UNIVERSITY OF MANNHEIM

Team Project HWS 2021

Data Integration using Deep Learning

Reminder: Corona Check-in

Don’t forget to check in!

If you haven’t done so already, please visit http://checkin.uni-mannheim.de/ and check in for this meeting

Hallo

- **Prof. Dr. Christian Bizer**
- Professor for Information Systems V
- Research Interests:
  - Web Data Integration
  - Data and Web Mining
  - Adoption of Data Web Technologies
  - Knowledge Base Construction
- Room: B6 - B1.15
- eMail: chris@informatik.uni-mannheim.de
- Consultation: Wednesday, 13:30-14:30

Hallo

- **Ralph Peeters**
- Graduate Research Associate
- Research Interests:
  - Product Data Integration
  - Entity Matching using Deep Learning
- Room: B6, 26, C 1.04
- eMail: ralph@informatik.uni-mannheim.de

Agenda of Today‘s Kickoff Meeting

1. You and Your Experience
2. Motivation and Project Goals
3. The WDC Schema.Org Table Corpus
4. Organization
5. Specific Subtasks
6. Schedule
7. Formal Requirements

You and Your Experience

- A Short Round of Introductions
  - What are you studying? Which semester?
  - Which DWS courses did you already attend?
  - What are your programming and data wrangling skills?
  - Did you already work on any data integration or cleansing projects?
- Participants

1. Jennifer Hahn
2. Jannik Reißfelder
3. Wafaa Ibrahim Mahmoud AbuObidalla
4. Kim-Carolin Lindner
5. Niklas Sabel
6. Cheng Chen
7. Marvin Rösel
8. Estelle Weinstock
9. Luisa Theobald

Motivation of the Team Project

The Web is a rich source of (tabular) data

- Tables contain information about various subjects
  - Webtables
  - Wiki(pedia) tables

Actor		Born	Died	Age	Nomina-tions	Wins	Lead and sup-porting details	First winning film role or first nomination (also see list of all nominated roles)	First year	Last year
Barkhad Abdi	M	1985	~	36	1	0	S	*Captain Phillips*	2013	2013
F. Murray Abraham	M	1939	~	81	1	1	L	*Amadeus*	1984	1984
Amy Adams	F	1974	~	47	6	0	1L:5S	*American Hustle* (2013)	2005	2018
Nick Adams	M	1931	1968	36	1	0	S	*Twilight of Honor*	1963	1963
Isabelle Adjani	F	1955	~	66	2	0	L	*Story of Adele H., The*	*1975*	1989
Casey Affleck	M	1975	~	46	2	1	1L1:1S0	*Manchester by the Sea*	2007	*2016*
Shohreh Aghdashloo	F	1952	~	69	1	0	S	*House of Sand and Fog*	2003	2003
Brian Aherne	M	1902	1986	83	1	0	S	*Juarez*	1939	1939

Wikipedia Table listing Academy Award nominees

- Furthermore, annotated entities (e.g via schema.org) can be used to derive tables
  - we can build tables from web crawls using this information
  - For some entity types unique identifiers may be annotated, allowing us to link them across websites (e.g. GTIN for products, ISBN for books, etc.)

Many websites describe different aspects of the same entities

- Aggregating information across different tables helps us to build richer knowledge and solve business problems, e.g. product catalogs, yellow pages/maps, knowledge graphs
- Problem: Heterogeneity between table representations makes integration hard
  1. Schema of tables different or semantics are unclear (attr1, attr2, etc.)
  2. Entity representations across tables vary (abbreviations, erroneous values, etc.)

Task 1: Schema Matching

Schema Matching: Automatically or semi-automatically discover correspondences between schemata.

→ Match columns describing the same attribute across tables

Challenges:

- Semantic heterogeneity (synonyms, homonyms, normalization)
- Generic names (attr1, attr2, attr3)
- Esoteric naming conventions and different languages

Most methods focus on 1:1 correspondences

- Also 1:n and n:1 possible, e.g. actor name in table 1 vs firstname and lastname in table 2
- We will focus solely on 1:1 for this project

Task 2: Entity Matching

Entity Matching: Identify all records in all data sources that describe the same real-world entity.

→ Match rows across (and inside) tables describing the same real world entity

Challenges:

- Semantic heterogeneity (differences in meaning)
  - Synonyms, homonyms, vague entity names
  - Different surface forms
  - Conflicting data
- Quadratic runtime complexity when comparing everything

Features that help us distinguish entities

- identifiers (GTINs, MPNs, ISBNs, …)
- titles (entity name and maybe selected features)
- identifying attributes (e.g. RAM size, phone numbers, number of pages)

