summary
- **Abstractive:** Generate a summary given the input

Extractive Models

- PACSUM (Zheng and Lapata 2019)
    - Chooses most important sentence based on word frequency
- BERTSumExt (Liu and Lapata 2019)
    - Selects most important sentence using BERT embeddings
- MatchSum (Zhong et al., 2020)
    - Uses a BERT model to score the entire summary instead of a single extraction
- Oracle
    - Given the gold summary, choose the sentence with the highest ROUGE Score (shows an upper bound of performance)

Abstractive Models

- BART (Lewis et al 2019)
    - Encoder-decoder transformer model
- BART finetuned on XSUM then finetuned on our dataset
    - XSUM: news summarization dataset

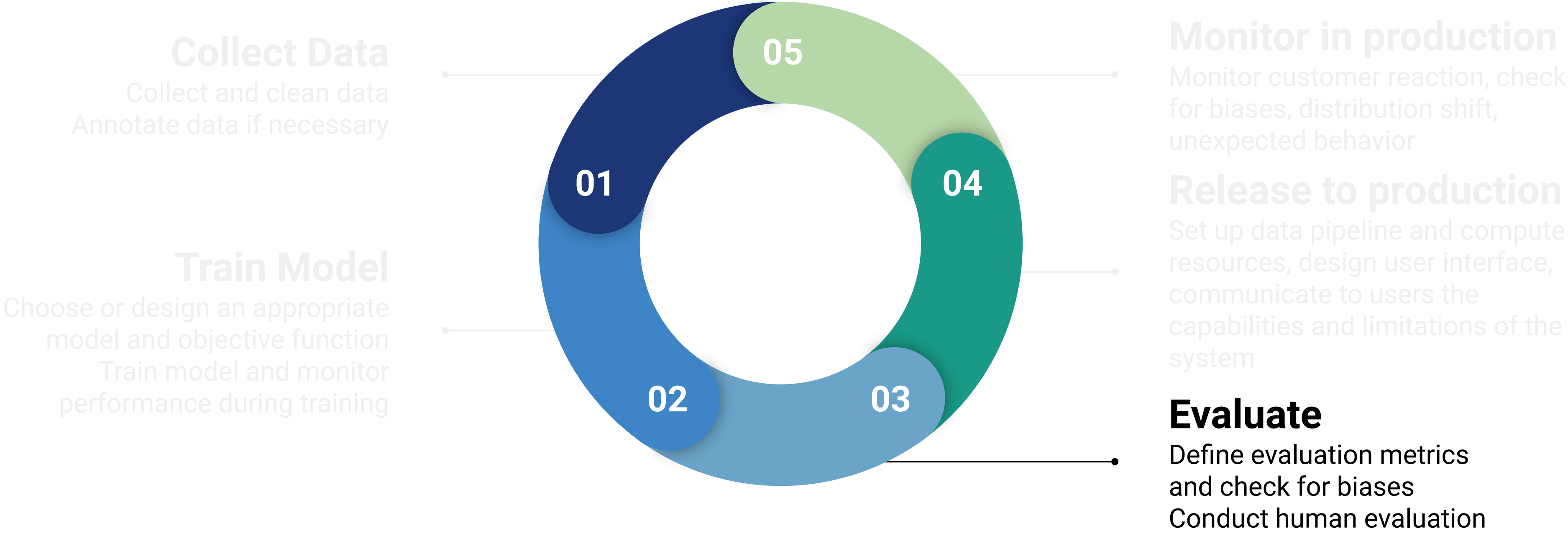
Other modelling choices

- Representing the full text of the paper is computationally expensive and impossible for some models (e.g. BART has a max sequence length of 1024)
- The full text of a paper isn't always open access

➡️ We'll use the Abstract only or the Abstract + Introduction + Conclusion

Implementation Details

- All models implemented in Python
- The authors from the extractive models released their code
- BART implemented in fairseq
  - Toolkit from Facebook
  - Uses Pytorch framework
- Code available here: https://github.com/allenai/scitldr

