Deep Data Integration

Wang-Chiew Tan
Facebook AI
wangchiew@fb.com

Disclaimer: All opinions presented in this talk are my own.
Data Integration

- The data integration problem:
  - provide uniform access to disparate data sources
- The user sees only one data source
Data Integration

● The data integration problem:
  ○ provide uniform access to disparate data sources
● The user sees only one data source

● Traditionally, two approaches:
  ○ Virtual Data Integration
  ○ Data Warehouse

● Today:
  ○ Data Lake
Virtual Data Integration

- Data reside at their original locations
- Global schema ⇒ uniform view of underlying data sources
**Data Warehouse**

- Data is consolidated at the warehouse
- Warehouse $\Rightarrow$ uniform view of underlying data sources
Data Lake

- Massive collection of raw data
- May not have a schema
- May have different types
- May be in different locations

- How can we query the data lake?
The Data Integration Ecosystem

- **Data Discovery**: What are the relevant data sources?
- **Data Extraction**: How to identify and extract relevant information from sources?
- **Schema Matching/Schema Mapping**: How are data in different sources potentially related? How to specify the relationship between the source and global/warehouse schema?
- **Entity Matching**: How to identify identical entities in different sources?
- **Data Cleaning**: How to manage missing or erroneous data?
Outline

- Data Integration and Data Preparation
- Deep Learning
- Case Study: Entity Matching with Pre-trained Language Models
- Challenges and Opportunities
A Neural Network (NN)

- (1) an input layer – a numerical representation of data,
- (2) one or more hidden layers,
- (3) an output layer

- Input: a numerical representation of data
- Output: the answer

Deep = many many hidden layers
A Neuron

- Each neuron passes information as defined above
  - $w =$ weight, $b =$ bias, $f =$ activation function
- The learning process tunes $w$ and $b$:
  - compare predicted output with actual output
  - adjust $w$ and $b$ in all layers to minimize a loss function (e.g., mean squared error) through back propagation
The Network Zoo (https://www.asimovinstitute.org/neural-network-zoo/)
Transformers

A gentle introduction to BERT model – Anand Srivastava
https://inblog.in/A-gentle-introduction-to-BERT-Model-JfGFFXb97v
Transformers

● Self-Attention
  ○ Calculates vector representation of a token based on its relation to all neighboring tokens ➔ contextualized embeddings
    ○ “The river **bank** was covered with flowers”
    ○ ”The **bank** issued a financial statement”

● Multi-head attention
  ○ Contextualized embeddings for different relations (e.g., subj-verb, subj-adj relations)

● Positional embeddings
  ○ Self-attention is position invariant
  ○ Positional embeddings used to indicate relative word positions
BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding [Devlin+ NAACL 2019]

- Pre-training/fine-tuning paradigm
- Pre-trained on two unsupervised tasks simultaneously
  - Masked Language Model
  - Next Sentence Prediction
- Trained on large BookCorpus and English Wikipedia datasets
- Fine-tuning (later)

- Takes entire sequence of tokens as input simultaneously
Transformers War

BERT (DistillBert, BERT<sub>base</sub>, BERT<sub>large</sub>)

[Bert][Raffel+ JMLR2019 (Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer)]

GPT-3 (GPT2, GPT)

[Brown+ NeurIPS2020 (Language Models are Few Shot Learners)]

XLM-R

[XLM-R][Yang+ NeurIPS2019 (XLNet: Generalized Autoregressive Pretraining for Language Understanding)]

XLNet

[XLNet][Lewis ACL2020 (BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension)]

BART

Albert

[Conneau+ ACL2020 (Unsupervised Cross-lingual Representation Learning at Scale)]

[Albert][Conneau+ ACL2020 (Unsupervised Cross-lingual Representation Learning at Scale)]

DeBERTa

[Lan+ ICLR2020 (ALBERT: A Lite BERT for Self-supervised Learning of Language Representations)]