Difficulty of the Task also depends on the Entity Type

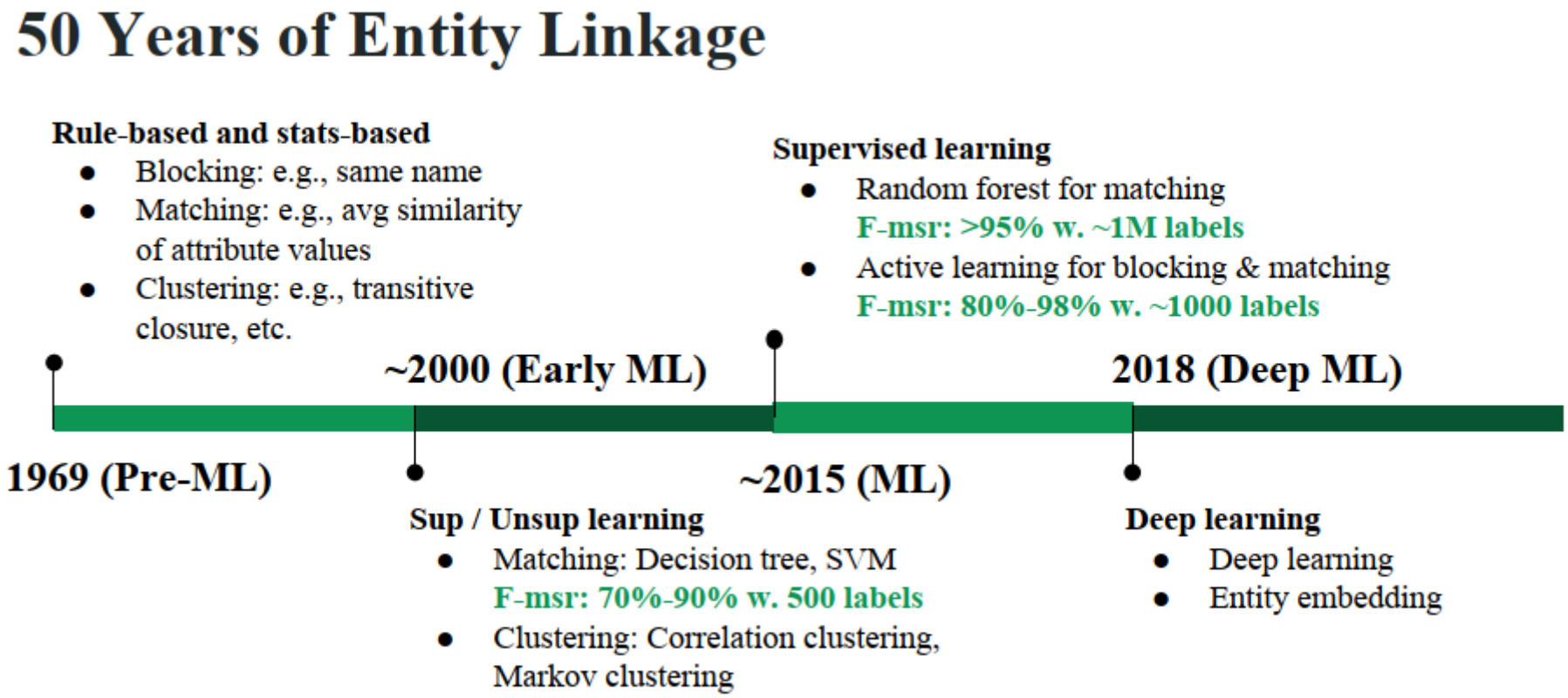
- Books
  - wide adoption of identification schema (ISBNs)
  - Entity matching problem mostly solved ☺
  - Schema matching often solvable via duplicate-bases methods
  - other features like title and author often only used for sanity checks

- Phones / Computers / Cameras
  - rather structured features
  - easier to find identifying combination of features
  - different tables often share attributes

- Cloths / Bags / ….
  - rather unstructured descriptions, not too many features
  - only weak agreement of attributes

  ➔ Methods focused on text need to be applied

Entity Matching over Time



Source: https://thodrek.github.io/di-ml/sigmod2018/sigmod2018.html

Schema Matching has a similar history.

Rahm: A Survey of Approaches to Automatic Schema Matching. VLDB 2001 + follow up article 10 years later.

Transformers in NLP

Transformer architectures like BERT had large impact on NLP

- stacked encoder layers with self-attention mechanism
- every token can attend to every other token in both directions
- contextual representations, i.e. embedding depends on context
- multiple attention heads allow learning of different concepts

Pre-training / Fine-tuning paradigm

- pre-training: self-supervised general (masked) language modeling
- fine-tuning: Further training on task-specific data

→ shown to work extremely well for a variety of problems

→ including entity matching ☺

→ Growing corpus of work regarding data integration using Transformers (see references)

Some Table-based Transformers (more in the references)

- **TURL** (http://www.vldb.org/pvldb/vol14/p307-deng.pdf)
  - General pre-training on ~600k Wikipedia tables
  - Using MLM and a Masked Cell Entity Recovery objective (have labels for matching entities across tables)
  - Evaluated on Entity Linking, Column Type Annotation, Column Relation Prediction, Row Population and Cell Filling
  - Code available
- **TABBIE** (https://aclanthology.org/2021.naacl-main.270.pdf)
  - General pre-training on 1.8M Wikipedia Tables and 24.8M WebTables
  - Using a cell corruption objective: Replace some cell contents with frequency-based cell sampling across all tables and then try to predict if a cell was changed or not
  - Evaluated on Column Type Annotation, Row Population, Column Population
  - Code available

Some Table-based Transformers (more in the references)

- **DoDuo** (https://arxiv.org/abs/2104.01785)
  - Default BERT no (further) pre-training, only fine-tuning
  - Uses multi-task training (Column Type Prediction and Column Relation Prediction)
  - Evaluated on Column Type Prediction and Column Relation Prediction
  - Code not yet available
- **RPT** (https://arxiv.org/abs/2012.02469)
  - Encoder-Decoder Transformer (based on BART)
  - Pre-training by masking tokens and subsequently trying to decode the correct values
  - Short evaluation on Pre-training task, no downstream application yet
  - Code not yet available
- **HTT**
  - Hierarchical Table Transformer model we are currently working on
  - Pre-training using cell corruption similar to TABBIE
  - Evaluation currently in progress (Column Type Annotation, Column Relation Prediction)

Project Goal

Explore performance of Transformer-based table models for the tasks of entity and schema matching.

- How does the performance compare to other neural and non-deep learning state-of-the-art methods?
- Where do these models excel and what are their weaknesses (error analysis)?
- Is general pre-training on tabular data necessary for good performance on these tasks or is the knowledge contained in language models like BERT enough?