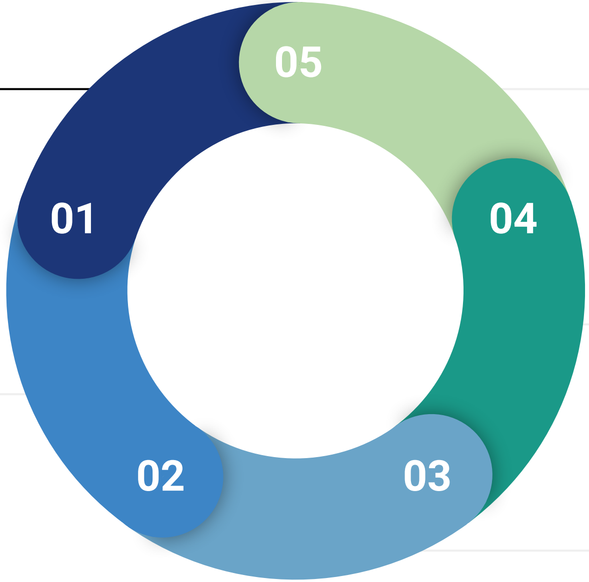

Collect Data
Collect and clean data
Annotate data if necessary
Train Model
Choose or design an appropriate model and objective function
Train model and monitor performance during training
01
02
03
04
05
Monitor in production
Monitor customer reaction, check for biases, distribution shift, unexpected behavior
Release to production
Set up data pipeline and compute resources, design user interface, communicate to users the capabilities and limitations of the system
Evaluate
Define evaluation metrics and check for biases
Conduct human evaluation

Rouge-1

Method	Abstract	AIC	Full Text
PACSUM	19.3	28.7	
BertSumExt	38.5	36.2	
MatchSum	42.7	38.6	
BART	43.3	42.9	
$BART_{XSUM}$	42.5	43.7	
Extractive Oracle	47.7	52.4	54.5

Collect Data

Collect and clean data
Annotate data if necessary



Train Model

Choose or design an appropriate model and objective function
Train model and monitor performance during training

Monitor in production

Monitor customer reaction, check for biases, distribution shift, unexpected behavior

Release to production

Set up data pipeline and compute resources, design user interface, communicate to users the capabilities and limitations of the system

Evaluate

Define evaluation metrics and check for biases
Conduct human evaluation

Use a related task - Title generation

- Titles are similar to TLDRs - they are short and contain salient information about the paper
- Titles are more widely available than TLDRs
- Task scaffolding
  - Prior work has shown that using an additional relevant task during training can support performance in the primary task (Swayamdipta et al., 2018; Cohan et al., 2019)

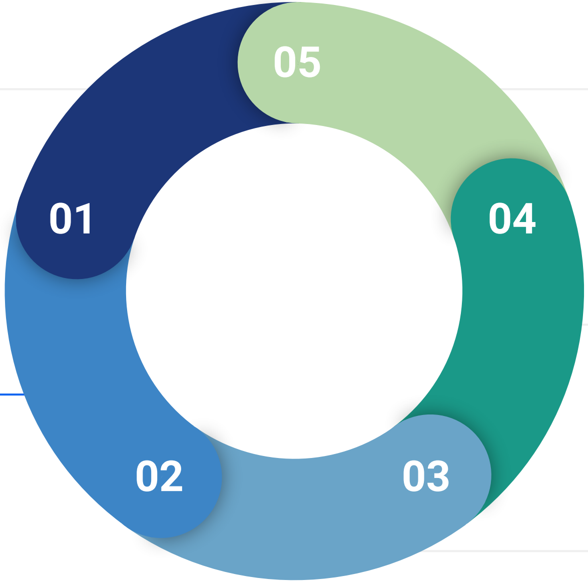
➡️ Collect an additional dataset of 20K Title - Paper pairs from arXiv

Collect Data

Collect and clean data
Annotate data if necessary

Train Model

Choose or design an appropriate model and objective function
Train model and monitor performance during training