[T5][He+ arXiv2020 (DeBERTa: Decoding-enhanced BERT with Disentangled Attention)]
Entity Matching (EM)

- Given two data sources, find all pairs of entities, one from each data source, that refer to the same entity
- One of the most prevalent problems in data integration
- Important for deduplication, KB construction, data search
- Work as early as [Felligi & Sunter J. American Statistical Assoc. 1969 (A Theory for Record Linkage)]
- The name itself needs entity resolution! [Gurajada+ CIKM2019 (Learning-Based Methods with Human-in-the-Loop for Entity Resolution)]

<table>
<thead>
<tr>
<th>Entity resolution</th>
<th>Record linkage</th>
<th>Reference reconciliation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Duplicate detection</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Ditto: Deep Entity Matching with Pre-trained Language Models

[Yuliang Li, Jinfeng Li, Yoshihiko Suhara, AnHai Doan, T. VLDB2021]

- **Input**: Two collections of data entries (tables, JSON files, text, ...)
- **Output**: all entry pairs that refer to the same entity (products, businesses, ...)

<table>
<thead>
<tr>
<th>title</th>
<th>manf./modelno</th>
<th>price</th>
</tr>
</thead>
<tbody>
<tr>
<td>instant immersion spanish deluxe 2.0</td>
<td>topics entertainment</td>
<td>49.99</td>
</tr>
<tr>
<td>adventure workshop 4th-6th grade 7th edition</td>
<td>encore software</td>
<td>19.99</td>
</tr>
<tr>
<td>sharp printing calculator</td>
<td>sharp el1192bl</td>
<td>37.63</td>
</tr>
</tbody>
</table>

Table A:

<table>
<thead>
<tr>
<th>title</th>
<th>price</th>
</tr>
</thead>
<tbody>
<tr>
<td>instant immers spanish dlux 2</td>
<td>36.11</td>
</tr>
<tr>
<td>encore inc adventure workshop 4th-6th grade 8th edition</td>
<td>17.1</td>
</tr>
<tr>
<td>new-sharp shr-el1192bl two-color printing calculator 12-digit lcd black red</td>
<td>56.0</td>
</tr>
</tbody>
</table>

Table B:
Two Phases of Entity Matching

- **Blocking**
  - Reduce the number of pairwise comparisons (otherwise $O(N^2)$)
  - Simple heuristics, e.g., two entries must share at least 1 token

- **Matching:**
  - Decide whether each candidate pair is a real match
  - Rules, Crowdsourcing, classic ML, **Deep Learning**, etc.

<table>
<thead>
<tr>
<th>title</th>
<th>manf./modelno</th>
<th>price</th>
</tr>
</thead>
<tbody>
<tr>
<td>instant immersion</td>
<td>topics</td>
<td>49.99</td>
</tr>
<tr>
<td>spanish deluxe 2.0</td>
<td>entertainment</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>adventure workshop</td>
<td>encore</td>
<td>19.99</td>
</tr>
<tr>
<td>4th-6th grade 7th edition</td>
<td>software</td>
<td></td>
</tr>
<tr>
<td>sharp printing calculator</td>
<td>sharp el1192bl</td>
<td>37.63</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>title</th>
<th>price</th>
</tr>
</thead>
<tbody>
<tr>
<td>instant immers spanish dlux 2</td>
<td>36.11</td>
</tr>
<tr>
<td>encore inc adventure workshop 4th-6th grade 8th edition</td>
<td>17.1</td>
</tr>
<tr>
<td>new-sharp shr-el1192bl two-color printing calculator 12-digit lcd black red</td>
<td>56.0</td>
</tr>
</tbody>
</table>
Entity Matching is Challenging