Project Goal

1. Profile schema.org tables from a large number of websites and build train / test sets for experimental evaluation for entity and schema matching
2. Experiment with state-of-the-art tabular transformers and baselines for both tasks
3. Compare performance of methods w.r.t.:
   - Simple Baselines (e.g. Random Forest / SVM)
   - Standard textual Transformers
   - Usage of general tabular pretraining
   - (Subject-)Entity categories
4. Try to explain why models perform better than others
   - Classify test examples into a set of matching challenges
   - Conduct an error analysis

Learning Targets

Improve your technical skills

- Work as a **Data Scientist**: clean, profile, integrate, classify data
- Understand the nature of **Web Data**
- Improve your technical expertise concerning **Deep Learning**
- Improve your programming skills

Improve your soft skills

- Work as part of a bigger team on a more complex project
- Organize yourself and assign tasks based on your skills
- Communicate and coordinate your work

The WDC Schema.Org Table Corpus

1. Semantic Annotations in HTML Pages
2. Web Data Commons Project
3. Web Data Commons – Schema.org Table Corpus

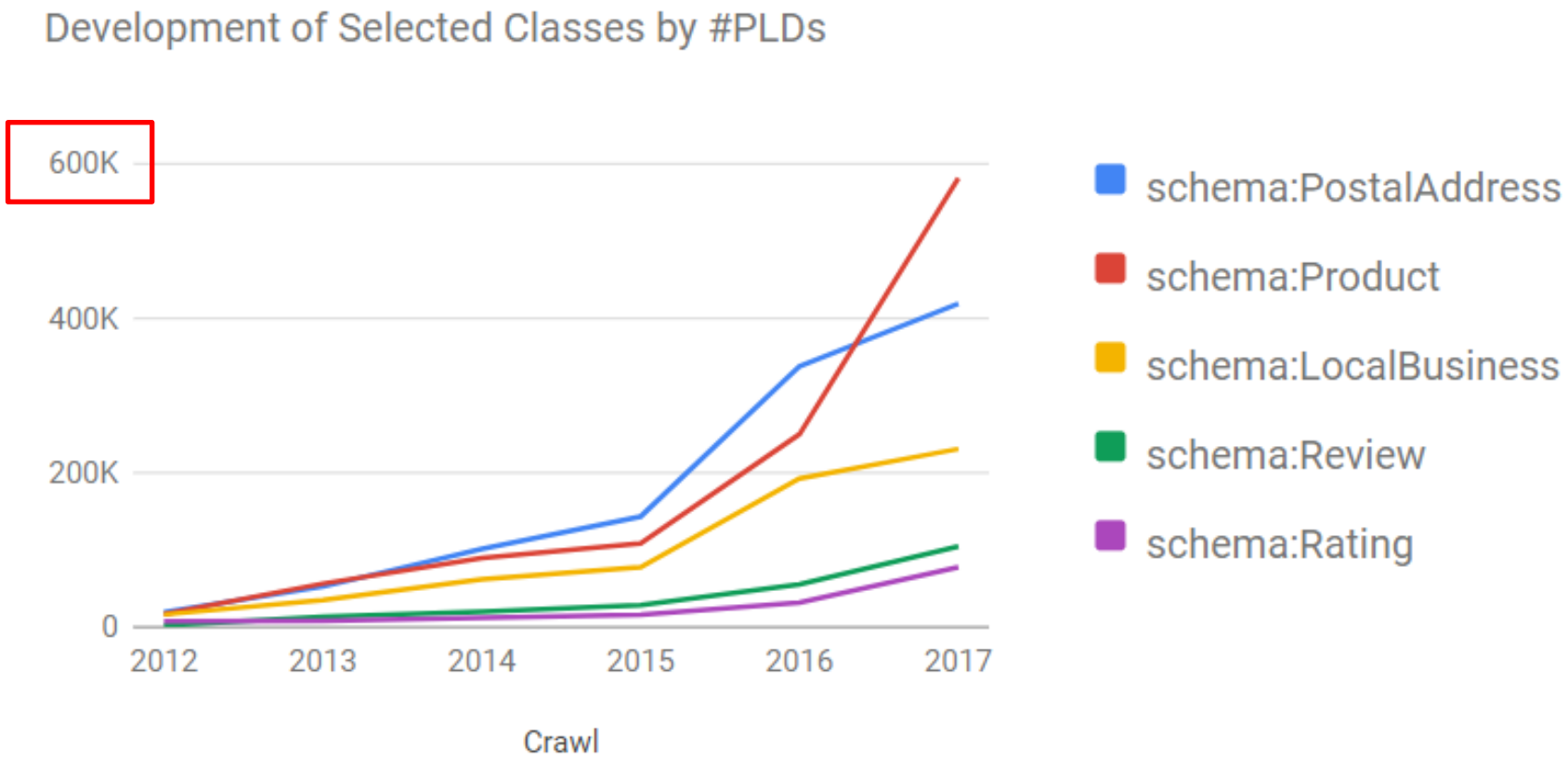
Semantic Annotation of HTML Pages: Schema.org

- ask site owners since 2011 to annotate data for enriching search results
- 675 Types: Event, Place, Local Business, Product, Review, Person
- Encoding: Microdata, RDFa, JSON-LD

Example: Microdata Annotations in HTML


```
<div  itemtype="http://schema.org/Product">
  <span itemprop="name">Sony GTK-XB5L Audiosystem</span>
  <span itemprop=“gtin13“>04048945021687</span>
  <span itemprop=“description">high-power home audio system with Bluetooth technology</span>
</div>
 <div itemprop="aggregateRating" itemscope itemtype="http://schema.org/AggregateRating">
    <span itemprop="ratingValue"> 4 </span> stars-based on
    <span itemprop="reviewCount"> 250 </span> reviews.
 </div>
```

schema.org Annotations: Most Popular Classes



http://webdatacommons.org/structureddata/

Properties used to Describe Products 2017

Top 15 Properties	PLDs	
	#	%
schema:Product/name	535,625	92%
schema:Offer/price	462,444	80%
schema:Product/offers	462,233	79%
schema:Offer/priceCurrency	430,556	74%
schema:Product/image	419,391	72%
schema:Product/description	377,639	65%
schema:Offer/availability	337,876	58%
schema:Product/url	263,720	45%
schema:AggregateRating/ratingValue	184,004	32%
schema:Product/sku	126,696	22%
schema:AggregateRating/reviewCount	112,408	19%
schema:Product/aggregateRating	101,434	17%
schema:Product/brand	73,934	13%
schema:Product/productID	35,211	6%
schema:Product/manufacturer	21,967	4%

