UNIVERSITY OF MANNHEIM
Data and Web Science Group

GPT4 versus BERT
Which Model Is More Suitable for Web Data Integration?

Prof. Dr. Christian Bizer

19th International Conference on
Web Information Systems and Technologies

WEBIST 2023

November 16, 2023

Hello

UNIVERSITY OF MANNHEIM
Data and Web Science Group

- **Prof. Dr. Christian Bizer**
- Chair of Information Systems:
  Web-based Systems

- Research Areas:
  - Large-scale data integration
  - Information extraction from semi-structured sources
  - Knowledge base construction
  - Analysis of the adoption of semantic web technologies

- Email: christian.bizer@uni-mannheim.de

Structured Data on the Web

Data Portals

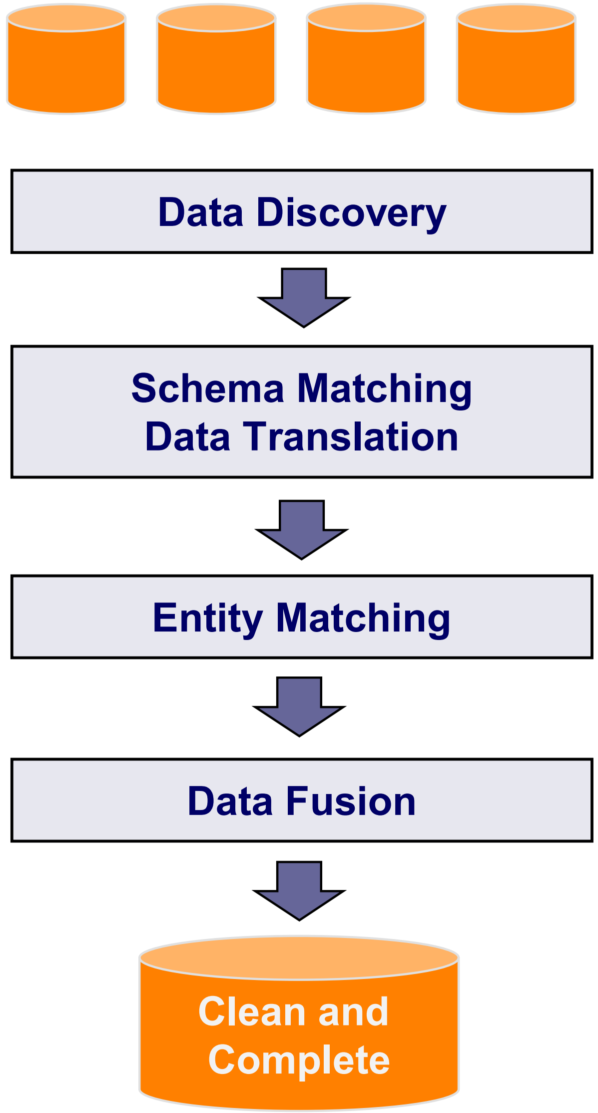
COVID-19 Portal

Web APIs

Web of Data

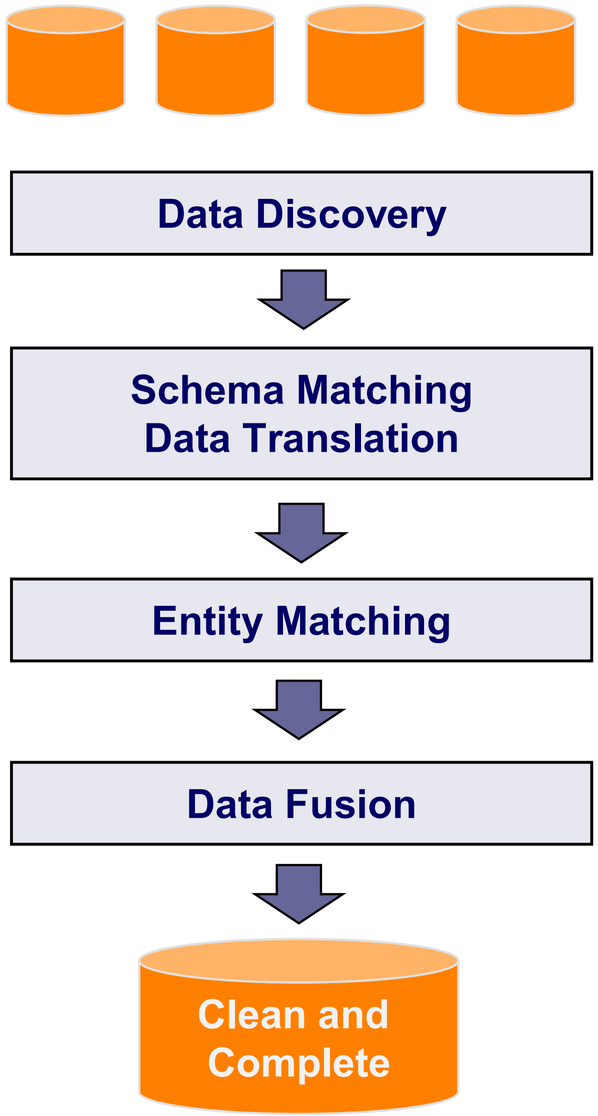
schema.org

The (Web) Data Integration Process



Outline

1. **Entity Matching**
   – BERT-based Methods
   – GPT-based Methods
2. **Table Annotation**
   – BERT-based Methods
   – GPT-based Methods
3. **Conclusions**



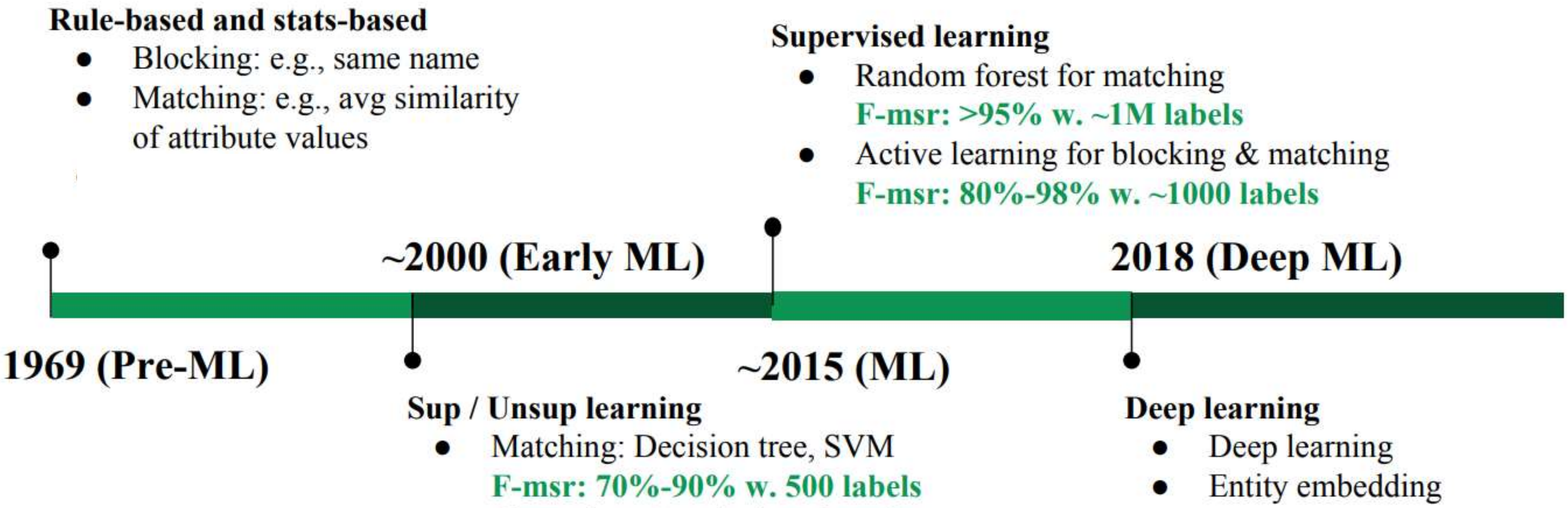
1. Entity Matching

Goal: Find all records that refer to the same real-world entity.

Brand	Product	Model No.	RAM	Color	Release
Samsung	Galaxy	S21	64	Blue	2021/1/29
Samung	Gal.	S 21 TGB12	64 GB	blau	Feb. 2021
NULL	Galaxy S20 Blue TGB12 64GB	NULL	64000	NULL	2020/1/29

Vassilis, et al.: **End-to-End Entity Resolution for Big Data.** ACM Surveys, 2020.
Barlaug and Gulla: **Neural Networks for Entity Matching: A Survey.** TKDD, 2021.

50 Years of Entity Matching



Luna Dong: **ML for Entity Linkage. Data Integration and Machine Learning: A Natural Synergy**. Tutorial at SIGMOD 2018.

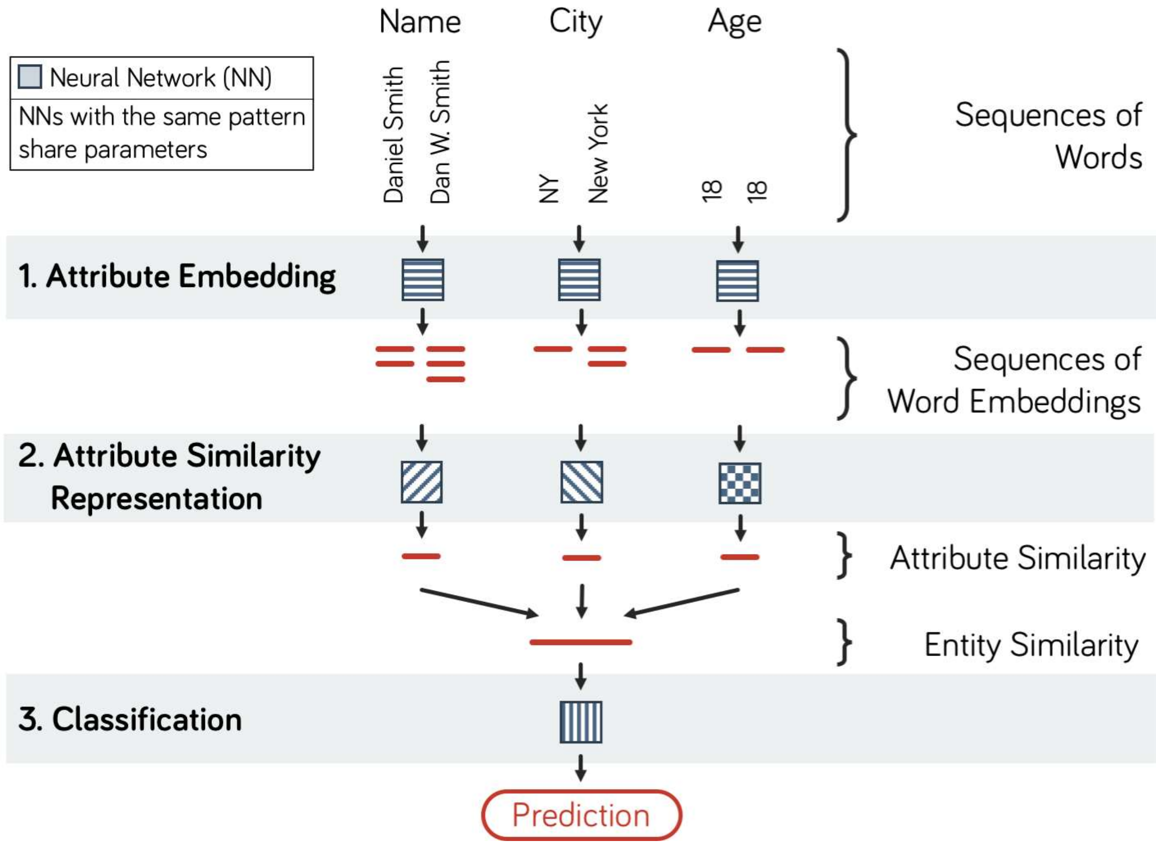
Entity Matching Benchmarks

Type	Dataset	Topic	# Pairs	# Matches	# Attrib.	# Sources
Structured	iTunes-Amazon	Music	539	132	8	2
	DBLP-ACM	Bibliographic	12,363	2,220	4	2
	DBLP-Scholar	Bibliographic	28,707	5,347	4	2
	Walmart-Amazon	Products	10,242	962	5	2
Textual	Abt-Buy	Products	9,575	1,028	3	2
	Amazon-Google	Products	11,460	1,167	3	2
	WDC Computers	Products	33,359	6,146	4	745
	WDC Products	Products	28,000	9,471	4	3259

schema.org

Anna Primpeli and Christian Bizer: **Profiling Entity Matching Benchmark Tasks**. CIKM 2020.
Papadakis, et al.: **A Critical Re-evaluation of Benchmark Datasets for Learning-Based Matching Algorithms.** Arxiv, 2023.