<table>
<thead>
<tr>
<th>title</th>
<th>manf./modelno</th>
<th>price</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>instant immersion spanish deluxe 2.0</em></td>
<td>topics entertainment</td>
<td>49.99</td>
</tr>
<tr>
<td><em>adventure workshop 4th-6th grade 7th edition</em></td>
<td>encore software</td>
<td>19.99</td>
</tr>
<tr>
<td><em>sharp printing calculator</em></td>
<td>sharp el1192bl</td>
<td>37.63</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>title</th>
<th>price</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>instant immers spanish dlux 2</em></td>
<td>36.11</td>
</tr>
<tr>
<td>encore inc adventure workshop 4th-6th grade 8th edition</td>
<td>17.1</td>
</tr>
<tr>
<td>new-sharp shr-el1192bl two-color printing calculator 12-digit lcd black red</td>
<td>56.0</td>
</tr>
</tbody>
</table>

State-of-the-art EM solutions fail to match/non-match in all these 3 cases! (as of April 2020)
Challenges

- Observations:
  - Language understanding is an important component of EM
  - What to pay attention to for each record
  - Dirty data

<table>
<thead>
<tr>
<th>title</th>
<th>manf./modelno</th>
<th>price</th>
</tr>
</thead>
<tbody>
<tr>
<td>instant immersion spanish deluxe 2.0</td>
<td>topics entertainment</td>
<td>49.99</td>
</tr>
<tr>
<td>adventure workshop 4th-6th grade 7th edition</td>
<td>encore software</td>
<td>19.99</td>
</tr>
<tr>
<td>sharp printing calculator</td>
<td>sharp el1192bl</td>
<td>37.63</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>title</th>
<th>price</th>
</tr>
</thead>
<tbody>
<tr>
<td>instant immers spanish dlux 2</td>
<td>36.11</td>
</tr>
<tr>
<td>encore inc adventure workshop 4th-6th grade 8th edition</td>
<td>17.1</td>
</tr>
<tr>
<td>new-sharp shr-el1192bl two-color printing calculator 12-digit lcd black red</td>
<td>56.0</td>
</tr>
</tbody>
</table>
Fine-tuning Pre-trained Language Models

- Pre-trained LM are already trained on a large dataset
- Strong baselines for several NLP tasks
- “Cheaper” to fine-tune a pre-trained LM with labeled data for your needs than to pre-train a model from scratch

- Train some layers, freeze the others
- E.g., Freeze all layers, attach new layers, train the weights of the new layers
Ditto’s Model Architecture
Serialization

- Serialize each entity:
  
<table>
<thead>
<tr>
<th>COL</th>
<th>VAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>title</td>
<td>instant immers spanish dlux 2</td>
</tr>
<tr>
<td>manf./modelno</td>
<td>NULL</td>
</tr>
<tr>
<td>price</td>
<td>36.11</td>
</tr>
</tbody>
</table>

- Apply LM (e.g., BERT) for sequence pair classification!

  [CLS] serialize(e) [SEP] serialize(e’) [SEP]
Ditto’s Model Architecture

RoBERTa for better performance and DistilBERT for fast training / prediction
Optimizations in Ditto

- **Injecting Domain-Knowledge:**
  - allow the user to specify information that is more important (e.g., PID)
  - e.g., “… new-sharp [ID] shr-el1192bl [/ID] two-color …”

- **Span typing:** Use spacy or regex to identify and assign entity types

<table>
<thead>
<tr>
<th>Entity Type</th>
<th>Types of Important Spans</th>
</tr>
</thead>
<tbody>
<tr>
<td>Publications, Movies, Music Organizations, Employers Products</td>
<td>Persons (e.g., Authors), Year, Publisher Last 4-digit of phone, Street number Product ID, Brand, Configurations (num.)</td>
</tr>
</tbody>
</table>

- **Span normalization:** Normalize spans (e.g., numbers, years) into the same formats
Optimizations in Ditto

- **Summarization:**
  - Transformers have a max sequence length (e.g., 512)
  - Keep only the essential information → keep tokens of high TF-IDF

- **Data Augmentation:**
  - Allows the model to learn “harder” by modifying the training data
  - e.g., Dropping a span, delete an attribute, swapping two attributes, ...
  - MixDA: performs a convex interpolation on original and augmented text to generate a new one
Experiments