Samsung Galaxy S4 Verizon AT&T T-Mobile GSM Unlocked Smartphone SRF

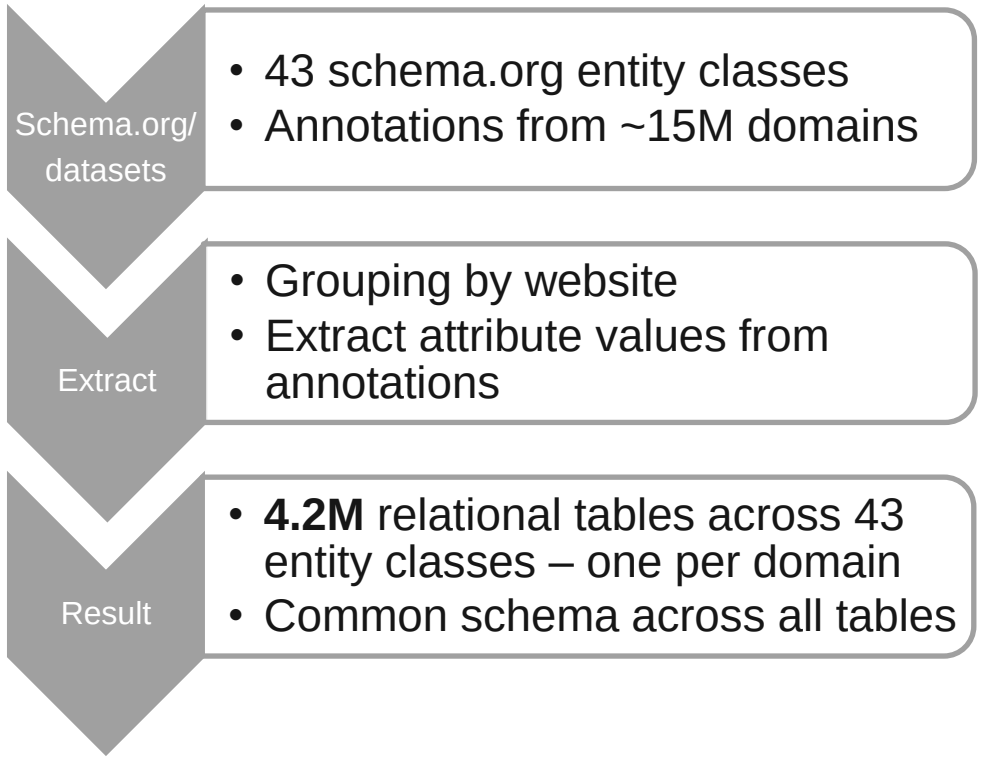
Das Samsung Galaxy S4 ist der unterhaltsame und hilfreiche Begleiter für Ihr mobiles Leben. Es verbindet Sie mit Ihren Liebsten. Es lässt Sie gemeinsam unvergessliche Momente erleben und festhalten. Es vereinfacht Ihren Alltag.

UPC 610214632623

UPC 610214632623

The WDC Schema.Org Table corpus

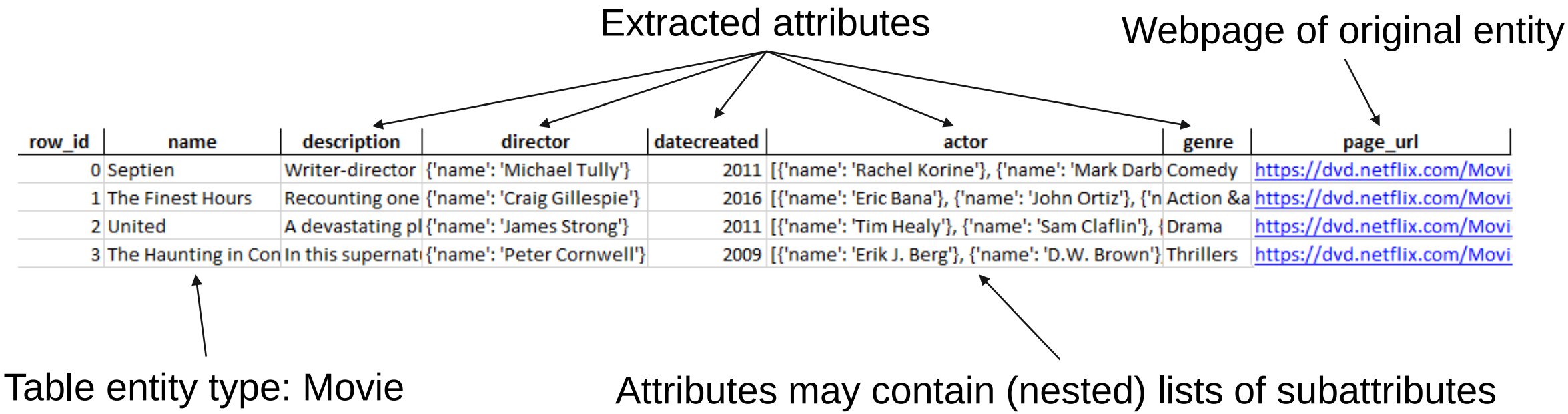
- Extract annotations for 43 schema.org entity classes
- Extract attribute values and group entities by website
- Initial cleaning steps are performed
- **Final result**: 4.2M tables - one table per domain, containing all annotated entities after cleaning.
- All tables share the same schema



Details and Download:

- **http://webdatacommons.org/structureddata/schemaorgtables/**

An Example Table from the Corpus



row_id	name	description	director	datecreated	actor	genre	page_url
0	Septien	Writer-director	{'name': 'Michael Tully'}	2011	[{'name': 'Rachel Korine'}, {'name': 'Mark Darb	Comedy	https://dvd.netflix.com/Movi
1	The Finest Hours	Recounting one	{'name': 'Craig Gillespie'}	2016	[{'name': 'Eric Bana'}, {'name': 'John Ortiz'}, {'n	Action &a	https://dvd.netflix.com/Movi
2	United	A devastating pl	{'name': 'James Strong'}	2011	[{'name': 'Tim Healy'}, {'name': 'Sam Claflin'}, {	Drama	https://dvd.netflix.com/Movi
3	The Haunting in Con	In this supernat	{'name': 'Peter Cornwell'}	2009	[{'name': 'Erik J. Berg'}, {'name': 'D.W. Brown'}	Thrillers	https://dvd.netflix.com/Movi

