PLMs for Data Preparation

——Fine-tune language models for single tasks

Outline

- **Overview**
  - Motivation of PLMs
  - Basic Concepts of PLMs
  - Non-contexual Embeddings for Data Preparation
  - Contextual Embeddings for Data Preparation
  - Domain Adaptation
  - Unified Data Matching
- **Challenges and Open Problems**

Outline

- ☐ Overview
  - Motivation
  - Basic Concepts of PLMs
  - Non-contexual Embeddings for Data Preparation
  - Contextual Embeddings for Data Preparation
  - Domain Adaptation
  - Unified Data Matching
- ☐ Challenges and Open Problems

Motivation

- The Dilemma of complex ML tasks (e.g.,NLP)

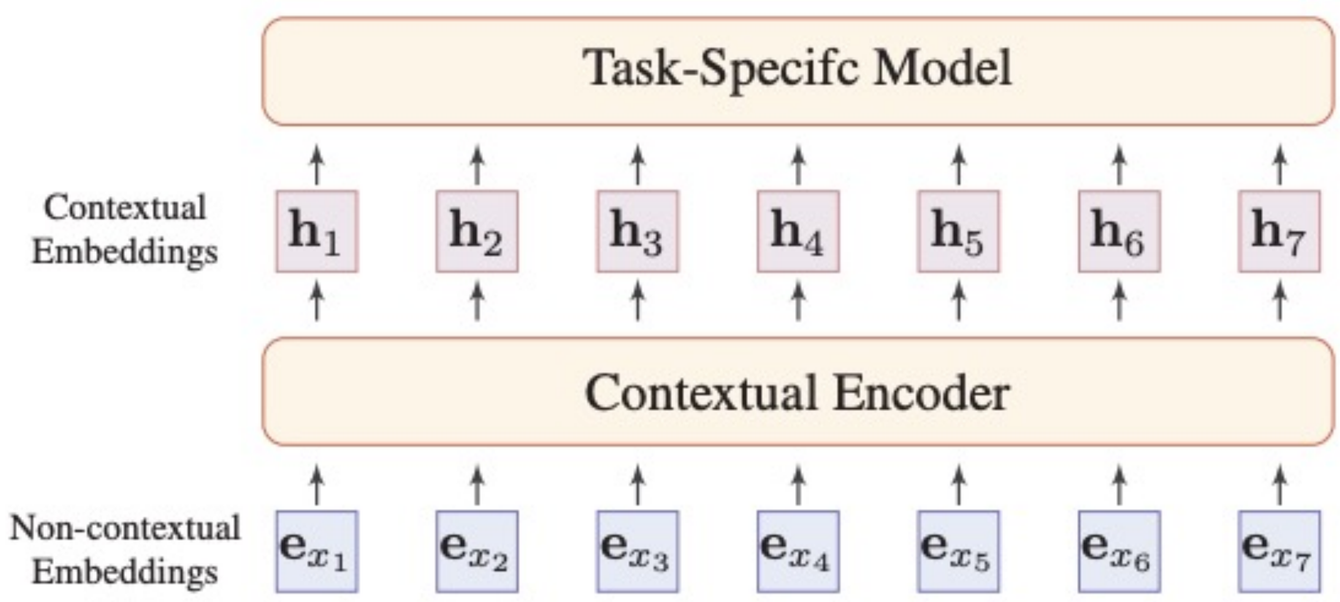
What?	➢ Models are likely to overfit training data ➢ Models fail to generalize well
Why?	➢ Lack of large-scale annotation data ➢ Model has plenty of parameters
How?	➢ Design PLMs ➢ Learn universal representations

Outline

- Overview
  - Motivation
  - Basic Concepts of PLMs
  - Non-contextual Embeddings for Data Preparation
  - Contextual Embeddings for Data Preparation
  - Domain Adaptation
  - Unified Data Matching
- Challenges and Open Problems

Basic Concepts

□Two types



Non-Contextual	Contextual
Static embedding	Dynamic embedding
Out of vocabulary words	Distinguish different semantics