- **Benchmark 1:** ER-Magellan
  - 13 datasets
  - 3 domains: publications, products, and businesses
  - 3 categories: Structured, Dirty, and Textual

- **Benchmark 2:** WDC Product Matching
  - >200K of product pairs
  - 4 product categories: computers, cameras, shoes, and watches
  - small (1/20), medium (1/8), large (1/2), and xlarge (1/1)

- **Baseline:** DeepMatcher (DM), the SOTA deep learning model for matching
  - We compare the F1 score and the training time

- Also ran on a real company matching dataset
## Experiments: ER-Magellan datasets (w/ RoBERTa)

<table>
<thead>
<tr>
<th>Datasets</th>
<th>Size</th>
<th>Ditto</th>
<th>DeepMatcher</th>
</tr>
</thead>
<tbody>
<tr>
<td>Structured/Amazon-Google</td>
<td>11,460</td>
<td>75.58</td>
<td>69.30</td>
</tr>
<tr>
<td>Structured/Beer</td>
<td>450</td>
<td>94.37</td>
<td>78.80</td>
</tr>
<tr>
<td>Structured/DBLP-ACM</td>
<td>12,363</td>
<td>98.99</td>
<td>98.40</td>
</tr>
<tr>
<td>Structured/DBLP-GoogleScholar</td>
<td>28,707</td>
<td>95.60</td>
<td>94.70</td>
</tr>
<tr>
<td>Structured/Fodors-Zagats</td>
<td>946</td>
<td>100.00</td>
<td>100.00</td>
</tr>
<tr>
<td>Structured/iTunes-Amazon</td>
<td>539</td>
<td>97.06</td>
<td>91.20</td>
</tr>
<tr>
<td>Structured/Walmart-Amazon</td>
<td>10,242</td>
<td>86.76</td>
<td>71.90</td>
</tr>
<tr>
<td>Dirty/DBLP-ACM</td>
<td>12,363</td>
<td>99.03</td>
<td>98.10</td>
</tr>
<tr>
<td>Dirty/DBLP-GoogleScholar</td>
<td>28,707</td>
<td>95.75</td>
<td>93.80</td>
</tr>
<tr>
<td>Dirty/iTunes-Amazon</td>
<td>539</td>
<td>95.65</td>
<td>79.40</td>
</tr>
<tr>
<td>Dirty/Walmart-Amazon</td>
<td>10,242</td>
<td>85.69</td>
<td>53.80</td>
</tr>
<tr>
<td>Textual/Abt-Buy</td>
<td>9,575</td>
<td>89.33</td>
<td>62.80</td>
</tr>
<tr>
<td>Textual/Company</td>
<td>112,632</td>
<td>93.69</td>
<td>92.70</td>
</tr>
</tbody>
</table>

Ditto consistently outperforms DM

More robust to noisy, small, and text-heavy data

Up to 32% F1 improvement (9.43% in average)
Experiments: WDC product datasets (w/ DistillBERT for faster training)