Corpus Statistics for Top 10 Entity Type

	Overall		Top 100				Minimum3			Rest	
Schema.org Class	# tables	# rows	# tables	# rows	median rows per table	# rows in largest table	# tables	# rows	median rows per table	# tables	# rows
Product	2,028,974	231,742,974	100	4,481,576	27,436	391,572	1,662,173	226,782,151	24	366,701	479,247
Person	921,777	6,644,475	100	1,256,440	8,193	101,758	188,361	4,514,105	5	733,316	873,930
LocalBusiness	465,816	7,410,544	100	1,114,508	9,420	29,007	50,506	5,858,029	14	415,210	438,007
CreativeWork	252,106	15,482,053	100	1,339,782	10,873	102,056	173,855	14,039,752	14	78,151	102,519
Event	229,980	7,347,918	100	1,067,749	6,239	222,817	143,898	6,165,838	9	85,982	114,331
Place	76,679	2,575,159	100	607,337	4,878	22,849	27,355	1,909,471	9	49,224	58,351
Restaurant	44,486	889,801	100	410,985	2,720	22,481	6,470	438,654	15	37,916	40,162
Recipe	39,246	4,102,300	100	796,503	6,636	26,909	30,733	3,294,594	22	8,413	11,203
JobPosting	33,570	3,061,693	100	590,494	4,257	31,424	23,497	2,458,033	15	9,973	13,166
Hotel	25,528	1,685,008	100	792,301	5,140	79,544	13,014	877,856	11	12,414	14,851

- Tables are sorted into three groups:
  - Top100: the 100 largest tables of an entity type
  - Minimum 3: tables containing at least 3 entries
  - Rest: any smaller tables

Attribute statistics for interesting entity types

Product		Local Business		Hotel		Movie	
column	occurs in % of tables	column	occurs in % of tables	column	occurs in % of tables	column	occurs in % of tables
name	99	name	100	name	100	name	97
offers	87	address	92	address	91	description	86
description	78	telephone	68	aggregaterating	75	director	85
brand	62	geo	41	pricerange	62	aggregaterating	68
sku	47	aggregaterating	39	description	58	actor	61
aggregaterating	27	description	38	starrating	49	datecreated	52
productid	20	url	27	geo	49	duration	39
image	16	pricerange	22	telephone	33	genre	33
url	15	image	14	checkintime	28	datepublished	28
mpn	14	review	14	reviews	20	image	20

- 3-4 attributes filled in most tables
- Some attributes that can help with finding matching entities
- Consult statistics files for per-table stats and more (available for download on the corpus website)

Team Project Organization

Duration: 6 months (01.10.2021 – 31.03.2022)

Participants: 9 people

Type of work: Team and subgroup based

Milestones: 4 project phases

ECTS Points: 12

Evaluation

- Intermediate presentations
- Final report
- Individual contribution to the deliverables is graded

Questions and Subtasks

1. Which entity types to consider? → **Data Selection and Profiling I**
2. Which tables to select (enough train/test tables available?) → **Data Selection and Profiling II**
3. How to split tables for training and testing sets? → **Data Selection and Profiling II**
4. Which baseline methods to consider? → **Experiments I & II**
5. Which Transformer-based models to try? → **Experiments I & II**
6. How do the different methods compare? → **Experiments I & II**
7. How important is general pretraining on tabular data? **-> Experiments I & II**
8. Are tabular entity representations helpful for pairwise matching? **-> Experiments I & II**
9. What can you learn from the results? → **Evaluation & Explanation**
10. How can you explain the results? → **Evaluation & Explanation**

Phase 1a: Data Selection and Data Profiling I

Participants: all team members / two subgroups (one subgroup per task)

Duration: 01.10. – 15.10.

Deliverables: 15 min. presentation to Ralph, code & data

Input: WDC Schema.Org Table Corpus (*provided here*)

Goal: Identify at least **2** entity types for entity matching / at least **20** for schema matching

Tasks:

1. (Entity Matching) Find **2** entity types for which:
   - **enough** (see slide 34) tables and overlap exist
   - Allow you to easily find matches across tables (see next slides)
2. (Schema Matching) Find **20** entity types for which:
   - Schemata are (partly) ambigous (see next slides)
3. Profile the tables of the selected types for both tasks and present the results

Phase 1a: Entity Matching - How to get started?

- Get the Tables (here)
- Get acquainted with data, attributes, etc. (have a look at the statistics files)
- Look at **product tables** first and get the mapping of table rows to entity clusters here. Every row containing the same real world entity will be identifiable by the same cluster_id. More information about the clustering is available here.
- For possible additional mappings not covered, look for product ID attributes in the tables and match them.
- Profile the tables and estimate the overlap (how many products are described by how many tables?)
- Find a second entity type (Hint: try **LocalBusiness**, match using phone number)

Phase 1a: Schema Matching - How to get started?