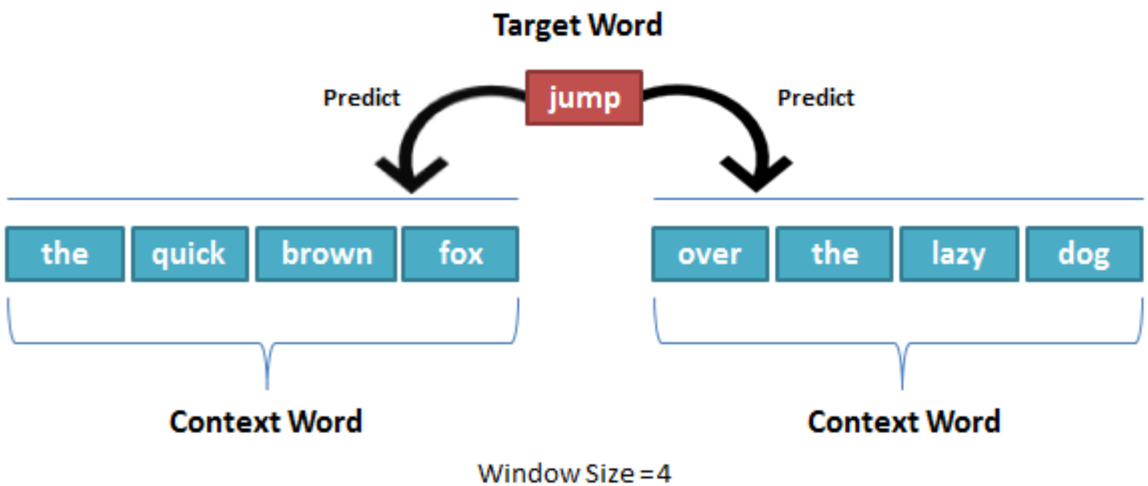
Main Difference

“Apple is really delicious”

“Apple phone is very useful”

Non-contextual Embeddings

- Learn good embeddings without much considering the downstream task
  - **Skip-Gram: capture sementics using nearby words**

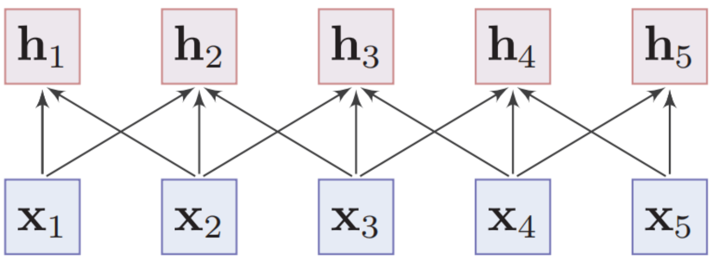


$$\mathcal{L} = -\log \mathbb{P}(w_{c,1}, w_{c,2}, \ldots, w_{c,C} | w_o) = -\log \prod_{c=1}^{C} \mathbb{P}(w_{c,i} | w_o)$$

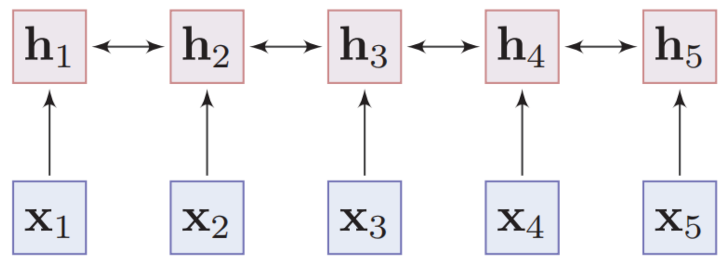
Contextual Embeddings

☐Learn contextual embeddings

$$[\mathbf{h}_1, \mathbf{h}_2, \cdots, \mathbf{h}_T] = f_{\text{enc}}(x_1, x_2, \cdots, x_T)$$