DeepMatcher (2018)



- Embeddings: FastText
- Summarization: Bi-RNN with attention
- Similarity computation: element-wise difference and multiplication, concatenation
- Classification: Fully connected neural net, cross entropy loss

Mudgal, Sidharth, et al.: **Deep Learning for Entity Matching: A Design Space Exploration.** SIGMOD, 2018.

Evaluation: DeepMatcher versus Magellan

UNIVERSITY OF MANNHEIM
Data and Web Science Group

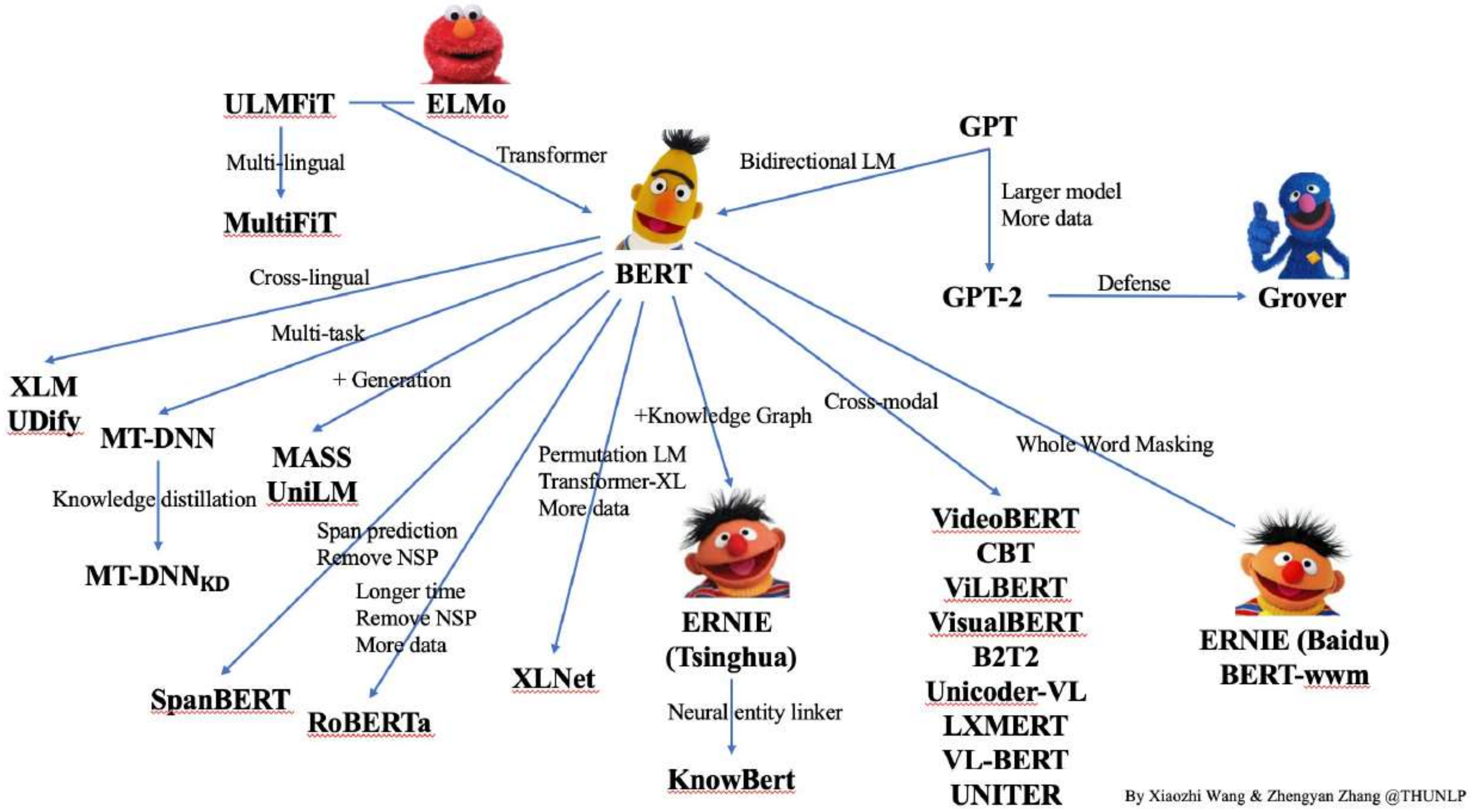
Type	Dataset	DeepMatcher F1	Magellan F1	Difference F1
Structured	iTunes-Amazon	88.5	91.2	-2.7
	DBLP-ACM	98.4	98.4	+0.0
	DBLP-Scholar	92.3	94.7	+2.4
	Walmart-Amazon	66.9	71.9	-5.0
Textual	Abt-Buy	62.8	43.6	+19.2
	Amazon-Google	69.3	49.1	+20.1
	WDC Computer - Large	89.5	64.5	+25.0
	WDC Computer - Small	70.5	57.6	+12.9

– DeepMatcher outperforms traditional methods on textual data
– mixed results on structured data

Konda, et al.: **Magellan: Toward Building Entity Matching Management**. PVLDB, 2016.

Transformers started to win all benchmarks in NLP

- Self-supervised pre-training on large text corpora
- Fine-tuning for downstream tasks



https://**huggingface.co**/docs/transformers/index

DITTO (2021)

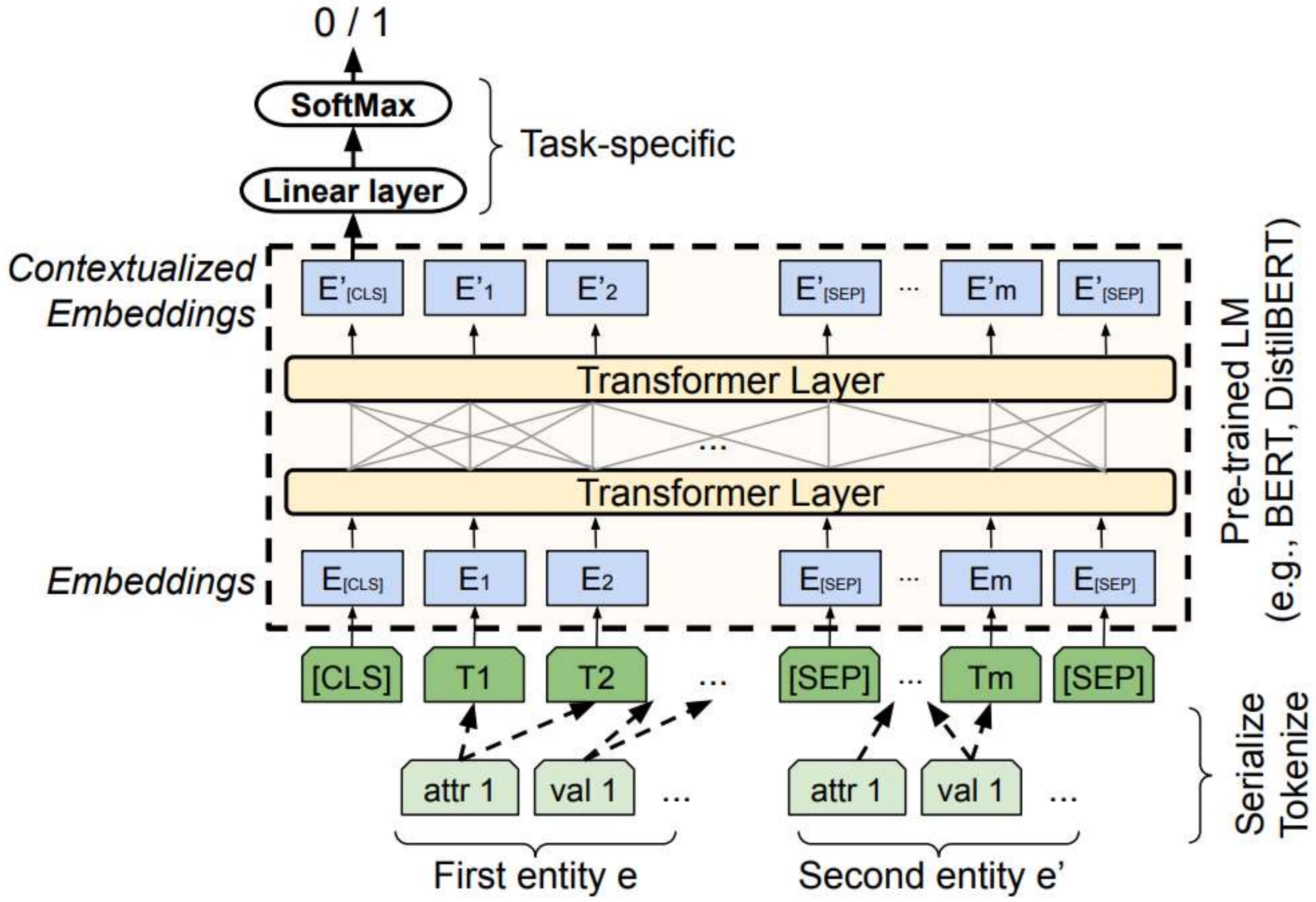
– applies BERT, DistilBERT, RoBERTa for entity matching

– Entity serialization for BERT
  – Pair of entity descriptions are turned into single sequence
  – [CLS] Entity Description 1 [SEP] Entity Description 2 [SEP]
  – Entity Description = [COL] $attr_1$ [VAL] val_1 . . . [COL] $attr_k$ [VAL] val_k

[CLS][COL] Title [VAL] DYMO D1 - Glossy tape [COL] Price [VAL] 1,99 €
[SEP][COL] Title [VAL] DYMO 45017 D1 Tape [COL] Price [VAL] 2,19 € **[SEP]**

Yuliang, et al: **Deep entity matching with pre-trained language models.** PVLDB, 2021.