- Get the Tables (here)
- Get acquainted with data, attributes, etc. (have a look at the statistics files)
- As all tables already have the same schema it is easy for you get labels for matching columns across tables (but we assume Product/name to be a different label than e.g. Hotel/name)
- We will be doing **instance-based** schema matching in the experiments, meaning that we try to match columns using the attribute values, not the header name, which we assume to be generic.
- Find at least 20 entity types for which some schema instances look similar

→ e.g. Person/name vs. Movie/director

- Profile the corresponding tables and present statistics about size and similarity of values

Languages and how to detect them

- Problem: Tables may contain languages other than English
- Remove any tables that do not contain English entities
- Remove any non-English rows from the remaining tables
- How to detect languages?
  - Use 3-step approach:
    1. Use TLD for selection of relevant tables
    2. Apply language detection algorithm of your choice to rows of the tables
    3. Remove step 1. and run 2 on all tables if filtering by TLD is too harsh
  - Example algorithms:
    - Dictionary-based approach (likely too simple)
    - Machine-learning approach like fastText language detection

Phase 1a: Deliverables

- Statistics about the data you have selected:
  - Which entity types did you choose and why?
  - How many relevant tables remain per type?
  - (entity matching) What is the distribution of matching entities across the tables?
  - (schema matching) What are the ambiguous columns across types?
  - What is the table size distribution?
  - (entity matching) Example of 3 hard matches and 3 hard non-matches for entities per entity type
  - (schema matching) Example of 6 ambiguous column types across entity types
- Must haves:
  - Entity and column type distribution histograms

Phase 1b: Data Selection and Profiling II

Participants: two subgroups (one subgroup per task)

Duration: 15.10. – 8.11.

Deliverables: 30 min. presentation to Chris and Ralph, code & data

Input: Selected entity types and corresponding tables

Tasks:

1. select tables following the rules on next slides
2. Build training sets and testing sets for both tasks
   - Framing the entity matching problem as multi-class classification (row has corresponding entity label) and pair-wise classification (two entities form a pair and are labeled as match or non-match)
   - Framing the schema matching problem as multi-class classification (column has corresponding column type label)
3. Downsample training set so you have 3 different sizes: small, medium, large

Phase 1b: Entity Matching - How to select suitable tables? (multi-class)

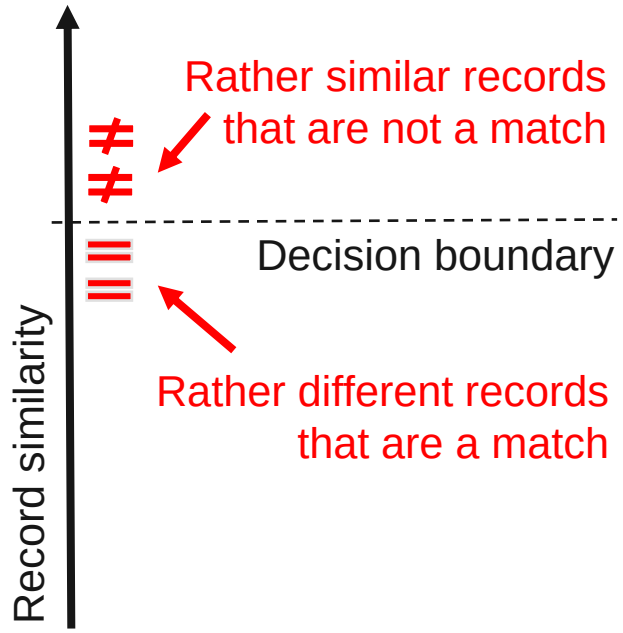
- Try to find tables such that you have at least 150 entities per entity type:
- The table set should:
  - Contain at least 15 (10 training / 5 testing) tables which contain entity descriptions for each of the 150 entities
  - Make sure that for every entity you select, you also select at least 3 very similar entities (e.g. *iPhone 6* vs *iPhone 6s*)

    → Hard to distinguish corner-cases, otherwise the matching can be trivial
  - important: You have to manually check that the testing offers actually describe the correct entity, otherwise you will introduce noisy labels in your testing data!
- Ways to find similar entities:
  - Use keyword search, e.g. model name for products
  - Calculate similarity metric and order by similarity
  - Exploit IDs in the data, e.g. GTINs, Phone numbers
- Try to get as much data as possible!

Phase 1b: Entity Matching - How to get good training data (pair-wise)?

From the tables/entities you selected for the multi-class case:

- Build Training data which should:
  1. be balanced as random pairs would be highly skewed towards non-matches
  2. contain corner cases as they are most informative
     - especially “near-miss” negative examples are more informative for training than randomly selected pairs which tend to be “easy” non-matches.
     - iPhone 6 vs. iPhone 6s
     - rule of thumb: 50% corner cases
     - match offers using several simple matching techniques (e.g. Jaccard) and sort offer pairs according to their similarity
- You can automatically assemble this training set using the supervision from the multi-class set
- You do not need to manually check each pair



Phase 1b: Entity Matching - How to get good testing data (pair-wise)?

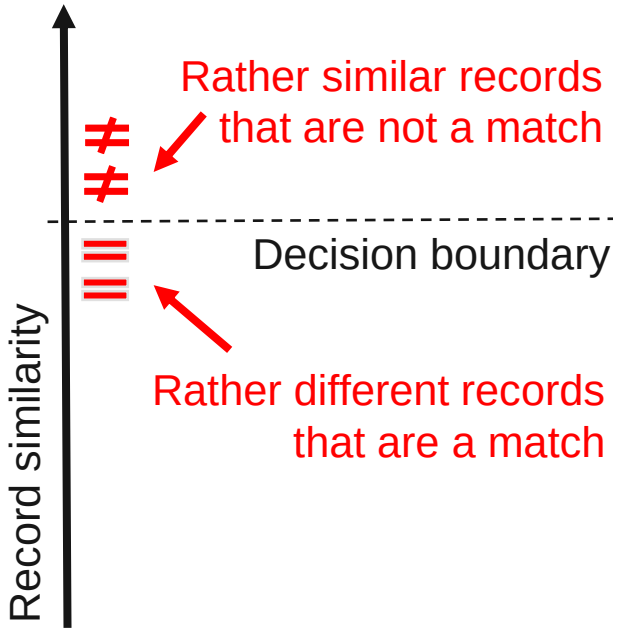
From the tables/entities you selected for the multi-class case:

- manually label a set of record pairs (e.g. 500 pairs) including corner cases

- Use the following rule of thumb:
  1. matching record pairs (25% of GS)
  2. non-matching record pairs (75% of GS)

  → 50% of each group corner-cases
  the other 50% random pairs

- You have to verify these pairs manually!