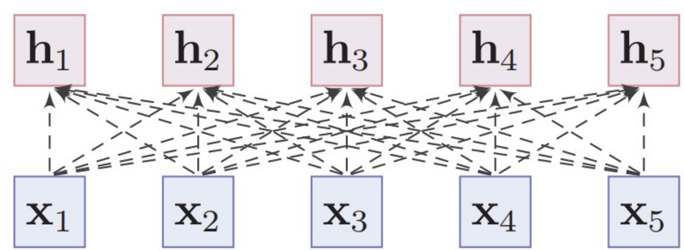


(a) Convolutional Model



(b) Recurrent Model

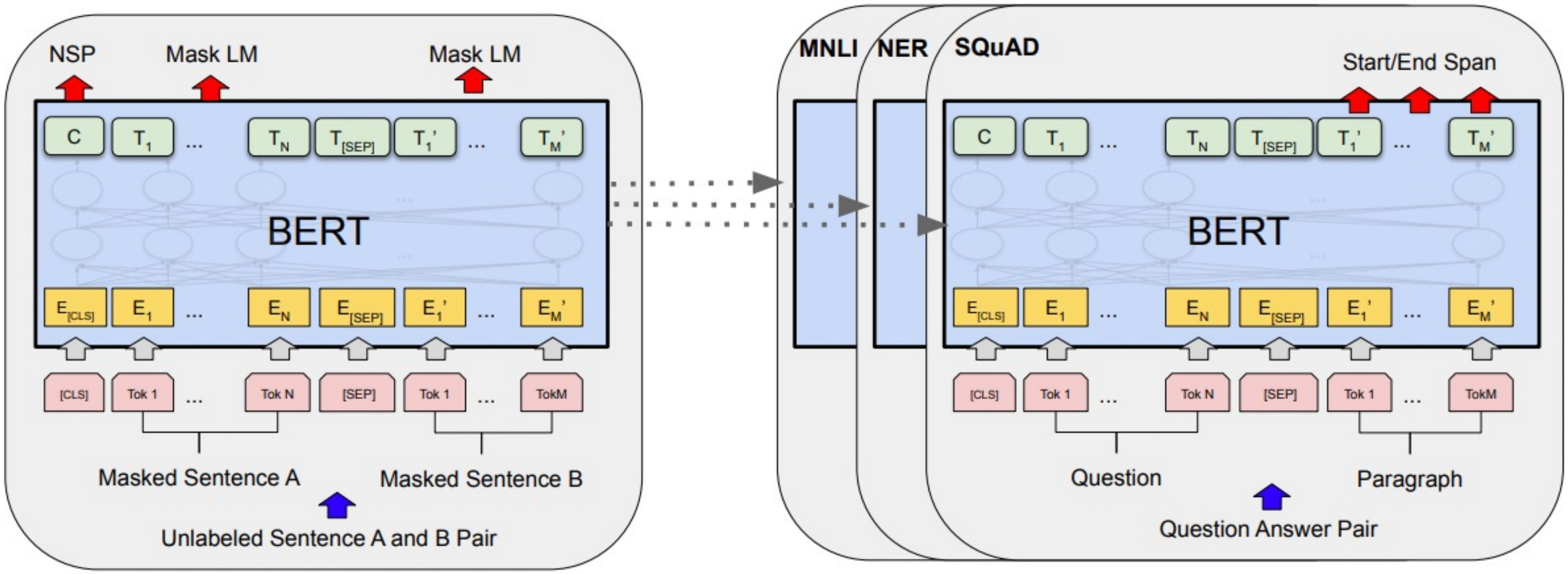
- ➢ Convolutional Model: Aggregate local info from neighbours
- ➢ Recurrent Model: Bi-directional LSTM, GRUs
- ➢ Transformer:Model the relation of every two words



(c) Fully-Connected Self-Attention Model

Contextual Embeddings

☐BERT



Pre-training Fine-Tuning

Pre-train:

- Masked LM: mask a random percentage of tokens and try to predict those masked tokens.
- Next Sentence Prediction: A $\longrightarrow$ B(50%) A $\not\rightarrow$ B(50%)

Fine-tune:

- Swapping out the appropriate input and outputs.

Outline

- ☐ **Overview**
  - **Motivation**
  - **Basic Concepts of PLMs**
  - **Word Embeddings for Data Preparation**
  - Contextual Embeddings for Data Preparation
  - Domain Adaptation
  - Unified Data Matching

- ☐ Challenges and Open Problems

Word Embeddings for Data Preparation

☐Entity Matching

- As a core problem of data integration, entity matching is to determine whether two data instances refer to the same real-world entity.
  - E.g., matching products from two e-commerce websites

Table A

id	name	description	price
a_1	samsung 52 ' series 7 black flat ...	samsung 52 ' series 7 black flat panel lcd ...	NULL
a_2	sony 46 ' bravia ...	bravia z series ...	NULL
a_3	linksys wirelessn ...	security router ...	NULL