Monitor in production

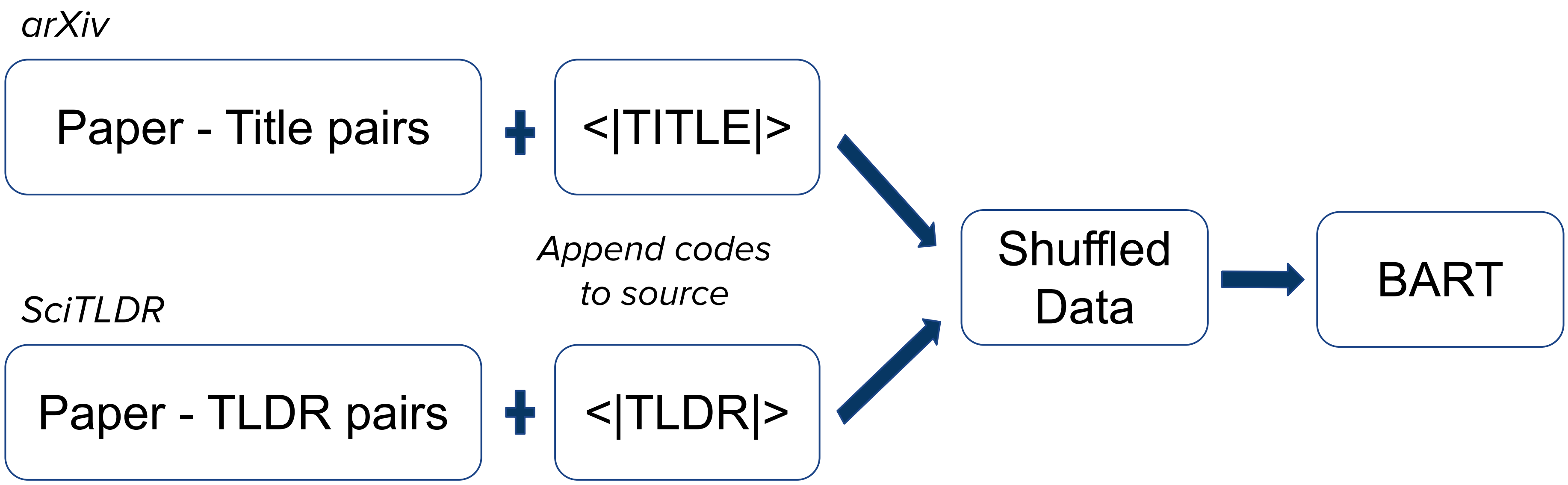
Monitor customer reaction, check for biases, distribution shift, unexpected behavior

Release to production

Set up data pipeline and compute resources, design user interface, communicate to users the capabilities and limitations of the system

Evaluate

Define evaluation metrics and check for biases
Conduct human evaluation



CATTS - Controlled Abstraction for TLDRs with Title Scaffolding

Collect Data
Collect and clean data
Annotate data if necessary

Train Model
Choose or design an appropriate model and objective function
Train model and monitor performance during training

Monitor in production
Monitor customer reaction, check for biases, distribution shift, unexpected behavior

Release to production
Set up data pipeline and compute resources, design user interface, communicate to users the capabilities and limitations of the system

Evaluate

Define evaluation metrics and check for biases
Conduct human evaluation

Rouge-1

Method	Abstract	AIC
PACSUM	19.3	28.7
BertSumExt	38.5	36.2
MatchSum	42.7	38.6
BART	43.3	42.9
$BART_{XSUM}$	42.5	43.7
CATTS	43.8	**44.9**
$CATTS_{XSUM}$	**44.3**	44.6
Extractive Oracle	47.7	52.4

Human Evaluation

- Are the summaries factual?
- Are the summaries fluent?
- Are the summaries informative?

Are the summaries factual?

Evaluating correctness requires careful reading and understanding of the source paper

First or second authors

1. False or misleading
2. Partially accurate
3. Mostly correct

- We perform this evaluation on both BART and CATTS
- We received responses from 29 unique authors with annotations covering 64 arXiv papers
- Both models receive an average rating of 2.5
- We observe 42 ties, 10 cases where BART is more correct, and 12 cases where CATTS is more correct

Are the summaries fluent?