DITTO: Architecture



– [CLS] token summarizes the pair of entities

– linear layer on top of [CLS] token for matching decision

DITTO: Evaluation

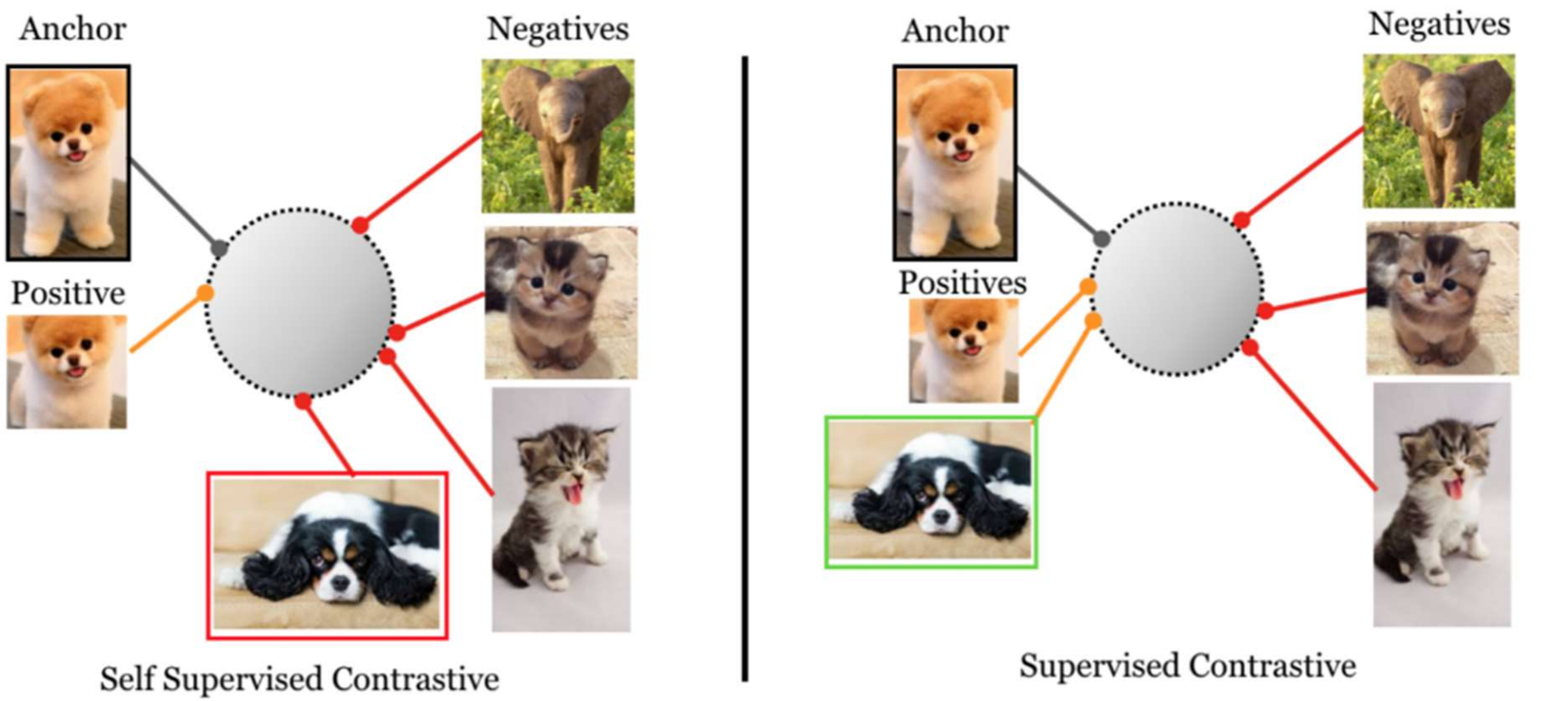
Type	Dataset	DITTO F1	DeepMatcher F1	Magellan F1
Structured	iTunes-Amazon	97.0	88.5 +8.5	91.2 +5.8
	DBLP-ACM	99.0	98.4 +0.6	98.4 +0.6
	DBLP-Scholar	95.6	92.3 +3.3	94.7 +0.9
	Walmart-Amazon	86.8	66.9 +19.9	71.9 +14.9
Textual	Abt-Buy	89.3	62.8 +26.5	43.6 +45.7
	Amazon-Google	75.6	69.3 +6.3	49.1 +26.5
	WDC Computer - Large	91.7	89.5 +3.2	64.5 +27.2
	WDC Computer - Small	80.8	70.5 +10.3	57.6 +23.2

- constant improvement for structured data
- large performance gain for textual data

Zeakis, et al.: **Pre-trained Embeddings for Entity Resolution: An Experimental Analysis**. PVLDB, 2023.

Contrastive Pretraining in Vision

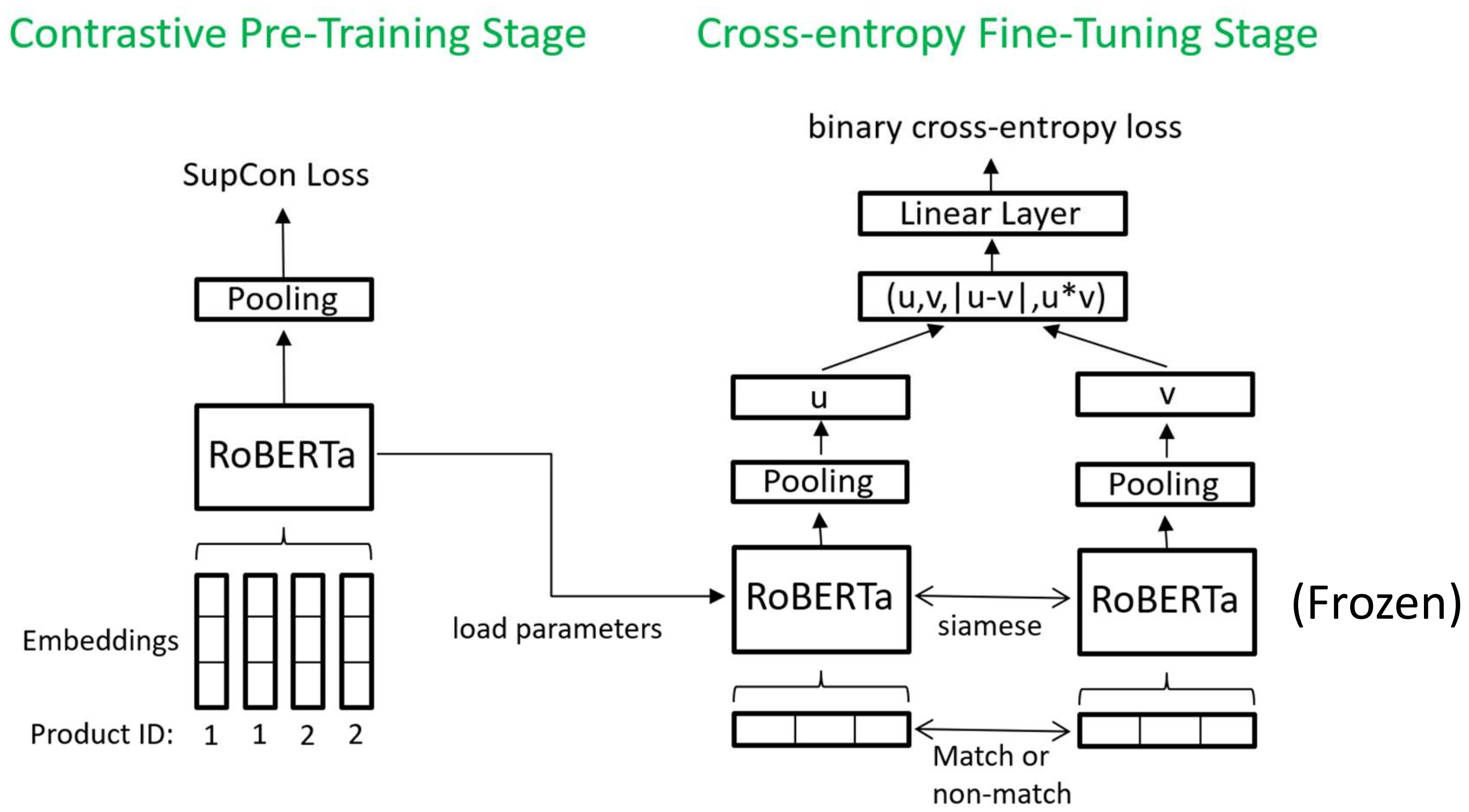
– maximizes distance between classes in the embedding space
– uses large batches containing many positive and negative examples



Khosla, et al.: **Supervised Contrastive Learning**. NeurIPS 2020.

Data and Web Science Group

Supervised Contrastive Pretraining for Entity Matching (2022)



Input: Batch of n product offers with product IDs

Input: Batch of product offer pairs with match/non-match labels

Peeters, Bizer: **Supervised Contrastive Learning for Product Matching**. WWW Companion 2022.

Evaluation: Supervised Contrastive Pretraining SupCon

	Abt-Buy	Amazon-Google	WDC Computers			
# Training Pairs	~7.5K	~9K	~3K (small)	~8K (medium)	~23K (large)	~68K (xlarge)
DeepMatcher	62.80	70.70	61.22	69.85	84.32	88.95
RoBERTa	91.05	74.10	86.37	91.90	94.68	94.73
Ditto	89.33	75.58	80.76	88.62	91.70	95.45
R-SupCon	93.70	**79.28**	93.18	97.66	98.16	**98.33**
R-SupCon+augmen	**94.29**	76.14	**95.21**	**98.50**	**98.50**	98.33
Δ to best baseline	+ 3.24	+ 3.70	+ 8.84	+ 6.60	+ 1.60	+ 0.84

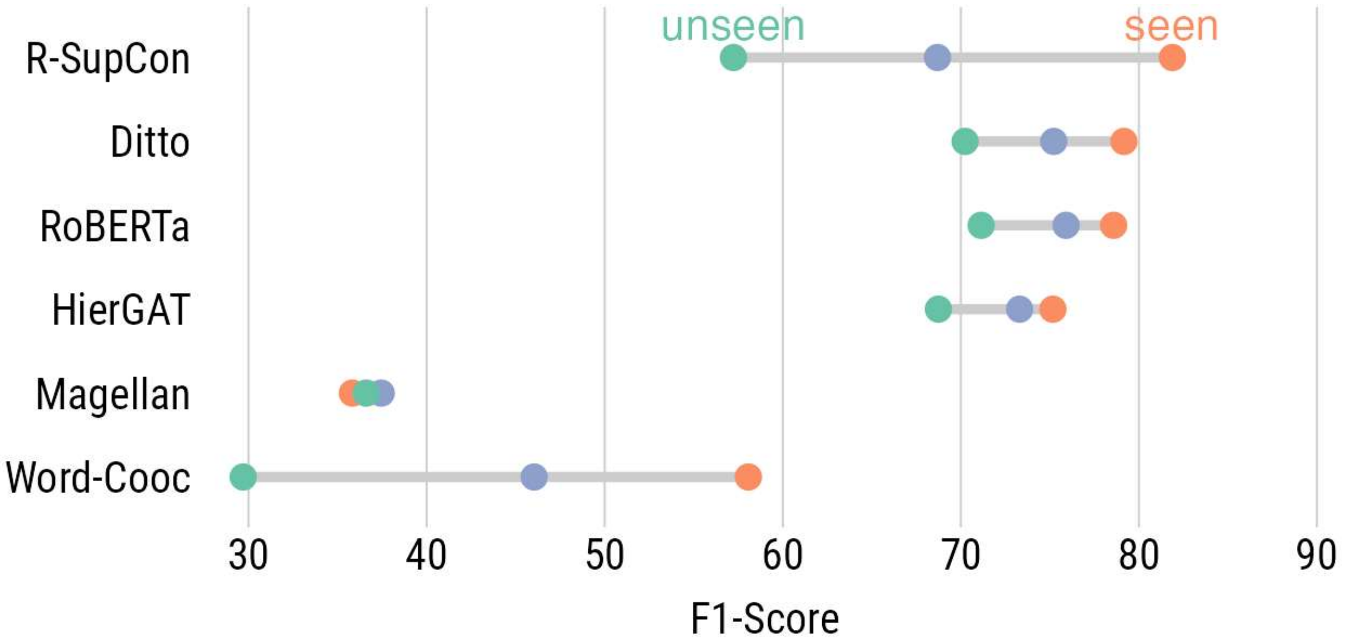
Large improvements for smaller training sets

