[**TALK1: TeraScale NLP To Accelerate Research**](#_z3u7dicy773z) **1**

[Abstract](#_mmxs6ghtf4o8) 1

[Logistics](#_pom4g0zh3q9m) 2

[Contributions](#_28hlorvgv22k) 2

[Longformer](#_rkky0jx8ai4) 2

[Exploiting structure In Document](#_yi9z2mretkp4) 3

[Sharing Dataset](#_kh4fcpdydisd) 3

[Discovery: Recommender for Scientic Articles](#_ct1zex359kx5) 3

[Data Collection](#_180nn2rl9o4h) 5

[Inference](#_at5sl9c0x5a8) 5

[Discovery: Personalized and Alert Explanation](#_ujepfmtkfn1n) 7

[Decide: How to Decide Whether To Read Paper](#_345992c012d8) 7

[Dataset:](#_xe2nh5wvqf1b) 8

[Read](#_3heypwu51y1c) 10

[**TALK2: What's Next In Infrastructure for ML**](#_3ofzsyezvtq7) **11**

[Abstract](#_4maa17yyii5m) 11

[Logistics](#_nc9oo9lyogbd) 11

[ML is different from Traditional Software](#_xgv6dwq97h6q) 12

[Approach1: Systems to Support production ML](#_o1qr4591rv7v) 13

[Approach2: Production friendly ML methods](#_dllusvim5wbx) 14

[**TALK: Data**](#_s2qzj8mrdmvb) **19**

[**TALK3: Toward an Easy to Use AI Platform: Data,Algorithm, and System**](#_efwnpn4qffwm) **19**

[Abstract](#_jryke2q8qz01) 19

[Data Preparation](#_pwoxm67qzn2b) 20

# **TALK1: TeraScale NLP To Accelerate Research**

## Abstract

## Logistics

<https://virtual.2021.kdd.org/plenary_session_invited_talk_by_Daniel_S_WELD.html>

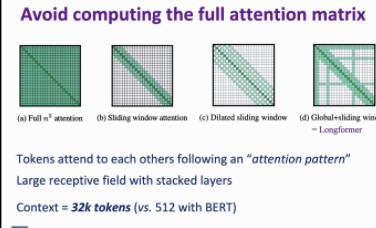
10:45AM-11:30AM

### Contributions

#### Longformer

### 

Transformers attention is n2, so does not scale well.



LongFormer can handle 32k tokens!

For loop is memory efficient, but not computation efficient

Sliding window --> dilated sliding window → global+sliding window

#### Exploiting structure In Document

Exploiting Strucutre of Docuemnt (Hierarchy): links, titles, references

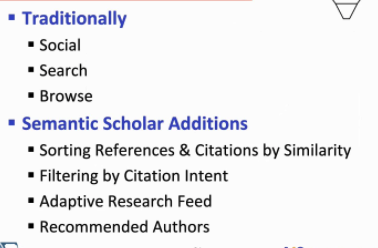
#### Sharing Dataset

research.semanticsscholor.rog.resources

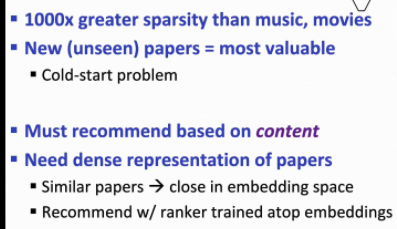
CORD-19

### Discovery: Recommender for Scientic Articles

#### 

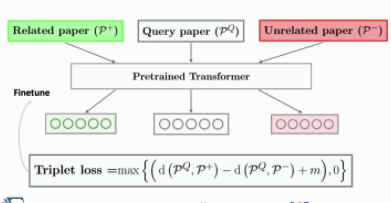


Recommender Challenges for Scientific papers



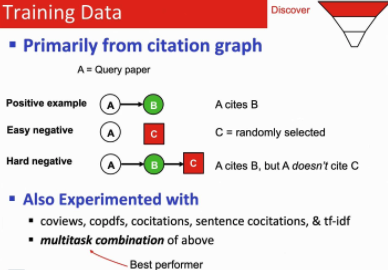
New paper is more interesting

Sparse → cannot use collaboaritive

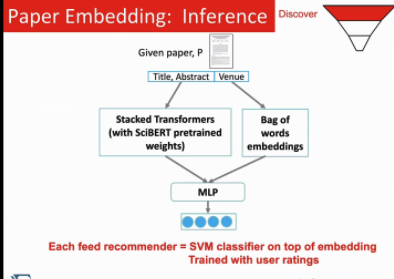


Use tripler loss: minimize distance between query, paper+, maximize diff between paper+, paper=