- Randomly sample 100 papers and their generated summaries
- Only need to conduct fluency evaluation on abstractive summaries
- Manually annotate them for fluency
  - Binary labels - fluent or not
- Found 96% fluency

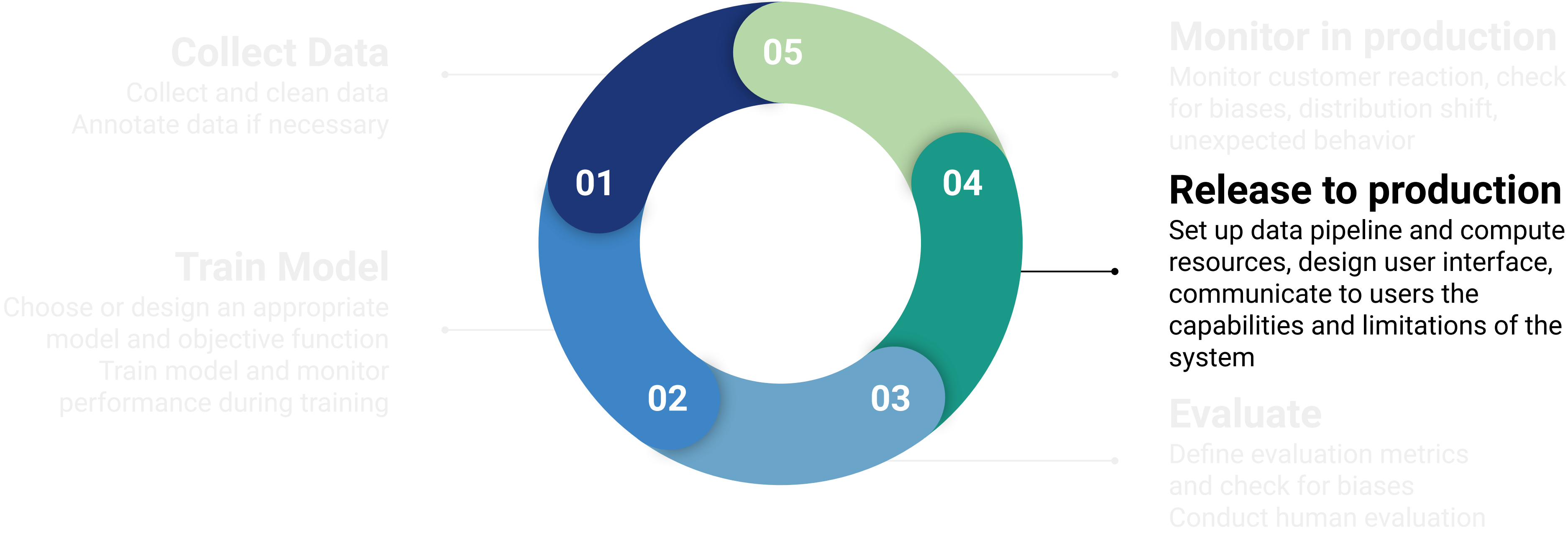
Are the summaries informative?

6 nuggets of information:

- Area, field, or topic of study
- Problem or motivation
- Mode of Contribution
- Details or description
- Results or findings
- Value or significance

	MRR	Avg. # nuggets	Avg. # words
TLDR-Auth (Gold)	0.53	2.5	20.5
TLDR-PR (Gold)	0.60	2.4	18.7
BART	0.42	2.2	19.4
CATTS	0.54	2.6	20.8

Higher MRR corresponds to variants that, on average, rank higher than others by length-normalized number of nuggets.