Potential Reasons for the Good Performance of BERT-based Methods

- Serialization allows to pay attention to all attributes
  - no strict separation between attributes
- WordPiece tokenizer breaks unknown terms into pieces
  - no problems with out of vocabulary terms
- Transfer learning from pre-training texts
  - different surface forms may already be close in embedding space
- Contextualization of the embeddings
  - potentially more suited for capturing differing semantics

Drawbacks of BERT-based Methods: Overfitting to Seen Entities

- Benchmark: WDC Products, training set: 9500 pairs
- Test set$_{seen}$: offers for the same products as in training, 4500 pairs
- Test set$_{unseen}$: offers for different products, 4500 pairs



- 24.65% F1

- 8.92% F1

- 7.44% F1

- 6.43% F1

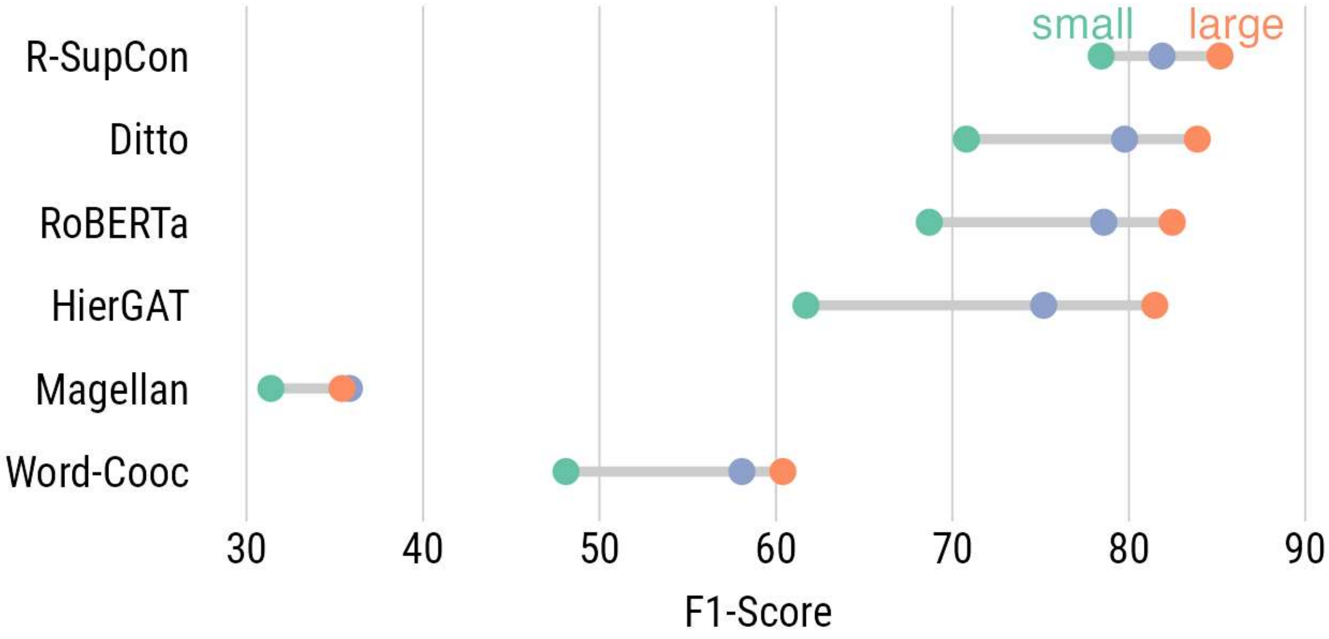
+ 0.78% F1

- 28.73% F1

Peeters, Bizer: **WDC Products: A Multi-dimensional Entity Matching Benchmark**. EDBT 2024.

Drawbacks of BERT-based Methods: Require Thousands of Training Examples

- WDC Products, Small training set: 5,000 pairs
- WDC Products, Large training set: 24,335 pairs

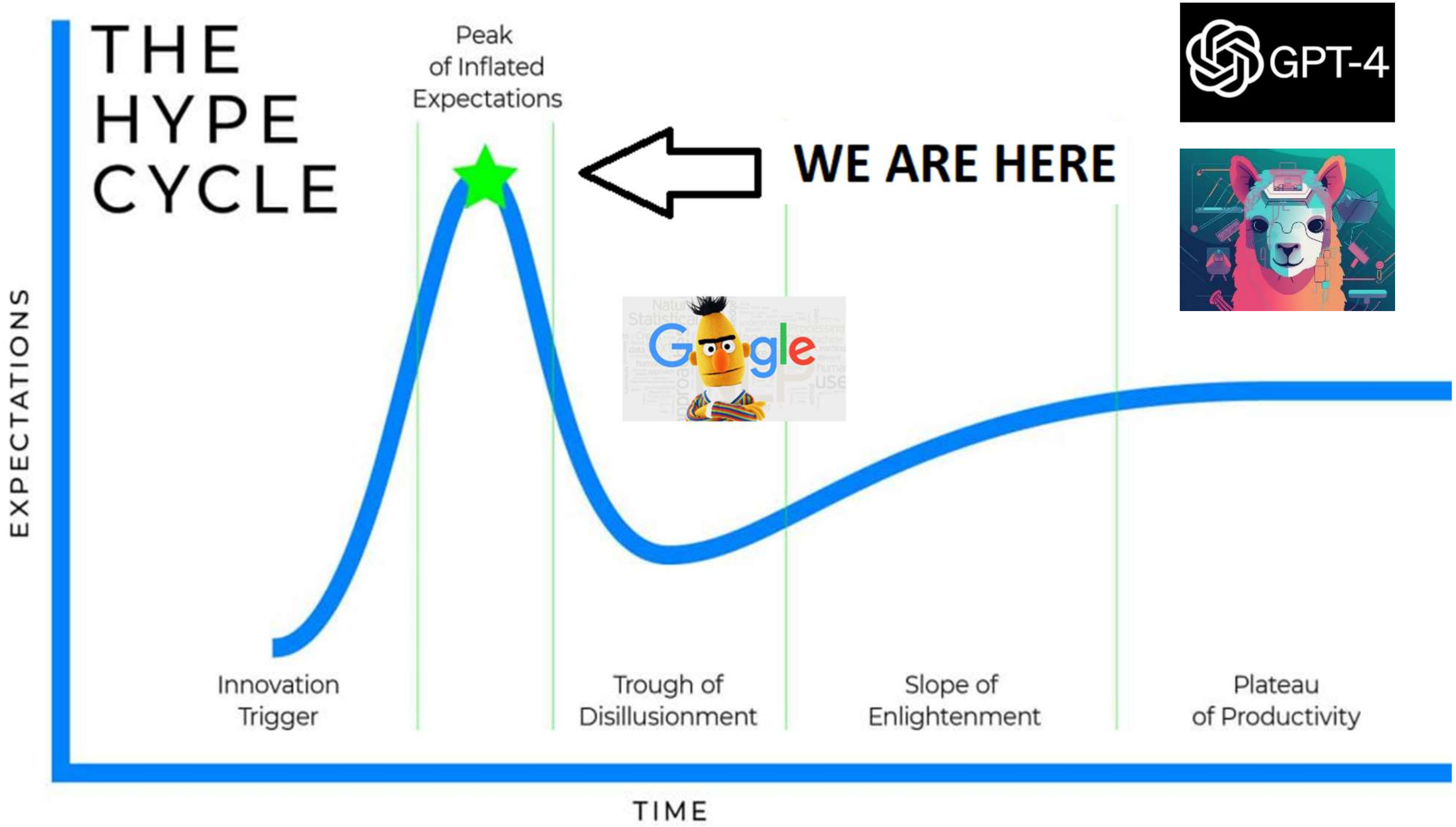


- 6.73% F1
- 13.69% F1
- 13.77% F1
- 19.77% F1
- 4.03% F1
- 12.29% F1

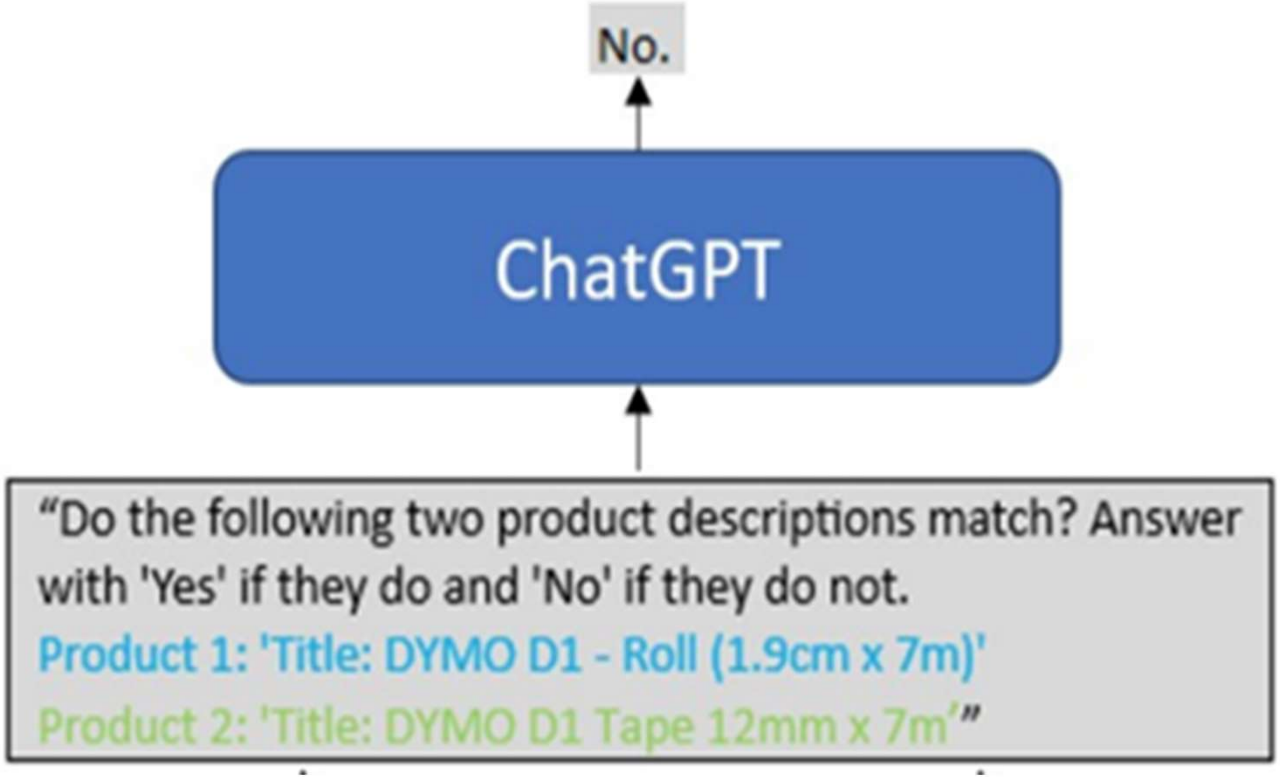
- significant effort for acquiring training labels
- continuous labeling and retraining necessary to cover new entities

UNIVERSITY OF MANNHEIM
Data and Web Science Group

Can Large Language Models (LLMs) address these drawbacks?