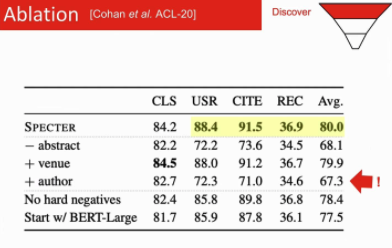
#### Data Collection



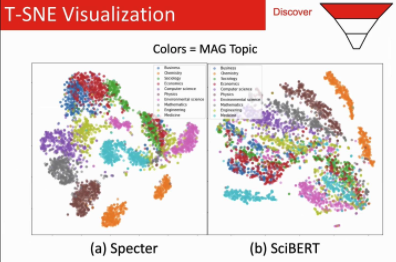
#### Inference



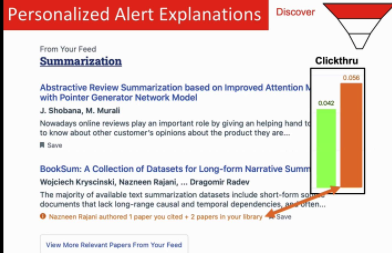
Results/Analysis



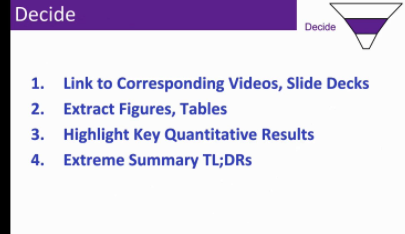
Model does not take into account of authors

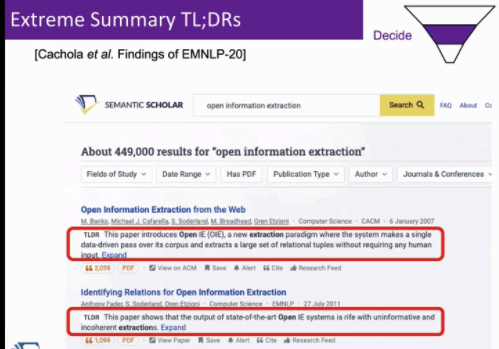


### Discovery: Personalized and Alert Explanation

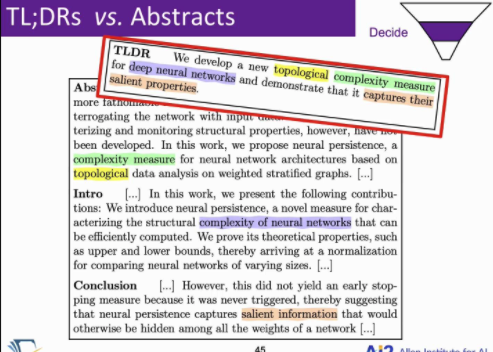


### Decide: How to Decide Whether To Read Paper



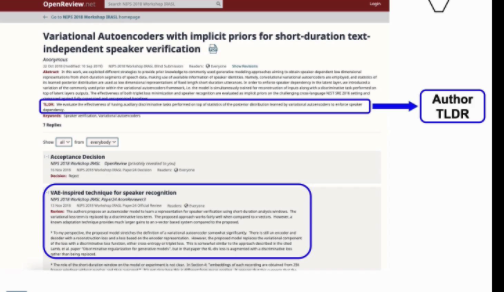


~15 tokens long!!!