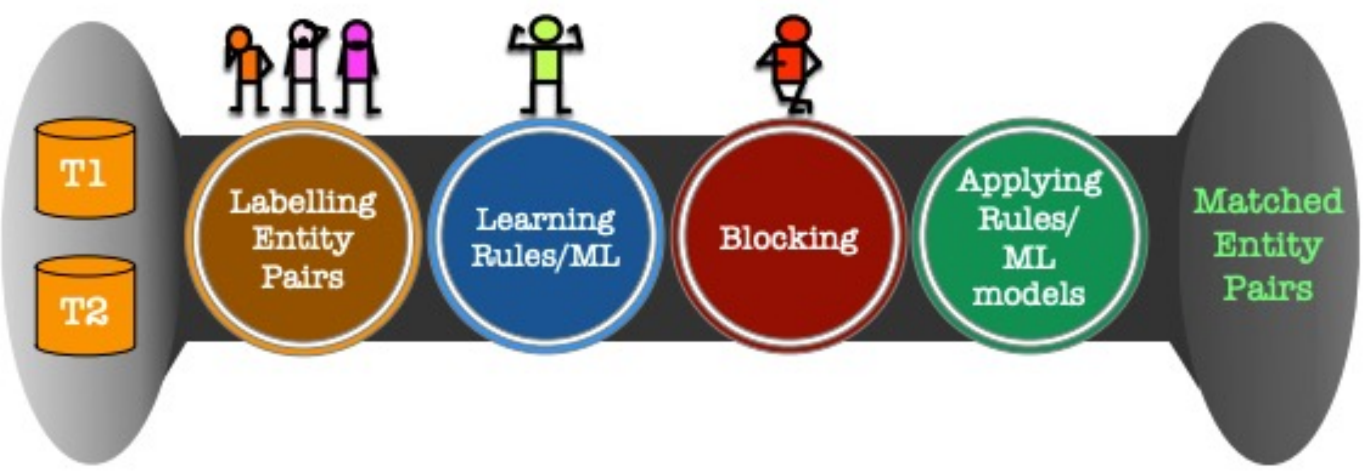
$(a_1, b_1, ?)$

$(a_2, b_2, ?)$

$(a_3, b_3, ?)$

Table B

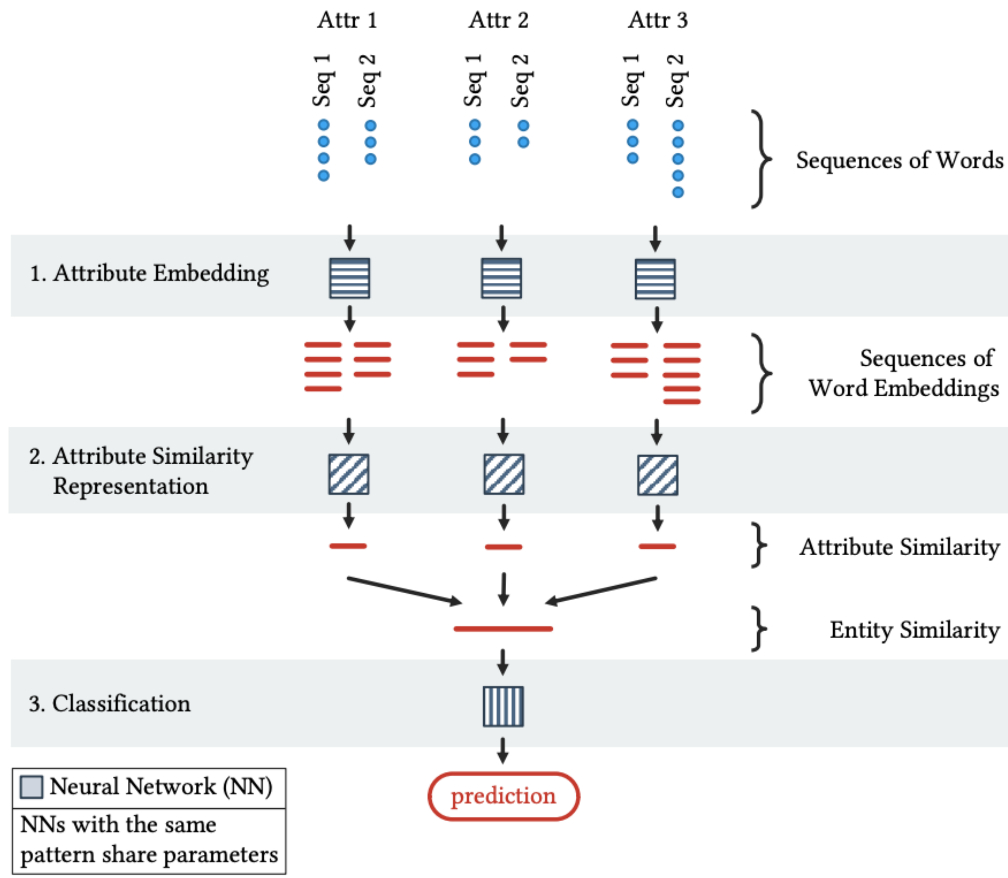
id	name	description	price
b_1	samsung ln52a750 ...	dynamic contrast ratio 120hz 6ms respons ...	2148.99
b_2	sony bravia ...	ntsc 16:9 1366 x 768 ...	597.72
b_3	linksys wirelessg ...	54mbps	NULL