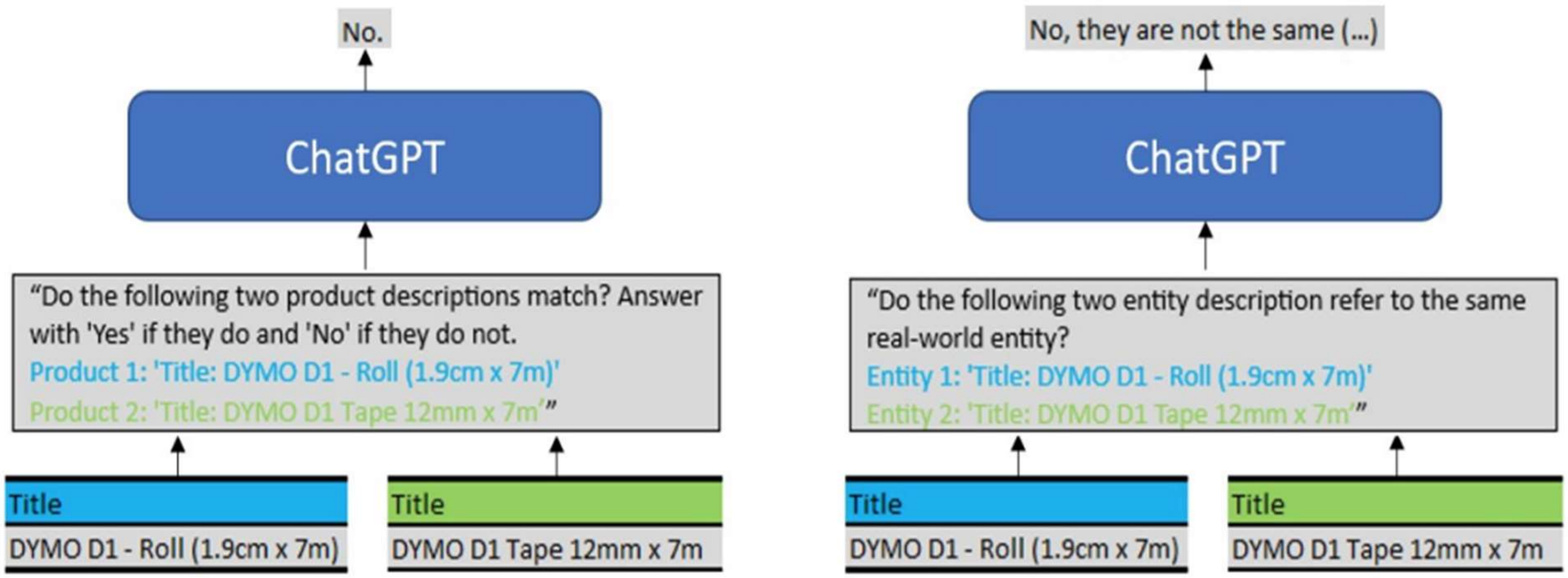
Entity Matching using Large Language Models



Peeters, Bizer: **Entity Matching using Large Language Models**. Arxiv, 2023.
Narayan, et al.: **Can Foundation Models Wrangle Your Data?** PVLDB, 2022.

UNIVERSITY OF MANNHEIM
Data and Web Science Group

Variations in the Prompt Formulation



Variations

- **general** vs. **domain-specific** wording
- **complex** vs. **simple** task description
- **free-form** vs. **forced** (restricted) answering

Peeters, Bizer: **Entity Matching using Large Language Models**. Arxiv, 2023.

Impact of Prompt Variations

- Models: gpt3.5-turbo-0301, gpt3.5-turbo-0613, gpt4-0613
- Benchmark: WDC Products, test set: 1250 pairs

Prompt/Model	Turbo03	Turbo06	GPT4
domain-complex-force	75.55	**74.96**	88.35
domain-complex-free	68.66	64.93	**89.61**
domain-simple-force	79.17	38.24	83.72
domain-simple-free	75.17	72.52	84.50
general-complex-force	76.51	60.62	85.83
general-complex-free	65.87	67.83	86.72
general-simple-force	78.33	14.02	77.39
general-simple-free	**79.70**	69.71	83.41
Mean	74.47	56.43	84.27
Standard deviation	4.28	17.87	3.42

Open-Source Models

UNIVERSITY OF MANNHEIM
Data and Web Science Group

- Models: SOLAR-0-70B-16Bit, StableBeluga2-70B (4 GPUs, 275GB VRAM)
- Benchmark: WDC Products, test set: 1250 pairs

Prompt/Model	Turbo03	GPT4	SOLAR	Beluga2	
domain-complex-force	75.55	88.35	67.93	**63.61**	- 34.64% F1
domain-complex-free	68.66	**89.61**	**72.95**	54.97	- 16.66% F1
domain-simple-force	79.17	83.72	26.71	44.19	
domain-simple-free	75.17	84.50	53.44	43.79	
general-complex-force	76.51	85.83	56.52	54.97	
general-complex-free	65.87	86.72	71.98	51.38	
general-simple-force	78.33	77.39	11.28	40.00	
general-simple-free	**79.70**	83.41	31.02	30.16	
Mean	74.47	84.27	51.43	47.58	- 32.84% F1 - 36.69% F1
Standard deviation	4.28	3.42	20.13	8.78	

Prompt as Hyperparmeter

Prompt	All Datasets (Mean F1)					WDC Products					Abt-Buy				
	Turbo03	Turbo06	GPT4	SOLAR	Beluga2	Turbo03	Turbo06	GPT4	SOLAR	Beluga2	Turbo03	Turbo06	GPT4	SOLAR	Beluga2
domain-complex-force	72.15	69.32	87.31	69.30	**68.30**	75.55	**74.96**	88.35	67.93	**63.61**	76.48	67.11	95.15	87.56	84.10
domain-complex-free	64.49	57.98	**88.01**	**75.98**	66.36	68.66	64.93	**89.61**	**72.95**	54.97	66.34	50.56	**95.78**	88.42	**85.79**
domain-simple-force	**77.47**	60.34	83.72	41.74	61.99	79.17	38.24	83.72	26.71	44.19	86.03	81.11	93.56	66.45	79.36
domain-simple-free	70.06	**73.27**	86.05	65.47	57.12	75.17	72.52	84.50	53.44	43.79	73.66	81.82	94.38	79.22	75.90
general-complex-force	70.43	66.79	86.41	55.59	63.33	76.51	60.62	85.83	56.52	54.97	74.32	82.30	94.40	85.04	83.51
general-complex-free	58.82	60.89	86.58	73.70	62.12	65.87	67.83	86.72	71.98	51.38	61.47	55.86	94.87	**89.20**	84.07
general-simple-force	74.40	39.41	77.78	21.77	54.57	78.33	14.02	77.39	11.28	40.00	**87.39**	69.07	93.23	52.30	77.47
general-simple-free	73.36	70.37	82.58	43.50	47.66	**79.70**	69.71	83.41	31.02	30.16	83.30	**83.83**	92.77	72.73	73.41
Narayan-simple	70.34	51.57	84.62	57.37	50.51	73.02	50.29	81.91	57.22	45.73	86.39	78.61	92.42	75.28	68.66
Narayan-complex	63.88	54.86	84.59	66.94	59.00	72.73	51.16	81.23	65.24	46.99	81.98	78.99	92.13	80.90	73.60
Mean	69.54	60.48	84.77	57.91	59.10	74.47	56.43	84.27	51.43	47.58	77.74	72.93	93.87	77.71	78.59
Standard deviation	5.31	9.70	2.82	16.18	6.34	4.28	17.87	3.42	20.13	8.78	8.43	11.23	1.17	11.01	5.44
Std. dev. Top2	1.54	1.45	0.35	1.14	0.97	0.27	1.22	0.63	0.48	4.32	0.50	0.77	0.31	0.39	0.85

Prompt	Walmart-Amazon					Amazon-Google					DBLP-Scholar				
	Turbo03	Turbo06	GPT4	SOLAR	Beluga2	Turbo03	Turbo06	GPT4	SOLAR	Beluga2	Turbo03	Turbo06	GPT4	SOLAR	Beluga2
domain-complex-force	67.88	60.26	89.00	74.92	69.64	60.37	61.25	75.61	43.75	**49.59**	80.46	**83.04**	88.44	72.32	74.58
domain-complex-free	52.80	42.59	89.33	81.61	**74.47**	55.61	49.53	75.57	52.03	45.56	79.06	82.28	89.78	**84.91**	71.03
domain-simple-force	74.38	57.52	88.78	44.36	63.32	63.64	44.38	75.32	14.67	47.54	**84.13**	80.45	77.21	56.50	**75.53**
domain-simple-free	60.24	**69.12**	88.67	75.00	61.79	**63.72**	**63.50**	74.51	41.64	35.26	77.53	79.39	88.20	78.04	68.88
general-complex-force	64.26	68.77	**89.67**	65.32	63.46	57.91	44.06	74.91	19.01	42.01	79.15	78.22	87.22	52.05	72.68
general-complex-free	44.42	45.82	89.45	**83.11**	66.46	45.88	51.91	74.38	44.31	40.50	76.47	83.03	87.50	79.91	68.18
general-simple-force	**74.81**	31.03	86.41	26.79	59.59	49.56	15.73	53.60	5.79	31.02	81.89	67.18	78.26	12.69	64.78
general-simple-free	68.63	65.19	88.60	58.25	51.47	57.63	52.31	66.67	14.12	28.77	77.53	80.81	81.47	41.38	54.49
Narayan-simple	68.42	42.45	84.72	62.30	49.81	56.35	29.61	75.70	39.62	34.34	67.52	56.90	88.37	74.40	54.02
Narayan-complex	52.67	40.16	83.37	71.93	62.82	43.35	33.33	**76.38**	**54.69**	45.97	68.67	70.67	**89.82**	78.82	65.62
Mean	62.85	52.29	87.80	64.36	62.28	55.40	44.56	72.27	32.96	40.06	77.24	76.20	85.63	63.10	66.98
Standard deviation	9.59	12.82	2.08	16.69	7.09	6.66	13.99	6.76	16.80	6.93	5.05	8.18	4.53	21.49	7.18
Std. dev. Top2	0.22	0.18	0.11	0.75	2.42	0.04	1.13	0.41	1.33	1.03	1.12	0.01	0.02	2.50	0.48

No single prompt works best for all model/dataset combinations

GPT-based versus BERT-based Entity Matching Methods

Model	WDC	Abt-Buy	Wal-Ama	Ama-Goog	DBLP-Sch
Turbo03	79.70	87.39	74.81	63.72	84.13
GPT4	**89.61**	**95.78**	**89.67**	76.38	89.82
RoBERTa	77.53	91.21	87.02	<u>79.27</u>	<u>93.88</u>
Ditto	<u>84.90</u>	<u>91.31</u>	<u>86.39</u>	**80.07**	**94.31**
Δ Best GPT/BERT	4.71	4.47	2.65	-3.69	-4.49

- GPT results are zero-shot: No task-specific training data!
- RoBERTa and DITTO are fine-tuned using 5K to 22K training pairs

Peeters, Bizer: **Entity Matching using Large Language Models**. Arxiv, 2023.