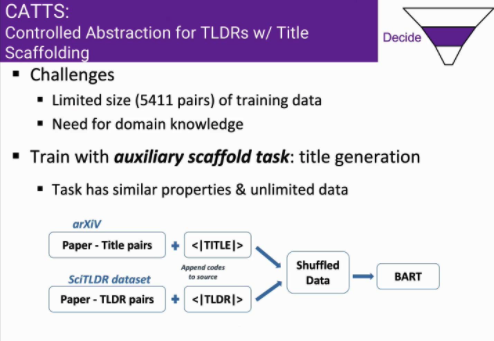


#### Dataset:

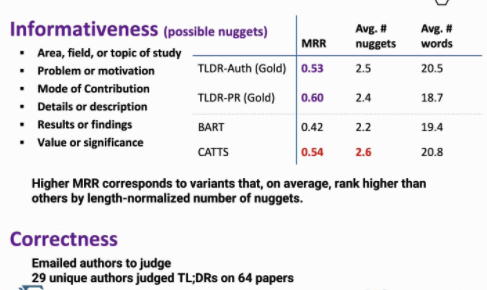
Scrape author TLDR



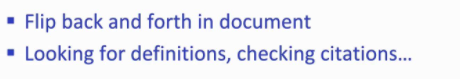
But still want more dataset: take open reviews and have an annotator condensi it more



5400 is still too little. Augment with title generation task

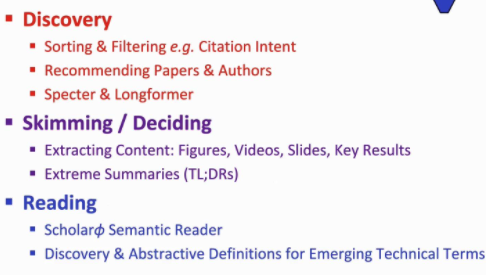


### Read



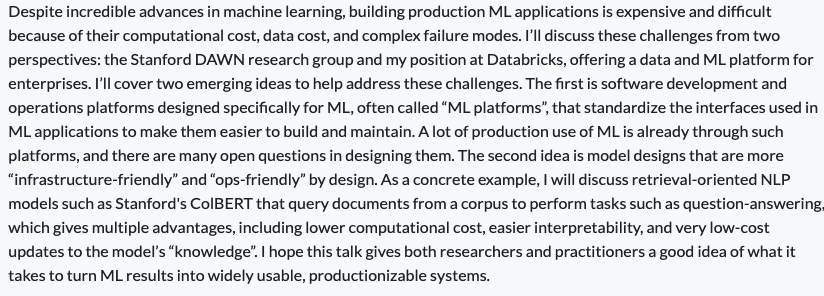
Reading is hard because you lose flow going back and forth





# **TALK2: What's Next In Infrastructure for ML**

## Abstract

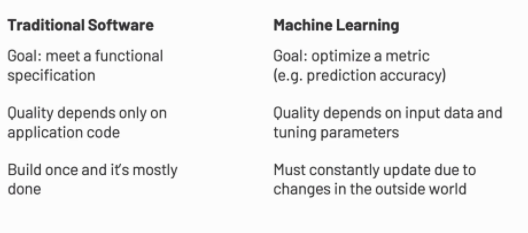


## Logistics

Monday, 1:45PM - 2:30 PDT

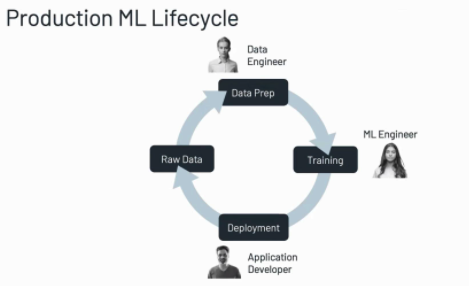
https://virtual.2021.kdd.org/plenary\_session\_invited\_talk\_by\_Matei\_ZAHARIA.html

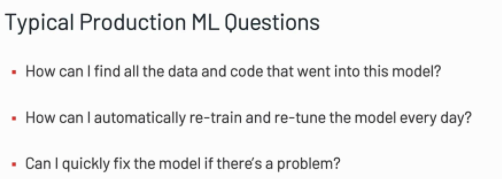
## ML is different from Traditional Software