<table>
<thead>
<tr>
<th>Size</th>
<th>Ditto</th>
<th>DeepMatcher</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Small (1/20)</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>computers</td>
<td>80.76</td>
<td>70.55</td>
<td>2834</td>
</tr>
<tr>
<td>cameras</td>
<td>80.89</td>
<td>68.59</td>
<td>1886</td>
</tr>
<tr>
<td>watches</td>
<td>85.12</td>
<td>66.32</td>
<td>2255</td>
</tr>
<tr>
<td>shoes</td>
<td>75.89</td>
<td>73.86</td>
<td>2063</td>
</tr>
<tr>
<td>all</td>
<td>84.36</td>
<td>76.34</td>
<td>9038</td>
</tr>
<tr>
<td><strong>Medium (1/8)</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>computers</td>
<td>88.62</td>
<td>77.82</td>
<td>8094</td>
</tr>
<tr>
<td>cameras</td>
<td>88.09</td>
<td>76.53</td>
<td>5255</td>
</tr>
<tr>
<td>watches</td>
<td>91.12</td>
<td>79.31</td>
<td>6413</td>
</tr>
<tr>
<td>shoes</td>
<td>82.66</td>
<td>79.48</td>
<td>5805</td>
</tr>
<tr>
<td>all</td>
<td>88.61</td>
<td>79.94</td>
<td>25567</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Size</th>
<th>Ditto</th>
<th>DeepMatcher</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Large (1/2)</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>computers</td>
<td>91.70</td>
<td>89.55</td>
<td>33359</td>
</tr>
<tr>
<td>cameras</td>
<td>91.23</td>
<td>87.19</td>
<td>20036</td>
</tr>
<tr>
<td>watches</td>
<td>95.69</td>
<td>91.28</td>
<td>27027</td>
</tr>
<tr>
<td>shoes</td>
<td>88.07</td>
<td>90.39</td>
<td>22989</td>
</tr>
<tr>
<td>all</td>
<td>93.05</td>
<td>89.24</td>
<td>103411</td>
</tr>
<tr>
<td><strong>xLarge (1/1)</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>computers</td>
<td>95.45</td>
<td>90.8</td>
<td>68461</td>
</tr>
<tr>
<td>cameras</td>
<td>93.78</td>
<td>89.21</td>
<td>42277</td>
</tr>
<tr>
<td>watches</td>
<td>96.53</td>
<td>93.45</td>
<td>61569</td>
</tr>
<tr>
<td>shoes</td>
<td>90.11</td>
<td>92.61</td>
<td>42429</td>
</tr>
<tr>
<td>all</td>
<td>94.08</td>
<td>90.16</td>
<td>214736</td>
</tr>
</tbody>
</table>

Ditto already outperforms DeepMatcher when given only 1/2 of training data!
## Ablation Analysis

<table>
<thead>
<tr>
<th></th>
<th>Ditto</th>
<th>Ditto (DA)</th>
<th>Ditto (DK)</th>
<th>Baseline</th>
</tr>
</thead>
<tbody>
<tr>
<td>Structured</td>
<td>88.48</td>
<td>87.98</td>
<td>88.20</td>
<td>85.99</td>
</tr>
<tr>
<td>Dirty</td>
<td>91.33</td>
<td>91.00</td>
<td>90.41</td>
<td>88.39</td>
</tr>
<tr>
<td>Textual</td>
<td>87.52</td>
<td>86.97</td>
<td>87.26</td>
<td>61.37</td>
</tr>
<tr>
<td>WDC_small</td>
<td>83.67</td>
<td>84.36</td>
<td>82.13</td>
<td>81.08</td>
</tr>
<tr>
<td>WDC_xlarge</td>
<td>94.11</td>
<td>94.08</td>
<td>91.78</td>
<td>91.63</td>
</tr>
</tbody>
</table>

- All 3 optimizations are effective
- DK is more effective on the ER-Magellan datasets
- DA is more effective on the WDC datasets
Case study: company matching

- Given two tables A and B of companies, find record pairs that refer to the same company.

<table>
<thead>
<tr>
<th>name</th>
<th>addr</th>
<th>city, state, zip</th>
<th>phone</th>
</tr>
</thead>
<tbody>
<tr>
<td>M-Theory Group</td>
<td>6171 W Century Blvd # 350</td>
<td>Los Angeles, CA 90045-5336</td>
<td>+1.877.682.4555</td>
</tr>
<tr>
<td>M-THEORY CONSULTING GROUP, LLC</td>
<td>6171 W. CENTURY BLVD.</td>
<td>LOS ANGELES, CA 90045</td>
<td>2137858058</td>
</tr>
</tbody>
</table>

- Ditto matches two tables of 789K and 412K entries with 96.5% F1.
The Complete Pipeline with Ditto

Step 1

Table A:

Table B:

Step 2

Step 3

Train Advanced Blocking

Step 4

Matched Pairs

Ditto

Blocker

Candidate Pairs

Serialize

Inject DK

Summarize

Augment

Train
Entity Matching & Deep Learning

- Concurrent work on applying pre-trained LM to EM. Technique is identical to Ditto’s baseline [Brunner, Stockinger EDBT20 (Entity Matching with Transformer Architectures - A Step Forward in Data Integration)]
- RNN based [Mudgal+ SIGMOD18 (Deep Learning for Entity Matching), Ebraheem+ VLDB18 (Distributed representations of tuples for entity resolution)]
- Mitigate data hungry DL based EM solutions:
  - Transfer Learning + Active Learning [Kasai+ACL19 (Low-resource Deep Entity Resolution with Transfer and Active Learning)]
  - Data Augmentation [Miao+SIGMOD21 (Rotom: A Meta-Learned Data Augmentation Framework for EM, Data Cleaning, Text Classification, and Beyond)]
- Contrastive DNN approach [Wang+ ICDM20 (CorDEL: A Contrastive Deep Learning Approach for Entity Linkage)]
- Transformer based Deep Learning models for EM [Tracz+ ACLWorkshop20 (BERT-based similarity learning for product matching)]
  - Bert-based similarity learning for product matching
- The Four Generations of Entity Resolution [Papadakis+ 21 Morgan&Claypool publishers]
- :
Deep Learning & other Data Integration Tasks

- **Information extraction:**
  - **Named Entity Recognition** [Li+ TKDE20 (A survey of DL methods for NER)]
  - **Relation Extraction** [Nayak+ ArXiv21 (Deep Neural approaches to relation triplets extraction)]
  - **Opinion Mining** [Irsoy, Cardie EMNLP14 (Opinion Mining with Deep Recurrent NN)] [Miao+ WWW20 (Snippext: Semi-supervised Opinion Mining with Augmented Data)]
  - **Sentiment Analysis** [Zhang, Wang, Liu Wiley18 (Deep Learning for Sentiment Analysis: A survey)]
Deep Learning & other Data Integration Tasks

- **Table understanding** [Deng+VLDB20 (TURL: Table Understanding through Representation Learning)] [Hulsebos+SIGKDD19 (Sherlock: A Deep Learning Approach to Semantic Data Type Detection.)] [Zhang+VLDB20 (Sato: Contextual Semantic Type Detection in Tables)] [Trabelsi+ arXiv20 (Semantic Labeling Using a Deep Contextualized Language Model)] [Herzig+ ACL20. (Tapas: Weakly supervised table parsing via pre-training)] [Yin+ACL20. (Tabert: Pretraining for joint understanding of textual and tabular data)] [Lockard+arXiv21 (TCN: Table Convolutional Network for Web Table Interpretation)] [Wang+arXiv 20. (Structure-aware Pre-training for Table Understanding with Tree-based Transformers)]

- **Data curation/preparation**
  - [Thirumuruganathan+EDBT20 (Data Curation with Deep Learning)]
  - [Tang+arXiv21 (RPT: Relational Pre-trained Transformer Is Almost All You Need towards Democratizing Data Preparation)]

- **Querying Tables/Text** [Thorne+VLDB21 (to appear) (From Natural Language Processing to Neural Databases)] [Yin+ACL20 Tabert: Pretraining for joint understanding of textual and tabular data]
Effectiveness of Deep Learning in Data Integration

- Suitable for tasks where rules are difficult to specify, features are hard to engineer
  - Many data integration problems are like this
  - Variations and nuances in language, heterogeneity in content and structure, dirty data, context
- Robust to data imperfections
  - Can deal with missing or wrong values, missing meta-data, heterogeneous data
Effectiveness of Deep Learning in Data Integration

- Immense language understanding
  - Pre-training:
    - Lower layers capture lexical structure.
    - Higher layers capture more semantic properties of a language
    - Deeper layers track longer-distance linguistic dependencies
    - BERT representations capture linguistic information in a compositional way that mimics classical, tree-like structures