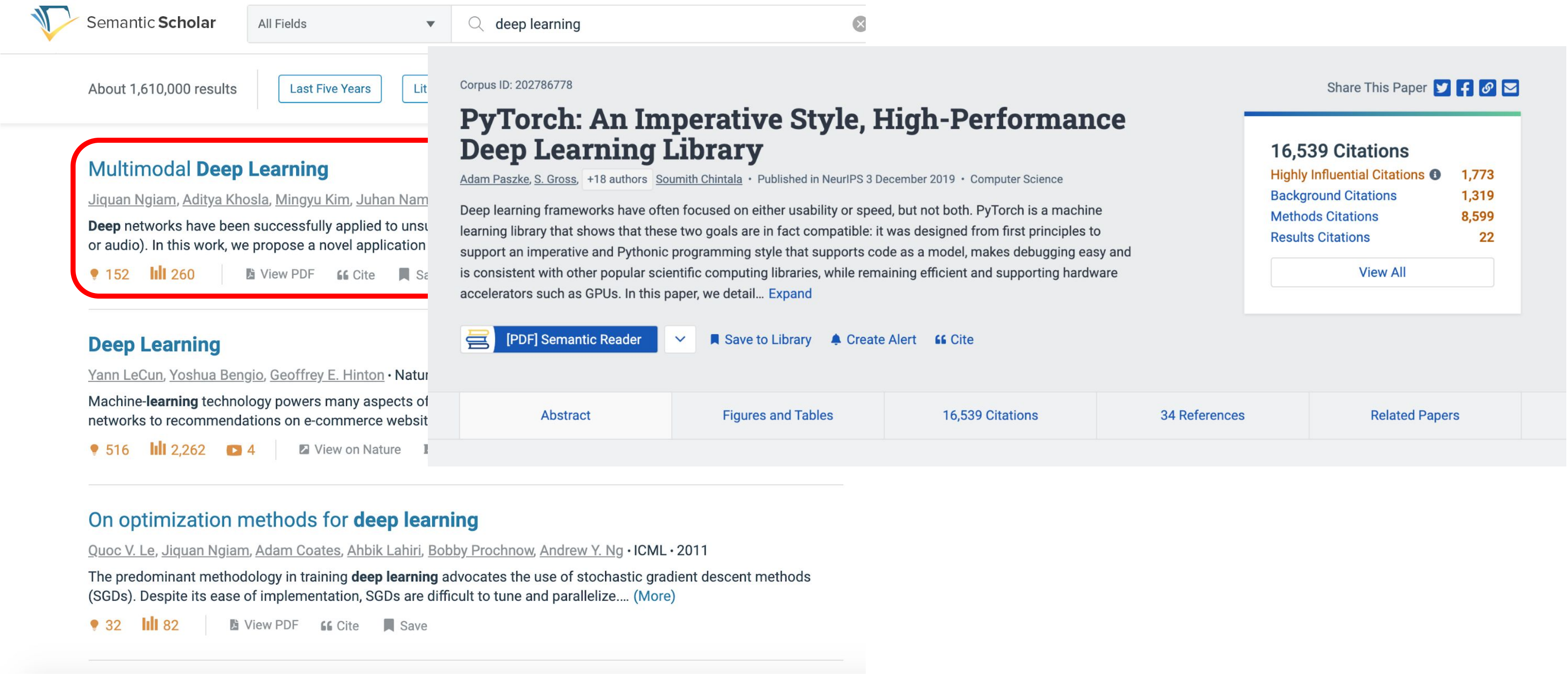

Collect Data
Collect and clean data
Annotate data if necessary
Train Model
Choose or design an appropriate model and objective function
Train model and monitor performance during training
01
02
03
04
05
Monitor in production
Monitor customer reaction, check for biases, distribution shift, unexpected behavior
Release to production
Set up data pipeline and compute resources, design user interface, communicate to users the capabilities and limitations of the system
Evaluate
Define evaluation metrics and check for biases
Conduct human evaluation

Goal: Release on Semantic Scholar

In order to release the model we need to...

- ❏ Decide the best use of TLDRs
- ❏ Build out an efficient backend
- ❏ Design the user interface
- ❏ Build method for user feedback
- ❏ Market and release

Decide the best use of TLDRs



Build an efficient backend

- Host the model on an AWS instance
- Build an API to easily interface the model
- Pre-process papers in batches and store generated TLDRs in database
  - Pros: Efficient client side, only need to pay for hosting when updating model
  - Cons: Harder to update model

Design User Interface

- How do we communicate what TLDRs are to the user?
- We want to ensure that the user understands the TLDRs are generated, without cluttering the UI too much
- Other considerations:
  - Integration with current features (bold face relevant terms?)
  - How to expand the abstract?
  - Release for all papers for just CS?