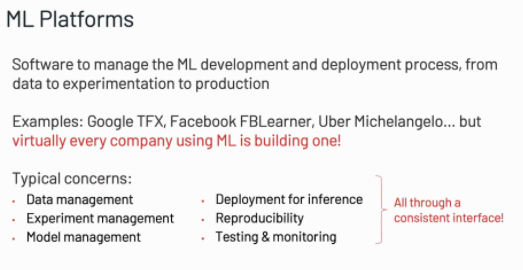
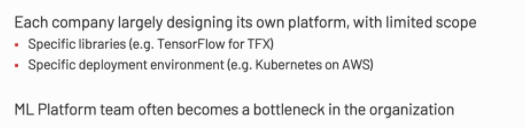
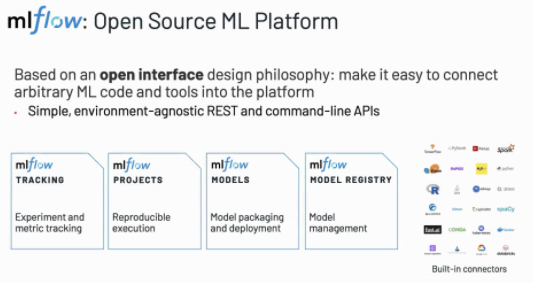
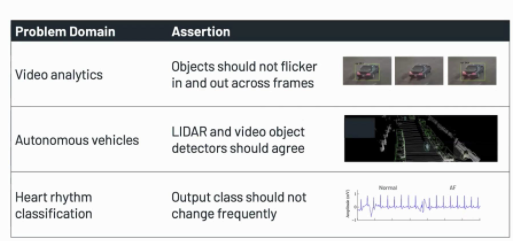
Data is harder to check for quality, that unit tests

* Training is getting easier; lifecyle is not there yet.

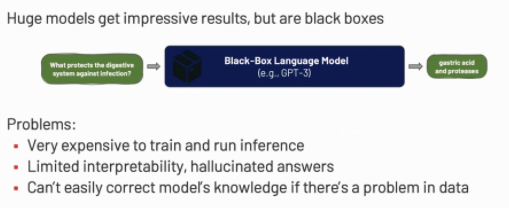
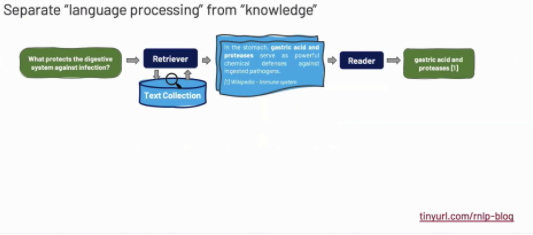
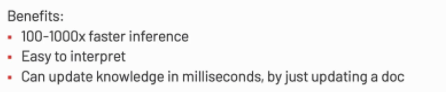
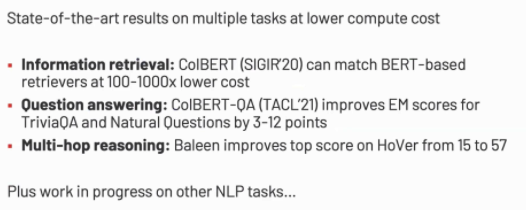
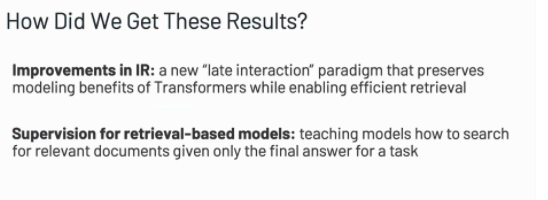
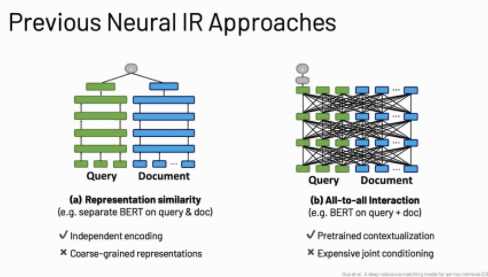
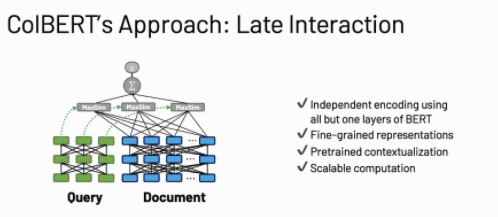
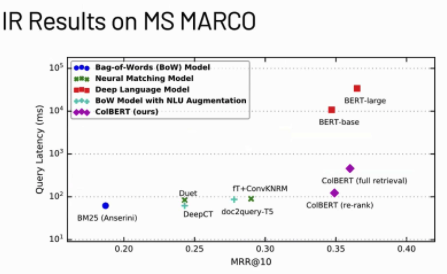
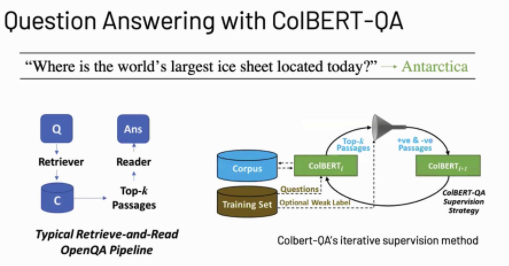
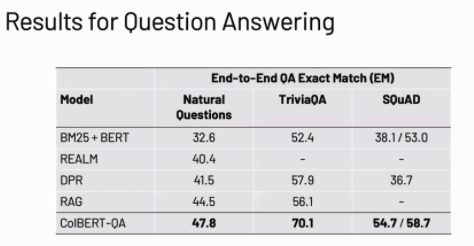
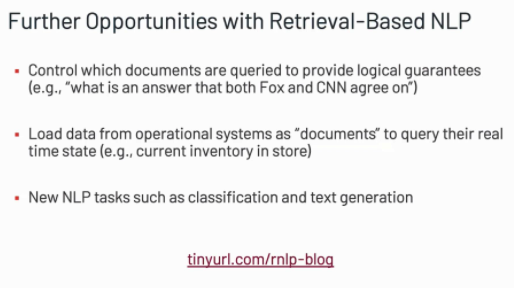