- Difference between “Sharp TV” vs “Sharp resolution”
- Similarity between “Stop hair loss” vs “Prevents thinning hair”
Effectiveness of Deep Learning in Data Integration

- Immense ability to learn from examples. Attention is key

<table>
<thead>
<tr>
<th>title</th>
<th>manf./modelno</th>
<th>price</th>
</tr>
</thead>
<tbody>
<tr>
<td>instant immersion spanish deluxe 2.0</td>
<td>topics entertainment</td>
<td>49.99</td>
</tr>
<tr>
<td>adventure workshop 4th-6th grade 7th edition</td>
<td>encore software</td>
<td>19.99</td>
</tr>
<tr>
<td>sharp printing calculator</td>
<td>sharp el1192bl</td>
<td>37.63</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>title</th>
<th>price</th>
</tr>
</thead>
<tbody>
<tr>
<td>instant immers spanish dlux 2</td>
<td>36.11</td>
</tr>
<tr>
<td>encore inc adventure workshop 4th-6th grade 8th edition</td>
<td>17.1</td>
</tr>
<tr>
<td>new-sharp shr-el1192bl two-color printing calculator 12-digit lcd black red</td>
<td>56.0</td>
</tr>
</tbody>
</table>
What is the catch?

- Data hungry
  - Quality of a DL model is directly dependent on its training data

From Andrew Ng’s NN and Deep Learning course

- Traditional ML models’ performance plateaus with more training data
- Larger NN tends to perform better with more training data
What is the catch?

- Data hungry
  - Quality of a DL model is directly dependent on its training data
  - The more training data, the better.

- Quality training data is expensive to obtain
  - Often a significant data integration problem
Disadvantages of using Deep Learning for Data Integration

- Data hungry
  - Quality of a DL model is directly dependent on its training data
  - The more quality training data, the better
  - Quality training data is expensive to obtain
  - Fairness/Bias in training data
- Requires high performance hardware
- Longer latency. Expensive to deploy
- Complex: lots of hyperparameters (BERT-base 110M, BERT-large 340M)
- Opaque
Challenges and Opportunities

- **Benchmarks for DI tasks**
  - Comprehensive benchmarks for data cleaning, table understanding, entity matching etc.
  - E.g., EM: include numerical heavy data, different types of dirty data and include metrics for measuring fairness/biasness in data
- **Techniques to mitigate data hungry DL solutions:**
  - Data Augmentation: generate additional training data fairly
  - Relational
  - Transfer learning, Active Learning, Weak supervision
Challenges and Opportunities

- **Model Explainability:**
  - Explain the results of your DI tasks
  - Generate rules for the DI task which are also explainable
  - Explain a model’s decision. E.g., LIME: Local Interpretable Model Agnostic Explanations
    - Generate explanations for why and why-not questions

- Querying heterogeneous heterogeneous data (different structure, different modalities)
  - Query data “outside the box”
    - Structured data/text/images/audio/video in a virtual DI setting
Andrew Ng on MLOps: From Model-centric to Data-centric AI

(March 2021)

“When a system isn’t performing well, many teams instinctually try to improve the code. But for many practical applications, it’s more effective instead to focus on improving the data.”

“If Google has BERT then OpenAI has GPT-3. But, these fancy models take up only 20% of a business problem. What differentiates a good deployment is the quality of data; everyone can get their hands on pre-trained models or licensed APIs.”
Can we integrate data for social good?

- World today:
  - Content: text/images/audio/video

- Can we integrate data to understand the world for a variety of purposes?
  - Understand the origins of content
  - Understand the entities and relationships between entities in the content, and related content
  - Understand the meaning or intent of content
Acknowledgements

Deep thanks to:
Yuliang Li, Jinfeng Li, Yoshihiko Suhara (Megagon Labs)
AnHai Doan (University of Wisconsin)

Some slides on Entity Matching are originally from Yuliang Li

Alon Halevy (Facebook AI)