BERT-based Methods
Generalization to other Datasets

– Roberta and DITTO trained on WDC and applied to other datasets

Model	WDC	Abt-Buy	Wal-Ama	Ama-Goog	DBLP-Sch
$\text{RoBERTa}_{\text{Seen}}$	77.53	91.21	87.02	79.27	93.88
$\text{DITTO}_{\text{Seen}}$	**84.90**	**91.31**	**86.39**	**80.07**	**94.31**
$\textbf{RoBERTa}_{\textbf{Unseen}}$	-	55.52	36.46	31.00	29.64
Δ $\text{RoBERTa}_{\text{Unseen}}$	-	-35.69	-50.56	-48.27	-64.24
$\textbf{DITTO}_{\textbf{Unseen}}$	-	48.74	31.55	33.12	32.82
Δ $\text{DITTO}_{\text{Unseen}}$	-	-42.57	-54.84	-46.95	-61.49

– BERT-based methods: Hardly any transfer between datasets

– GPT-based methods: Transfer from pre-training plus emergent effects

In-Context Learning via Demonstrations

USER: Do the following two product descriptions match?
Product 1: ‘DYMO D1 19 mm x 7 m’
Product 2: ‘Dymo D1 (19mm x 7m – BoW)’
ASSISTANT: Yes.

USER: Do the following two product descriptions match?
Product 1: ‘DYMO D1 Tape 24mm’
Product 2: ‘Dymo D1 19mm x 7m’
ASSISTANT: No.

USER: Do the following two product descriptions match?
Answer with 'Yes' if they do and 'No' if they do not.

Product 1: 'Title: DYMO D1 - Glossy tape - black on white - Roll (1.9cm x 7m) - 1 roll(s)’
Product 2: 'Title: DYMO 45017 D1 Tape 12mm x 7m sort p rd, S0720570'

ASSISTANT: No.

In-Context Learning via Matching Rules

SYSTEM: Your task is to decide if two product descriptions match.
The following rules need to be observed:
1. The brand of matching products must be the same if available
2. Model numbers of matching products must be the same if available
3. Additional features of matching products must be the same if available
4. Matching attributes may not have the exact same surface form.
5. If an attribute is missing for one description, it is likely still a match if the existing attributes match.

USER: Do the following two product descriptions match?
Answer with 'Yes' if they do and 'No' if they do not.

Product 1: 'Title: DYMO D1 - Glossy tape - black on white - Roll (1.9cm x 7m) - 1 roll(s)’
Product 2: 'Title: DYMO 45017 D1 Tape 12mm x 7m sort p rd, S0720570'

ASSISTANT: No.

Results In-Context Learning

Mean F1 over all 5 benchmark datasets

Prompt/Model	Shots	Turbo03	Turbo06	GPT4	SOLAR	Beluga2
Fewshot-related	6	71.87	70.28	85.22	63.10	68.27
	10	71.89	69.93	86.64	64.64	69.23
Fewshot-random	6	79.25	75.75	85.11	79.73	**77.33**
	10	**80.62**	**77.32**	85.77	**80.71**	77.13
Hand-written rules	0	78.59	70.95	85.77	77.24	75.49
Learned rules	0	57.17	67.33	85.04	75.69	75.29
Best zero-shot	0	77.95	74.95	**88.25**	76.97	69.80
Δ Best zero-shot	-	2.67	2.37	-1.61	3.74	7.53

– GPT3.5 and open-source models benefit from in-context learning
– For GPT4 the additional guidance is harmful!

Impact of In-Context-Learning for GPT4 and the Amazon-Google Dataset

Model	Amazon-Google
Ditto	80.07
$GPT4_{Zeroshot}$	76.38
Δ $GPT4_{Zeroshot}$/Ditto	**-3.69**
$GPT4_{Random10}$	78.76
$GPT4_{Related10}$	85.21
Δ $GPT4_{Related10}$/Ditto	**+5.21**

– For some datasets, in-context learning is needed to notch GPT4 into the right direction!

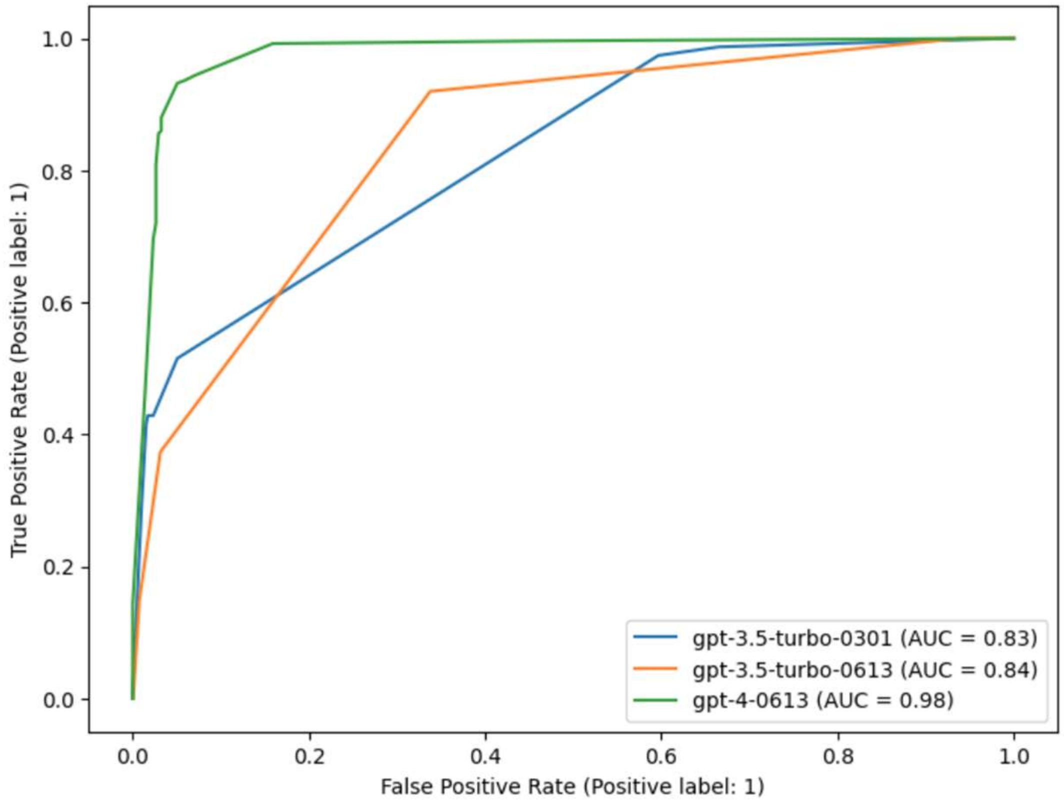
Peeters, Bizer: **Entity Matching using Large Language Models**. Arxiv, 2023.

Confidence Scores

USER: Do the following two product descriptions refer to the same product?
Provide a confidence score for your decision, 100% referring to full confidence. …

ASSISTANT: No. Confidence: 95%

The ROC curve shows the quality of the confidence scores



Explanations

USER: Do the following two product descriptions refer to the same product?
Please explain your decision using the following format …

Product 1: ZEBRA 800033-801 BLACK RIBBON (1000 PRINTS) GBP 9.5
Product 2: Zebra Black Monochrome Ribbon 800011-101 - 1000 Prints GBP £12.99

ASSISTANT: Yes.
attribute=brand; importance=0.20; values=ZEBRA, Zebra; similarity=1.00
attribute=model; importance=0.40; values=800033-801, 800011-101; similarity=0.00
attribute=color; importance=0.10; values=Black, Black; similarity=1.00
attribute=prints; importance=0.20; values=1000, 1000; similarity=1.00
attribute=price; importance=0.10; values=GBP 9.5, GBP £12.99; similarity=0.00

False positive!

Explanations for BERT-based entity matchers see:
Paganelli, Matteo et al.: **Analyzing How BERT Performs Entity Matching**. PVLDB, 2022.

Error Analysis (and Prompt Breeding?)

USER: The following list contains false positive and false negative product pairs from the output of a product matching system. Given the product pairs and associated attribute and importance information, come up with a set of error classes that explain why the systems fails on these examples. …. 10 false positives and 10 negatives …

ASSISTANT:
False Positives Error Classes:
1. Model Number Confusion
2. Semantic Misunderstanding
3. Minor Differences Overlooked
4. Attribute Importance Misalignment

False Negatives Error Classes:
1. Description Misinterpretation
2. Price and Currency Differences
3. Missing Information
4. Different Measurement Units

Fernando, et al.: **Promptbreeder: Self-Referential Self-Improvement Via Prompt Evolution.** *ArXiv,* 2023.

Summary:
GPT-based versus BERT-based Matching Methods

1. GPT-based matchers require less task-specific training data
   – $\text{GPT4}_{\text{zeroshot}}$ outperforms fine-tuned BERT models in many cases
2. GPT-based matchers are more robust to unseen entities
   – important for Web use cases that often involve unseen entities
3. Both approaches reduce the feature engineering effort
   – no information extraction necessary
   – less value normalization necessary due to pre-training
4. GPT-based matchers can explain matching decisions

2. Table Annotation

UNIVERSITY OF MANNHEIM
Data and Web Science Group

Goal: Annotate table columns with terms from a shared vocabulary.