## Approach1: Systems to Support production ML

* ML Platforms, new abstractions
* 
* 
* Companies with deep expertise and $building their own. Even after its built, need alot of support new models
* Google Cloud, Amazon Sagemaker:
* Open source code: ML
  + 
* Model base assertion
  + 
  + https://arxiv.org/pdf/2003.01668.pdf

## Approach2: Production friendly ML methods

* Ex: Retrieval-based NLP
* Problem with black box function models
  + 
  + Simon’s NER model: how to fix one bad case
* Retrieval Based NLP
  + 
  + Retrieva and reader model can be alot smaller than deep NN
  + 
  + 
  + 
    - IR: hash or invert table
* Intuition fo IR
  + Prevous
    - Method1:
      * Retrieval: Nearest neighbor lookup
      * Not good because document can have mulptple representations
    - Method2: (Google)
      * Inference is expensive: look at thousands
  + Colbert
    - 
    - Doing interaction at the latest level of query and document at the last level; efficient with index
    - 
    - Reference
      * <https://github.com/stanford-futuredata/ColBERT>
      * [Stanford](https://hai.stanford.edu/news/moderate-proposal-radically-better-ai-powered-web-search?utm_source=Stanford+HAI&utm_campaign=97fd06f5fa-Mailchimp_HAI_Newsletter_July_2021_2&utm_medium=email&utm_term=0_aaf04f4a4b-97fd06f5fa-214011318)
* Colbert applied to Question and Answering
  + 
  + Use the answer to the question as a weak supervision for next iteration
  + 
* Future Opportunities

# **TALK: Data**

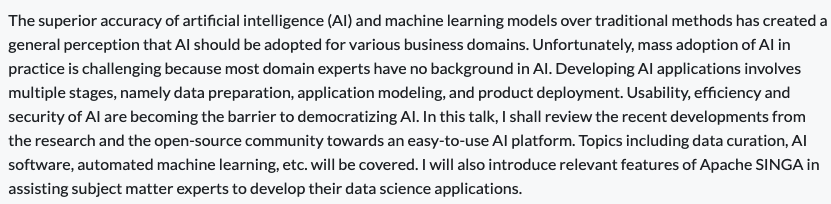
# 

There are deficiencies in all these approaches.