PyTorch: An Imperative Style, High-Performance Deep Learning Library

Adam Paszke, S. Gross, +18 authors Soumith Chintala • Computer Science • NeurIPS • 3 December 2019

TLDR This paper details the principles that drove the implementation of PyTorch and how they are reflected in its architecture, and explains how the careful and pragmatic implementation of the key components of its runtime enables them to work together to achieve compelling performance. Expand

PyTorch: An Imperative Style, High-Performance Deep Learning Library

Adam Paszke, S. Gross, +18 authors Soumith Chintala • Computer Science • NeurIPS • 3 December 2019

TLDR This paper details the principles that drove the implementation of PyTorch and how they are reflected in its
a ragmatic implementation of the key components of its runtime enables
t rformance. Expand

ve Alert Cite

> TLDR (short for Too Long, Didn't Read) is an automatically generated short summary of a paper.
>
> If you have feedback on this experience, contact us.

T to Adversarial Attacks

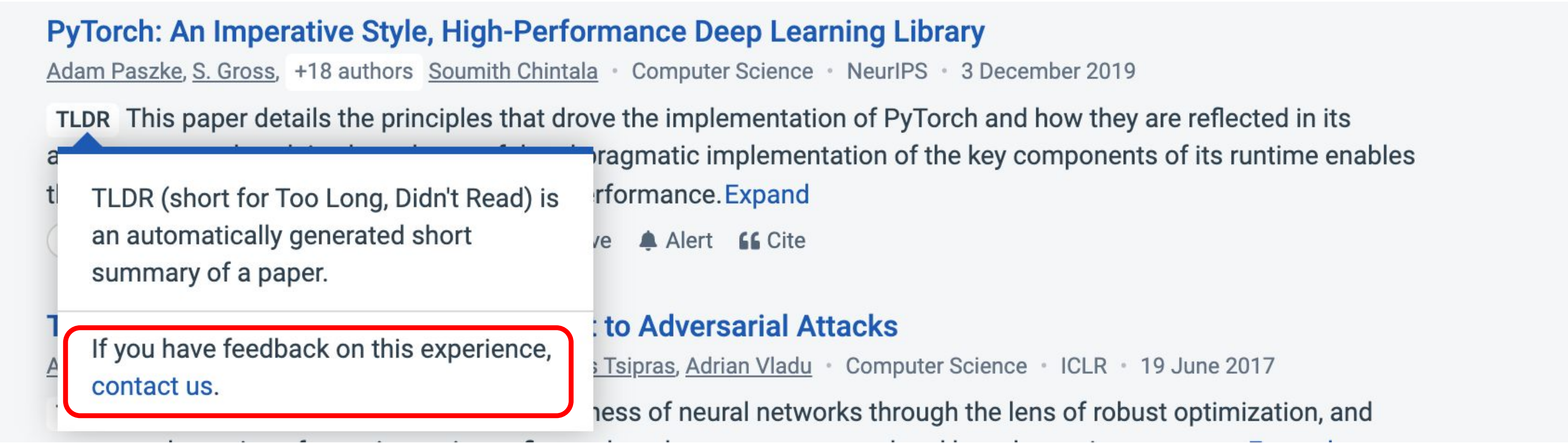
A s Tsipras, Adrian Vladu • Computer Science • ICLR • 19 June 2017

ness of neural networks through the lens of robust optimization, and

Build method for user feedback

Options:

- Ask users to rate generations
- Create feedback form
- Allow option to turn off feature -> track number of users that turn off feature
- Design other metrics (e.g. number of times user expands to abstract) and track those metrics


PyTorch: An Imperative Style, High-Performance Deep Learning Library
Adam Paszke, S. Gross, +18 authors Soumith Chintala • Computer Science • NeurIPS • 3 December 2019
TLDR This paper details the principles that drove the implementation of PyTorch and how they are reflected in its
ragmatic implementation of the key components of its runtime enables
rformance. Expand
Alert
Cite
TLDR (short for Too Long, Didn't Read) is an automatically generated short summary of a paper.
If you have feedback on this experience, contact us.
to Adversarial Attacks
Tsipras, Adrian Vladu • Computer Science • ICLR • 19 June 2017
ess of neural networks through the lens of robust optimization, and

Market and Release

Reload this page MANTIC SCHOLAR

Search over 203 million papers from all fields of science

Search

Account

SEMANTIC SCHOLAR TLDR BETA

Locate the right papers, and spend your time reading what matters to you.