Word Embeddings for Data Preparation

☐DeepER architecture

- Attribute embedding: Covert each pair of attributes to a pair of embeddings
  - Design space: word-based, character-based, learned
- Attribute similarity: Summairze the embedding pair to get the attribute similarity representation
  - Design space: RNN, attention, hybrid
- Classification: Compute the entity similarity representation.
  - Design space: MLP



Deep Learning for Entity Matching: A Design Space Exploration. SIGMOD 2018

Word Embeddings for Data Preparation

Table 3: Results for structured data.

Dataset	Model F_1 Score					ΔF_1
	SIF	RNN	Attention	Hybrid	Magellan	
BeerAdvo-RateBeer	58.1	72.2	64.0	72.7	78.8	-6.1
iTunes-Amazon$_1$	81.4	88.5	80.8	88.0	91.2	-2.7
Fodors-Zagats	100	100	82.1	100.0	100	0.0
DBLP-ACM$_1$	97.5	98.3	98.4	98.4	98.4	0.0
DBLP-Scholar$_1$	90.9	93.0	93.3	94.7	92.3	2.4
Amazon-Google	60.6	59.9	61.1	69.3	49.1	20.2
Walmart-Amazon$_1$	65.1	67.6	50.0	66.9	71.9	-4.3
Clothing$_1$	96.6	96.8	96.6	96.6	96.3	0.5
Electronics$_1$	90.2	90.6	90.5	90.2	90.1	0.5
Home$_1$	87.7	88.4	88.7	88.3	88.0	0.7
Tools$_1$	91.8	93.1	93.2	92.9	92.6	0.6

Table 4: Results for textual data (w. informative attributes).

Dataset	Model F_1 Score					ΔF_1
	SIF	RNN	Attention	Hybrid	Magellan	
Abt-Buy	35.1	39.4	56.8	62.8	43.6	19.2
Clothing$_2$	84.7	85.3	85.0	85.5	82.5	3.0
Electronics$_2$	90.4	92.2	91.5	92.1	85.3	6.9
Home$_2$	84.5	85.5	86.1	86.6	82.3	4.3
Tools$_2$	92.9	94.5	93.8	94.3	90.2	4.3

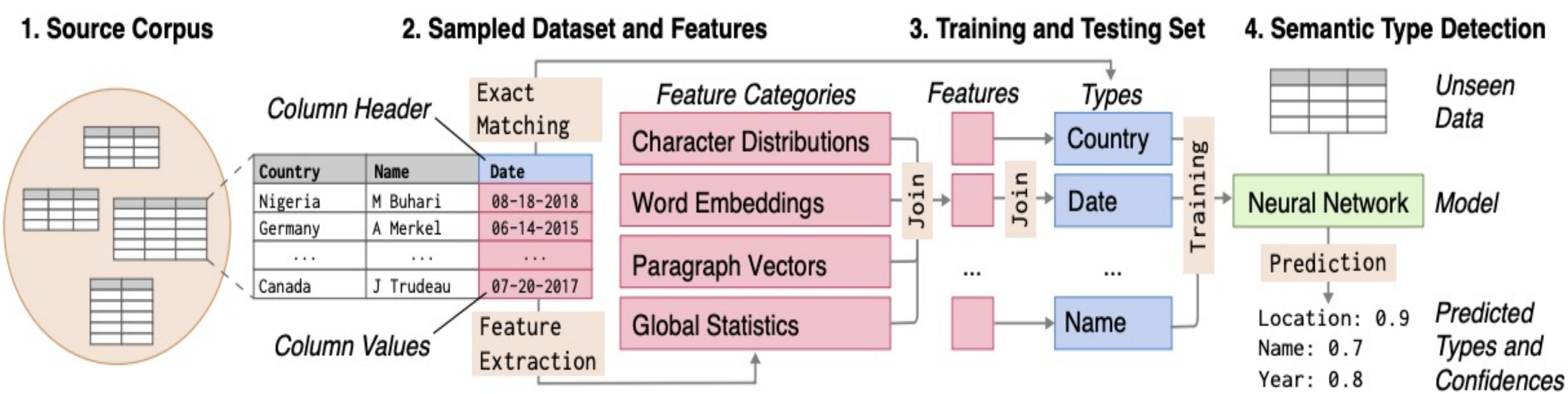
Table 5: Results for textual data (w.o. informative attributes).