If you make your own, is dataset balanced; what’s the license? How was it collectd?

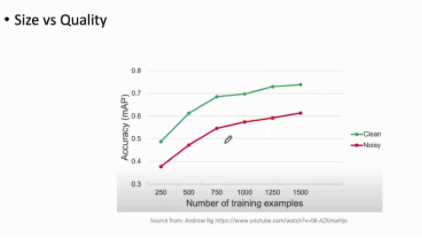
# **TALK3: Toward an Easy to Use AI Platform: Data,Algorithm, and System**

## Abstract

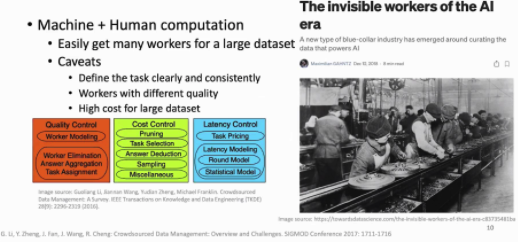




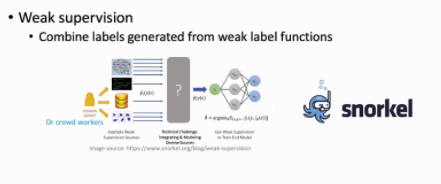
## Data Preparation



* More data can help with model
* But more data may require more labels, ie crowd-sourcing. But need to define task consistently. Combining workers results needs to be analyzed; can introduce noise

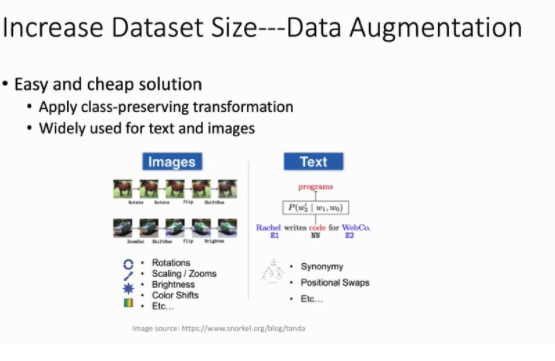


Another approach, is to use automatic labeling via weak supervision

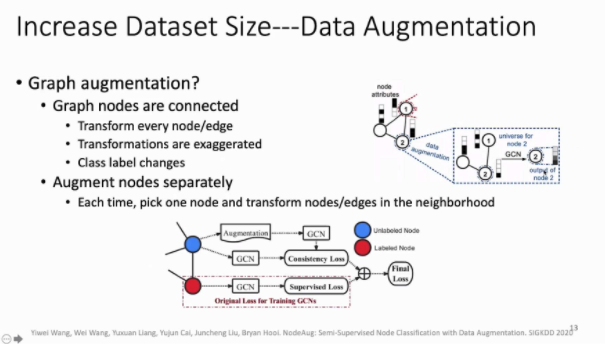


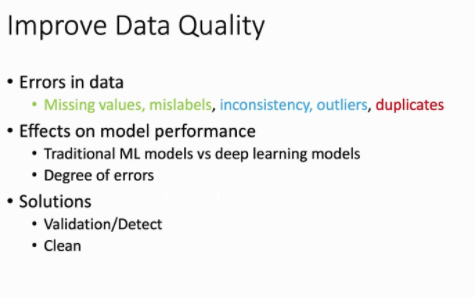
What is snorkel? TODO: look into it. Seems to be some sort of label probabage

Another approach



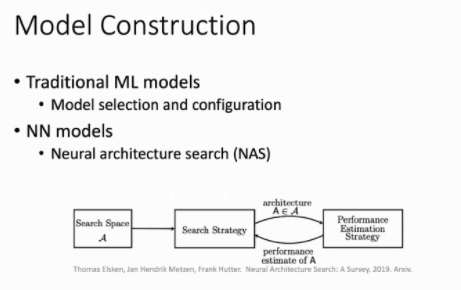
Graph augmentation:





5 types of errors. Greens mean fixing these wil improve error changes. Blue means no impact.

DL are somewhat immune to some level of errors.



Is NAS auto-ml? Or some sort of grid search.