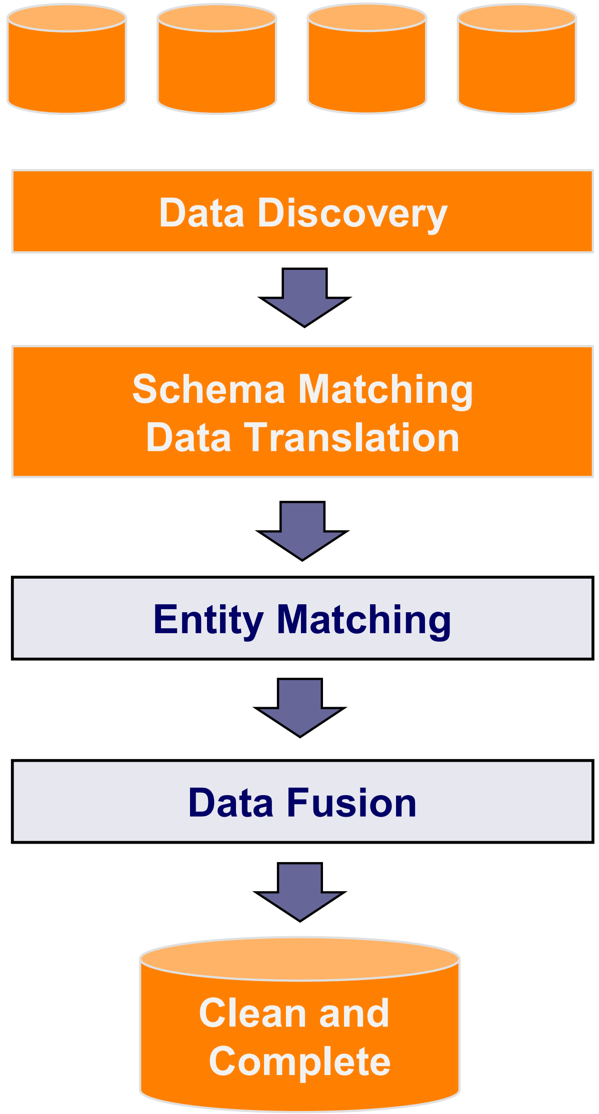
Use Cases:

1. data lake indexing for search
2. schema matching via global schema

schema.org DBpedia

annotate

film	director	producer	country
???	???	???	???
Happy Feet	George Miller, Warren Coleman, Judy Morris	Bill Miller, George Miller, Doug Mitchell	USA
Cars	John Lasseter, Joe Ranft	Darla K. Anderson	UK
Flushed Away	David Bowers, Sam Fell	Dick Clement, Ian La Frenais, Simon Nye	France



Column Type Annotation (CTA)

film director producer country

???	???	???	???
Happy Feet	George Miller, Warren Coleman, Judy Morris	Bill Miller, George Miller, Doug Mitchell	USA
Cars	John Lasseter, Joe Ranft	Darla K. Anderson	UK
Flushed Away	David Bowers, Sam Fell	Dick Clement, Ian La Frenais, Simon Nye	France

Column Property Annotation (CPA)

person	location	sports_team
Max Browne	Sammamish, Washington	Southern California
Thomas Tyner	Aloha, Oregon	Oregon
Derrick Henry	Yulee, Florida	Alabama

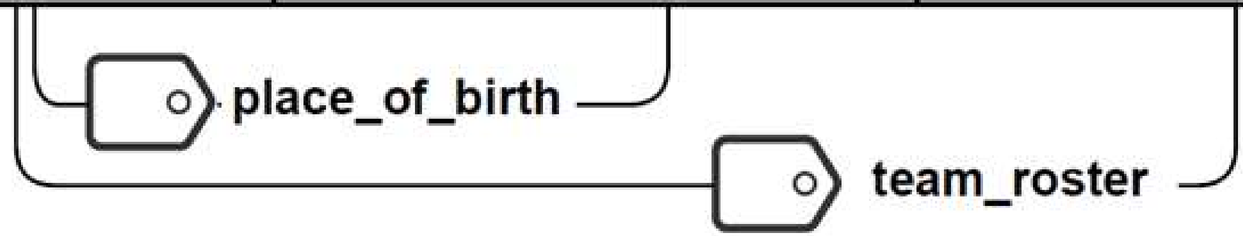


Table Annotation Benchmarks

Task	Dataset	# Tables	Vocabulary	# Terms	# Sources
CTA	WikiTables	410,000	Freebase	255	1
	GitTables SemTab	6,892	DBpedia	122	1
	WDC SOTAB V2	45,378	Schema.org	82	44,268
	WDC SOTAB Small	103	Schema.org	32	103
CPA	WikiTables	53,000	Freebase	121	1
	WDC SOTAB V2	29,723	Schema.org	110	29,540

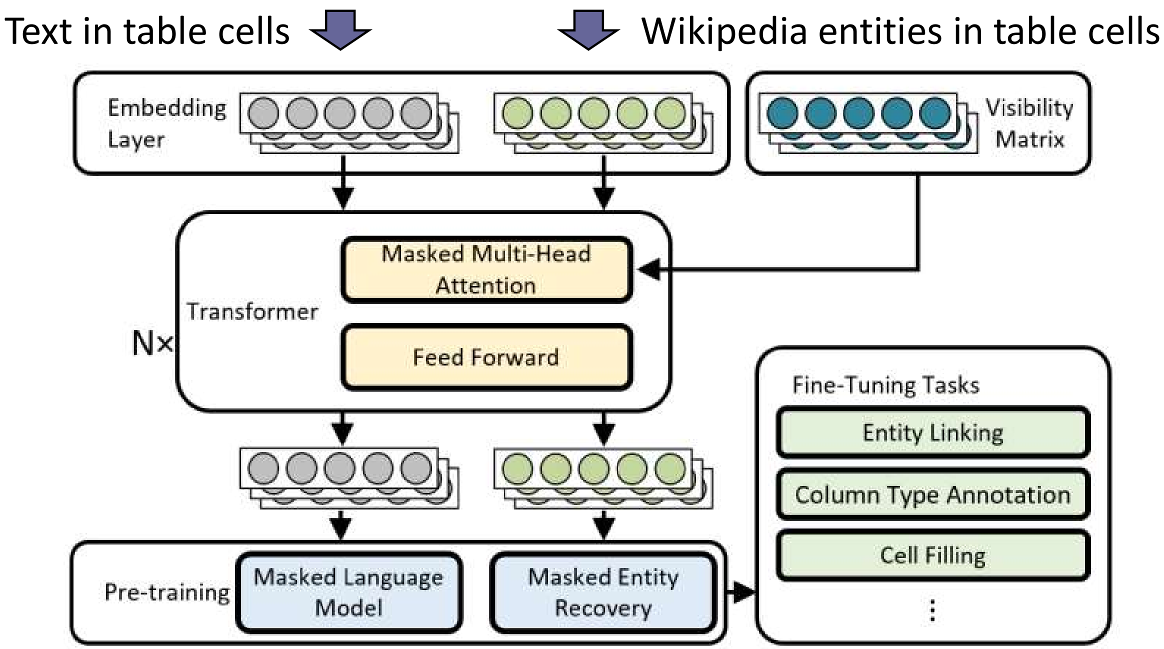
schema.org

SemTab Table Annotation Evaluation Campaign: https://www.cs.ox.ac.uk/isg/challenges/sem-tab/

Deng, et al.: **TURL: Table Understanding through Representation Learning**. PVLDB 2020.
Korini, et al.: **SOTAB: The WDC Schema. org table annotation benchmark.** SemTab Proceedings, 2022.

TURL (2020)

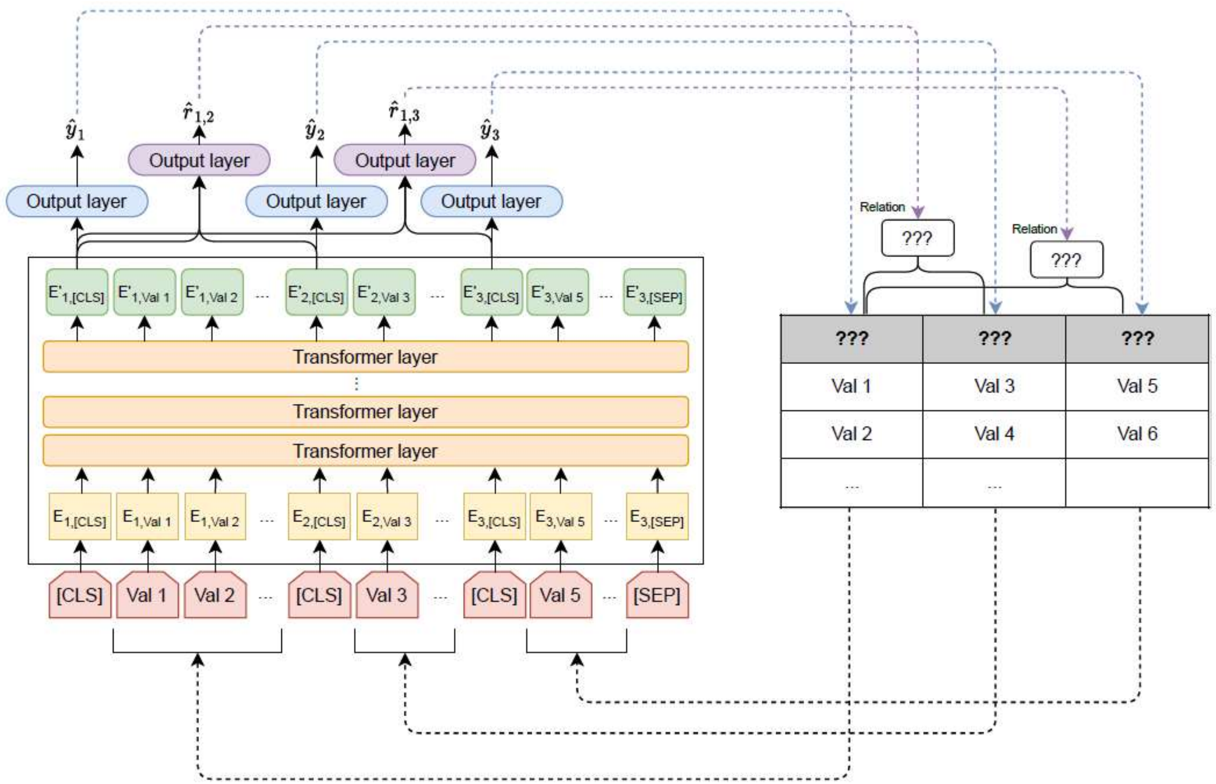
- aims at learning generic table representations that are useful across a wide range of tasks
  - Pre-training: Self-supervised table representation learning
  - Fine-tuning: For 6 specific downstream tasks

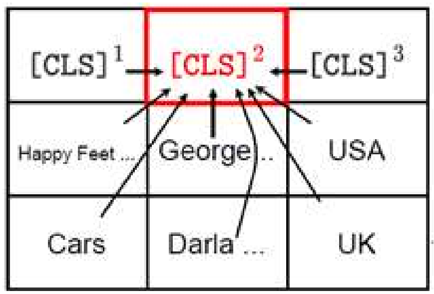


Deng, et al.: **TURL: Table Understanding through Representation Learning**. PVLDB 2020.
Pujara, et al.: **From Tables to Knowledge: Recent Advances in Table Understanding**. Tutorial at KDD2021.

DoDuo (2022)

- directly fine-tunes BERT for column and relation annotation tasks
- a table cell can pay attention to all neighboring cells
- exploits synergies between CTA and CPA task using multi-task learning





Suhara, et al.: **Annotating Columns with Pre-trained Language Models**. SIGMOD 2022.

Evaluation Results: Table Annotation

– Column Type Annotation (CTA): WikiTables

Method	F1	P	R
TURL (TinyBERT)	88.86	90.54	87.23
DoDuo (BERT)	92.45	92.45	92.21

– Column Property Annotation (CTA): WikiTables

Method	F1	P	R
TURL (TinyBERT)	90.94	91.18	90.69
DoDuo (BERT)	91.72	91.97	91.47

– good results around 90% F1 for both tasks
– use lots of training data for pre-training and fine-tuning
– all labels are covered in the training data, no unseen ones

Can Large Language Models (LLMs) do better for Column Type Annotation?

Korini, Bizer: **Column Type Annotation using ChatGPT**. VLDB Workshops, 2023.
Feuer: **ArcheType: A Novel Framework for Column Type Annotation using Large Language Models.** Arxiv, 2023.