Result: Training and Test sets for the pairwise matching task

Phase 1b: Schema Matching - How to select suitable tables?

- Try to find tables such that you have at least 200 column type labels
- The table set should:
  - Contain at least 15 (10 training / 5 testing) tables for each of the 200 column type labels
  - Make sure that for every column type you select, you also select at least 3 similar but different column types (value-wise, can be from same entity type or different one)

    → Hard to distinguish corner-cases, otherwise the matching can be trivial
  - important: You have to manually check that the testing columns actually describe the correct column type, otherwise you will introduce noisy labels in your testing data!
- Ways to find similar columns:
  - Compare average length of values to narrow down candidates
  - Calculate similarity metric and order by similarity
- Try to get as much data as possible!

Phase 1b: Downsampling Training sets

- Build the first training set with as much data as possible!
  - This will be your large training set
- Downsample the large training set in a stratified fashion
  - To collect the medium dataset
- Downsample the medium dataset in a stratified fashion
  - To collect the small dataset

→ Every larger set contains the smaller sets as subsets while generally keeping the distribution intact

- Rough guidelines for training set size:
  - Small: low thousands up to 2.5k samples
  - Medium: high thousands up to 10k samples
  - Large: low ten thousands up to 50k samples

Phase 2a: Experiments I

Participants: two subgroups (one subgroup per task)

Duration: 8.11. – 3.12.

Deliverables: 20 min. presentation to Ralph, code & data

Input: Train and test sets from phase 1

Tasks:

1. Apply simple baseline models to both tasks: Random Forest with TFIDF features, standard Transformer model, e.g. RoBERTa
2. Try TURL/TABBIE/HTT for both problems (HTT likely with domain-specific pre-training)
3. Use entity representations from two models as input for pair-wise matching algorithms
4. Come up with an experiment plan outlining the specific experiment setup **and send it to us by 11.11.**
5. Implement methods and start running experiments once you get our OK

Phase 2b: Analysis and Refinement

Participants: two subgroups (one subgroup per selected task)

Duration: 3.12. – 10.01.

Deliverables: 30 min. presentation to Chris and Ralph, code & data

Input: Results from Phase 2a

Tasks:

1. Check quality of created evaluation sets
   - Too easy? Too hard?
   - Think about refinement of dataset building / model training procedure
2. Come up with a refinement plan **and send it to us by 8.12.**
3. Refine datasets and experimental setup
   - E.g. add harder/more similar samples to increase difficulty
   - Try different models / adapt pre-training domains (sets)

Phase 3: Experiments II

Participants: two subgroups (one subgroup per task)

Duration: 10.01. – 04.02.

Deliverables: 20 min. presentation to Ralph, code & data

Input: Train and test sets from phase 2b

Tasks:

1. Rerun models
   - With implemented changes from phase 2b
   - On revised datasets
2. Run additional models
   - Which sound promising (e.g. latest research)
   - Maybe adapt existing model based on phase 2b

Phase 4: Evaluation and Explanation

Participants: two subgroups (one subgroup per selected task)

Duration: 04.02 – 11.03

Deliverables: 30 min. presentation to Chris and Ralph, code & data

Input: Results from Phase 3

Tasks:

1. Error analysis
   - Look at correctly/incorrectly classified examples for select models
   - Think about specific error classes and classify examples accordingly
   - Calculate statistics for error classes
2. Explanation
   - For what kind of problem does which algorithm perform better and why?
   - What do you conclude about the usability of transformers for table-based matching and future work?

Schedule

Date	Session
Thursday, 30.09.2021	Kickoff meeting (today)
	Phase 1a (in 2 subgroups): Data Selection and Data Profiling I
Friday, 15.10.2021	1st Deliverable: 15 minutes presentation, code & data
	Phase 1b (in 2 subgroups): Data Selection and Data Profiling II
Monday, 25.10.2021	Meet Ralph and report current plan and results
Monday, 8.11.2021	2nd Deliverable: 30 minutes presentation, code & data
	Phase 2a (in 2 subgroups): Experiments I
Friday, 12.11.2021	Meet Ralph and report current plan and results
Friday, 3.12.2021	3rd Deliverable: 20 minutes presentation, code & data
	Phase 2b (in 2 subgroups): Analysis and Refinement
Monday, 10.01.2022	4th Deliverable: 30 minutes presentation, code & data
	Phase 3 (in 2 subgroups): Experiments II
Friday, 04.02.2022	5th Deliverable: 20 minutes presentation, code & data
	Phase 4 (in 2 subgroups): Evaluation and Explanation
Friday, 25.02.2022	Meet Ralph and report current plan and results
Friday, 11.03.2022	6th Deliverable: 30 minutes presentation, code & data
Thursday, 31.03.2022	Final Report Submission

Formal Requirements & Consultation

Deliverables

1. On the deliverable dates provide us via e-mail with:
   - **Presentation slides**
   - **Task to member report**: excel sheet stating which team member conducted which subtask
   - **Code/Data**: link or zipped folder with your code and data

2. **Final Report**
   - **15 pages** including appendices, not including the bibliography
   - every additional page reduces your grade by 0.3
   - Created with Latex template of the Data and Web Science group (https://www.uni-mannheim.de/dws/teaching/thesis-guidelines/)

All deliverables should be sent to Chris & Ralph!