The TLDR feature from Semantic Scholar puts automatically generated single-sentence paper summaries right on the search results page

What Are TLDRs?

TLDRs (Too Long; Didn't Read) are super-short summaries of the main objective and results of a scientific paper generated using expert background knowledge and the latest GPT-3 style NLP techniques. This new feature is available in beta for nearly 60 million papers in computer science, biology, and medicine.

Market and Release

Semantic Scholar
@SemanticScholar

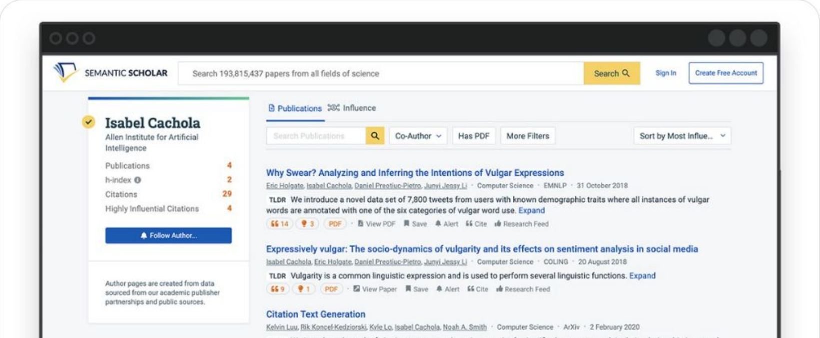
Introducing TLDRs on Semantic Scholar, now available in beta for ~10M Computer Science papers!

These auto-generated extreme summaries help you decide which papers are most relevant to your work.

Learn more:

medium.com/ai2-blog/intro...