CTA as Column Classification

- Benchmark: $SOTAB_{Small}$
- Topics: Restaurants, Events, Albums, Artists

RestaurantName	Postal Code	Payment Accepted	Time
Friends Pizza	2525	Cash Visa MasterCard	7:30 AM
Gourmandize	4551	Cash Visa MasterCard	7:00 AM
Marco's Organic Pizza	6060	Cash Visa	10:00 AM

SYSTEM: Classify the column given to you into one of these types that are separated by comma: RestaurantName, ArtistName, AlbumName, EventName, PriceRange, AddressRegion, Country, Telephone, PaymentAccepted, PostalCode, Coordinate, DayOfWeek, Time, RestaurantDescription, Review, Date, DateTime, Organization, EventDescription, EventStatusType, EventAttendanceModeEnumeration, Currency, Telephone, MusicRecordingName, Duration

USER: Column: 7:30 AM 7:00 AM 10:00 AM 5:00 PM 11:00 AM

Type:

ASSISTANT: Time

Using the Whole Table as Context for Disambiguation

RestaurantName	Postal Code	Payment Accepted	Time
Friends Pizza	2525	Cash Visa MasterCard	7:30 AM
Gourmandize	4551	Cash Visa MasterCard	7:00 AM
Marco's Organic Pizza	6060	Cash Visa	10:00 AM

SYSTEM: Classify the columns of a given table with only one of the following classes that are separated with comma: RestaurantName, ArtistName, AlbumName, PostalCode, AddressRegion, … {32 semantic types are listed here}

USER: Table: Column 1 || Column 2 || Column 3 || Column 4 \n
Friends Pizza || 2525|| Cash Visa MasterCard || 7:30 AM \n …
Classes:

ASSISTANT: RestaurantName, PostalCode, PaymentAccepted, Time

Providing Explicit Instructions

- we instruct the model by providing reasoning steps
- we explicitly specify that the input is a table

SYSTEM: Classify the columns of a given table with only one of the following classes that are separated with comma: {32 semantic types are listed here}

Instructions: 1. Look at the input given to you and make a table out of it.
2. Look at the cell values in detail.
3. For each column, select a class that best represents the meaning of all cells.
4. Answer with the selected class for each columns with the format Column1: class.

USER: Table: Column 1 || Column 2 || Column 3 || Column 4 \n Friends Pizza || 2525|| Cash Visa MasterCard || 7:30 AM \n …

Classes:

ASSISTANT: Column1: RestaurantName\n Column2: PostalCode\n Column3: PaymentAccepted\n Column4: Time

Column Type Annotation Results

- Benchmark: SOTAB Small, 32 terms, zero-shot

Prompt / Model	GPT03	GPT4	Stable Beluga2	Falcon40B
Column	45.85	86.31	**75.55**	**21.42**
Table	37.90	<u>94.19</u>	20.63	-
Column+Instructions	<u>78.61</u>	92.36	<u>74.84</u>	<u>11.67</u>
Table+Instructions	**85.25**	**95.14**	53.82	2.7

- Table approach with instructions works best for OpenAI models.
- open-source models are confused by complete tables
- this are zero-shot results without using task-specific training data
  ➔ The models "know" the terms from pre-training.

Korini, Bizer: **Column Type Annotation using ChatGPT**. VLDB Workshops, 2023.

GPT-based versus BERT-based Table Annotation Methods

- Training example (shot): Annotated column
- RoBERTa fine-tuned using concatenated cell values
- DODUO fine-tuned by embedding complete tables with column labels

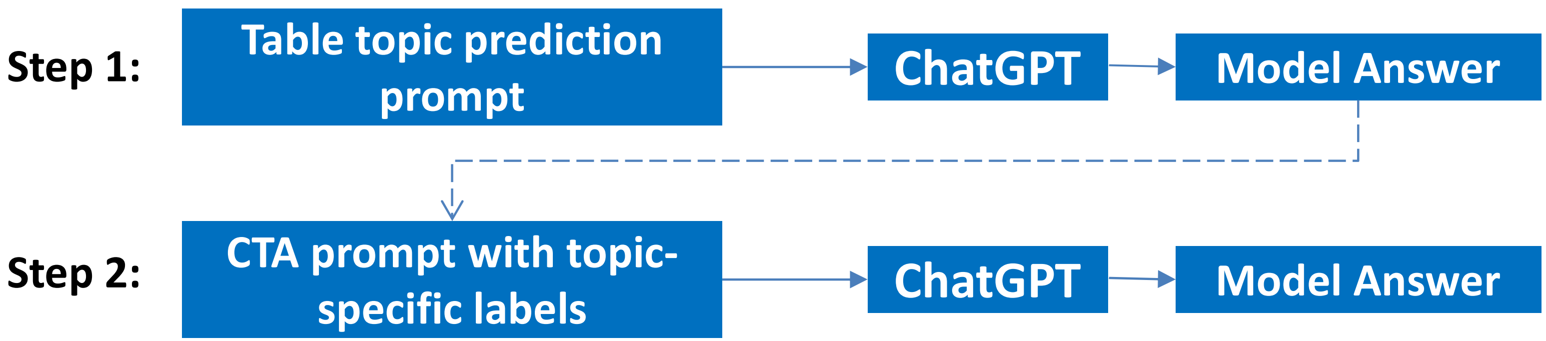
Model	Shots	F1	Δ F1
GPT4	0	**95.14**	-
RoBERTa	356	89.73	- 5,41
RoBERTa	1600	86.79	- 8,35
DoDuo	356	6.37	- 88,77
DoDuo	1600	53.6	- 41,54

- RoBERTa using 1600 examples performs worse than GPT4 zero-shot
- DoDuo confused due to low number of training tables

Challenge: Large Vocabularies

- **Idea:** Split CTA into two steps
  1. predict topic of complete table
  2. perform CTA using reduced set of topic-specific labels

- **Advantages:**
  1. save token space for large vocabularies
  2. simplify the annotation task as the model chooses from smaller set of labels



Step 1: Table Topic Prediction

SYSTEM: Your task is to classify if a table describes restaurants, events, music recordings or hotels.

SYSTEM: Your instructions are: 1. Look at the input given to you and make a table out of it. 2. Look at the cell values in detail. 3. Decide if the table describes a Restaurant, Event, Music Recording or Hotel. 4. Answer with Restaurant, Event, Music Recording or Hotel.

USER: Classify this table: Column 1 || Column 2 || Column 3 || Column 4 \n Friends Pizza || 2525|| Cash Visa MasterCard || 7:30 AM \n …

ASISSTANT: Restaurant

Step 2: Column Type Annotation

- the first system message uses only relevant subset of all labels
- e.g. only 11 out of 32 labels belonging to the "Restaurant" topic

SYSTEM: Your task is to classify the columns of a given table with only one of the following classes that are separated with comma: {relevant subset of all labels}

Your instructions are: 1. Look at the input given to you and make a table out of it. 2. Look at the cell values in detail. 3. For each column, select a class …

USER: Classify these table columns: Column 1 || Column 2 || Column 3 || Column 4 \n Friends Pizza || 2525|| Cash Visa MasterCard || 7:30 AM \n …

ASSISTANT: Column1: RestaurantName\n Column2: PostalCode\n Column3: PaymentAccepted\n Column4: Time

Results: Two-Step Approach

UNIVERSITY OF MANNHEIM
Data and Web Science Group

F1	GPT03	GPT4	Stable Beluga2	Falcon40B
Column	45.85	86.31	**75.55**	**21.42**
Table	37.90	94.19	20.63	-
Column+instructions	78.61	92.36	74.84	11.67
Table+instructions	85.25	**95.14**	53.82	2.7
Two-step Pipeline	**89.47**	94.95	31.57	-

– Two-step approach helps GPT03 to handle label space
– GPT4 does not require additional guidance for $\text{SOTAB}_{\text{small}}$

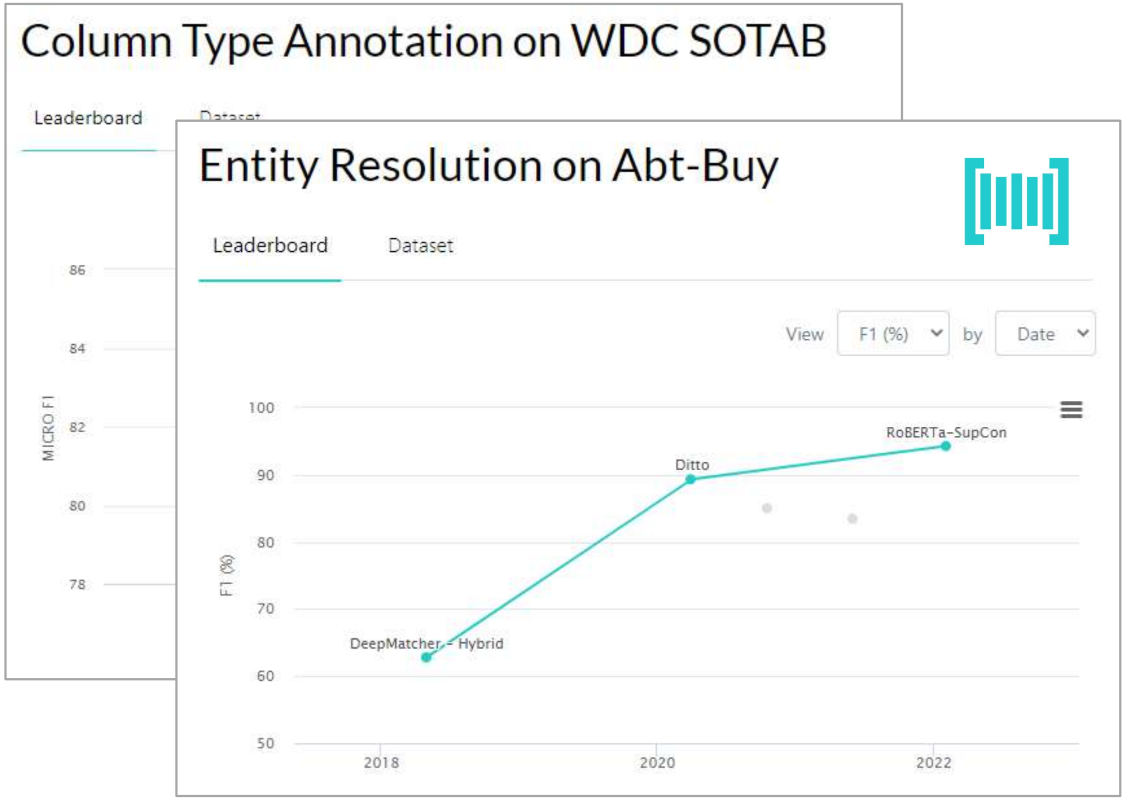
3. Conclusions

UNIVERSITY OF MANNHEIM
Data and Web Science Group

1. GPT-based methods require less task-specific training data
   – high zero-shot performance of GPT4
2. GPT-based methods are more robust to unseen entities
   – important for Web use cases that often involve unseen entities
3. BERT-based methods are cheaper to run
   – no API usage fees, less GPU required
4. GPTs ability to generate explanations might increase the trust of the user into the integration results

Staying Up To Date

- Papers with Code collects results for all discussed benchmarks



https://paperswithcode.com/**task/entity-resolution/**
https://paperswithcode.com/**task/table-annotation/**

Thank you.

Email: christian.bizer@uni-mannheim.de

Web: https://www.uni-mannheim.de/dws/people/professors/prof-dr-christian-bizer/