Formal Requirements & Consultation

Final grade

- 20% for each phase
- 20% for final report
- Late submission: -0.3 per day

Consultation

- Send one e-mail per team or subgroup stating your questions to Ralph

Useful Software

- Transformers code
  - HuggingFace Transformers: https://huggingface.co/transformers/
  - TURL: https://github.com/sunlab-osu/TURL
  - TABBIE: https://github.com/SFIG611/tabbie
- Processing and GPUs
  - Google Colab: https://colab.research.google.com/
  - BwUniCluster2.0: https://wiki.bwhpc.de/e/Category:BwUniCluster_2.0
- Team Cooperation
  - GitHub/Lab for the code base
  - Video-Chat application of your choice
  - Project Management Tool of your choice

Related Work: (Tabular) Transformers (1/2)

- X. Deng, H. Sun, A. Lees, Y. Wu, and C. Yu, “TURL: table understanding through representation learning,” Proc. VLDB Endow., vol. 14, no. 3, pp. 307–319, Nov. 2020.
- Y. Suhara et al., “Annotating Columns with Pre-trained Language Models,” arXiv:2104.01785 [cs], Apr. 2021
- H. Iida, D. Thai, V. Manjunatha, and M. Iyyer, “TABBIE: Pretrained Representations of Tabular Data,” arXiv:2105.02584 [cs], May 2021
- N. Tang et al., “RPT: Relational Pre-trained Transformer Is Almost All You Need towards Democratizing Data Preparation,” arXiv:2012.02469 [cs]
- P. Yin, G. Neubig, W. Yih, and S. Riedel, “TaBERT: Pretraining for Joint Understanding of Textual and Tabular Data,” arXiv:2005.08314 [cs], May 2020
- J. Herzig, P. K. Nowak, T. Müller, F. Piccinno, and J. Eisenschlos, “TaPas: Weakly Supervised Table Parsing via Pre-training,” in Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, Jul. 2020, pp. 4320–4333.
- D. Wang, P. Shiralkar, C. Lockard, B. Huang, X. L. Dong, and M. Jiang, “TCN: Table Convolutional Network for Web Table Interpretation,” arXiv:2102.09460 [cs], Feb. 2021

Related Work: (Tabular) Transformers (2/2)

- Z. Wang et al., “TUTA: Tree-based Transformers for Generally Structured Table Pre-training,” in Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, New York, NY, USA, Aug. 2021, pp. 1780–1790.
- Ralph Peeters, Christian Bizer, and Goran Glavaš. 2020. Intermediate Training of BERT for Product Matching. In DI2KG Workshop @ VLDB2020.
- Yuliang Li, Jinfeng Li, Yoshihiko Suhara, AnHai Doan, and Wang-Chiew Tan. 2020. Deep Entity Matching with Pre-Trained Language Models. arXiv:2004.00584 [cs] (April 2020).
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, et al. 2017. Attention Is All You Need. In Proceedings of the 31st International Conference on Neural Information Processing Systems. 6000–6010.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. 4171–4186.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, et al. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. arXiv:1907.11692 (2019).

Related Work: General (Deep) Entity Matching (1/2)

- Anna Primpeli, Ralph Peeters, and Christian Bizer. 2019. The WDC Training Dataset and Gold Standard for Large-Scale Product Matching. In Workshop on e-Commerce and NLP (ECNLP2019), Companion Proceedings of WWW. 381–386
- Ralph Peeters, Anna Primpeli, Benedikt Wichtlhuber, and Christian Bizer. 2020. Using schema.org Annotations for Training and Maintaining Product Matchers. In Proceedings of the 10th International Conference on Web Intelligence, Mining and Semantics.
- N. Barlaug and J. A. Gulla, “Neural Networks for Entity Matching: A Survey,” ACM Trans. Knowl. Discov. Data, vol. 15, no. 3, p. 52:1-52:37, Apr. 2021
- Sidharth Mudgal, Han Li, Theodoros Rekatsinas, AnHai Doan, Youngchoon Park, et al. 2018. Deep Learning for Entity Matching: A Design Space Exploration. In Proceedings of the 2018 International Conference on Management of Data. 19–34.
- Kashif Shah, Selcuk Kopru, and Jean David Ruvini. 2018. Neural Network Based Extreme Classification and Similarity Models for Product Matching. In Proceedings of the 2018 Conference of the Association for Computational Linguistics, Volume 3 (Industry Papers). 8–15.
- Muhammad Ebraheem, Saravanan Thirumuruganathan, Shafiq Joty, Mourad Ouzzani, and Nan Tang. 2018. Distributed Representations of Tuples for Entity Resolution. Proceedings of the VLDB Endowment 11, 11 (2018), 1454–1467.

Related Work: General (Deep) Entity Matching (2/2)

- Peter Christen. 2012. Data Matching: Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection. Springer-Verlag, Berlin Heidelberg.
- Vassilis Christophides, Vasilis Efthymiou, Themis Palpanas, George Papadakis, and Kostas Stefanidis. 2019. End-to-End Entity Resolution for Big Data: A Survey. arXiv:1905.06397 [cs] (2019).
- Hanna Köpcke, Andreas Thor, and Erhard Rahm. 2010. Evaluation of entity resolution approaches on real-world match problems. Proceedings of the VLDB Endowment 3, 1-2 (2010), 484–493.
- Petrovski, Petar, Volha Bryl, and Christian Bizer. "Integrating product data from websites offering microdata markup." *Proceedings of the 23rd International Conference on World Wide Web*. ACM, 2014.
- Ristoski, Petar, and Peter Mika. "Enriching product ads with metadata from HTML annotations." *International Semantic Web Conference*. Springer, Cham, 2016

Related Work: General (Deep) Schema Matching (1/2)

- J. Chen, E. Jimenez-Ruiz, I. Horrocks, and C. Sutton, “ColNet: Embedding the Semantics of Web Tables for Column Type Prediction,” arXiv:1811.01304 [cs]
- J. Chen, E. Jimenez-Ruiz, I. Horrocks, and C. Sutton, “Learning Semantic Annotations for Tabular Data,” in Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, Macao, China, Aug. 2019, pp. 2088–2094
- M. Hulsebos et al., “Sherlock: A Deep Learning Approach to Semantic Data Type Detection,” arXiv:1905.10688 [cs, stat], May 2019
- J. Zhang, B. Shin, J. D. Choi, and J. C. Ho, “SMAT: An Attention-Based Deep Learning Solution to the Automation of Schema Matching,” in Advances in Databases and Information Systems, Cham, 2021, pp. 260–274

Questions?