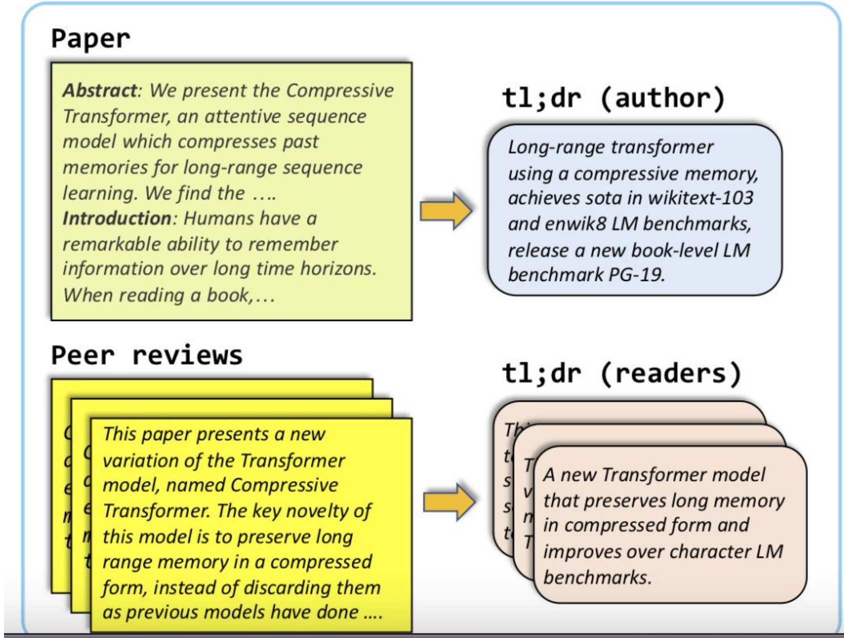
#TLDR #sciencetwitter



Isabel Cachola
@isabelcachola

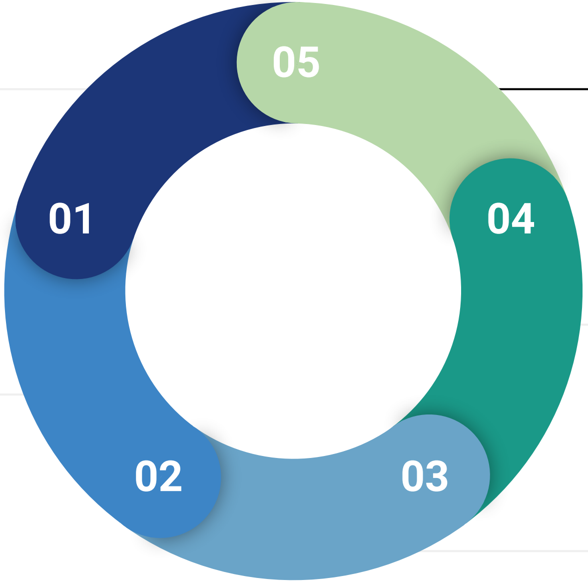
I'm exited to announce our paper "TLDR: Extreme Summarization of Scientific Documents"
With @kylelostat @armancohan @dsweld

Paper: arxiv.org/abs/2004.15011
Demo: scitldr.apps.allenai.org
Repo: github.com/allenai/scitldr
(1/4)



Collect Data
Collect and clean data
Annotate data if necessary

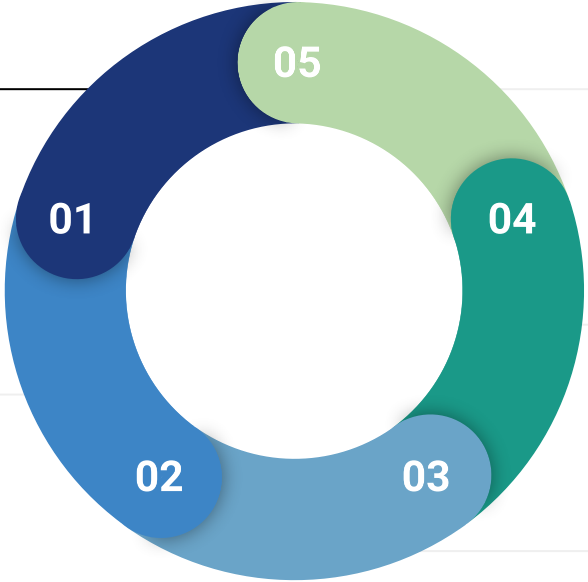
Train Model
Choose or design an appropriate model and objective function
Train model and monitor performance during training



Evaluate
Define evaluation metrics and check for biases
Conduct human evaluation

Release to production
Set up data pipeline and compute resources, design user interface, communicate to users the capabilities and limitations of the system

Monitor in production
Monitor customer reaction, check for biases, distribution shift, unexpected behavior



Collect Data

Collect and clean data
Annotate data if necessary

Train Model

Choose or design an appropriate model and objective function
Train model and monitor performance during training

Evaluate

Define evaluation metrics and check for biases
Conduct human evaluation

Release to production

Set up data pipeline and compute resources, design user interface, communicate to users the capabilities and limitations of the system

Monitor in production

Monitor customer reaction, check for biases, distribution shift, unexpected behavior

Final Presentation

Final Presentation

- Work in groups of 2-3 people
- Choose an application of language generation
- Walk through the 5 steps of the application lifecycle and discuss what you might do and the considerations at each step
- Discuss the ethical implications of your application
- Presentation should be ~10 minutes

Final Presentation Outline

1. Task introduction
   a. What is it
   b. What are the inputs/outputs
   c. Why is it important

Final Presentation Outline

2. The steps of ML
    a. Data collection - Where would you get the data?
    b. Modelling - What architecture might work and why?
    c. Evaluation - What method(s) of evaluation are appropriate?
    d. Release to production - What considerations are important to release your application?
    e. Monitor in production - What checks would you put in place to make sure you model continues to perform well?

Final Presentation Outline

3. Ethics
   a. What ethical considerations does your application have?

Next Week - Guest Lecture

- Speaker: Carlos Aguirre
- Title “Fairness and Biases in Language Models”
- Website: https://www.pocaguirre.com/