Dataset	Model F_1 Score					ΔF_1
	SIF	RNN	Attention	Hybrid	Magellan	
Abt-Buy	32.0	38.5	55.0	47.7	33.0	22.0
Company	71.2	85.6	89.8	92.7	79.8	12.9
Clothing$_2$	84.6	84.4	84.6	84.3	78.8	5.8
Electronics$_2$	89.6	90.4	90.8	91.1	82.0	9.1
Home$_2$	84.0	84.8	83.7	85.4	74.1	11.3
Tools$_2$	91.6	92.5	92.6	93.0	84.4	8.6

Word Embeddings for Data Preparation

☐ Column Type Annotation

- ➢ Annotate the type of each attribute in the relational table
- ➢ Consider the embeddings of both attributes and cell values.



Sherlock: A deep learning approach to semantic data type detection. KDD, 2019.

Word Embeddings for Data Preparation

☐ Column Type Annotation

➢ Multiple columns should be considered when annotating a column.

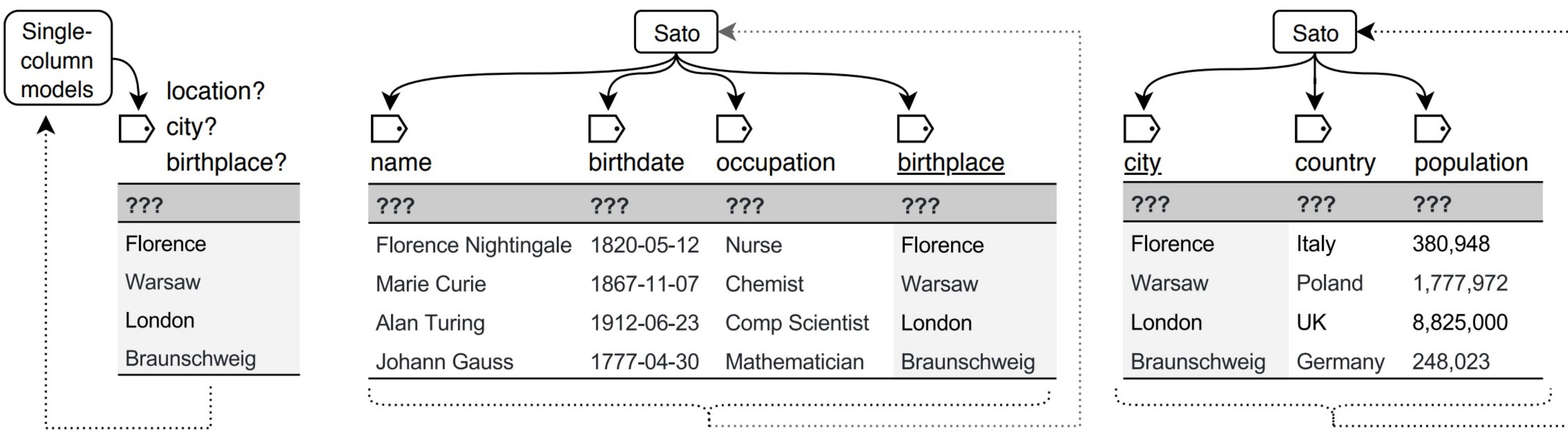


Table A: Influential people in history

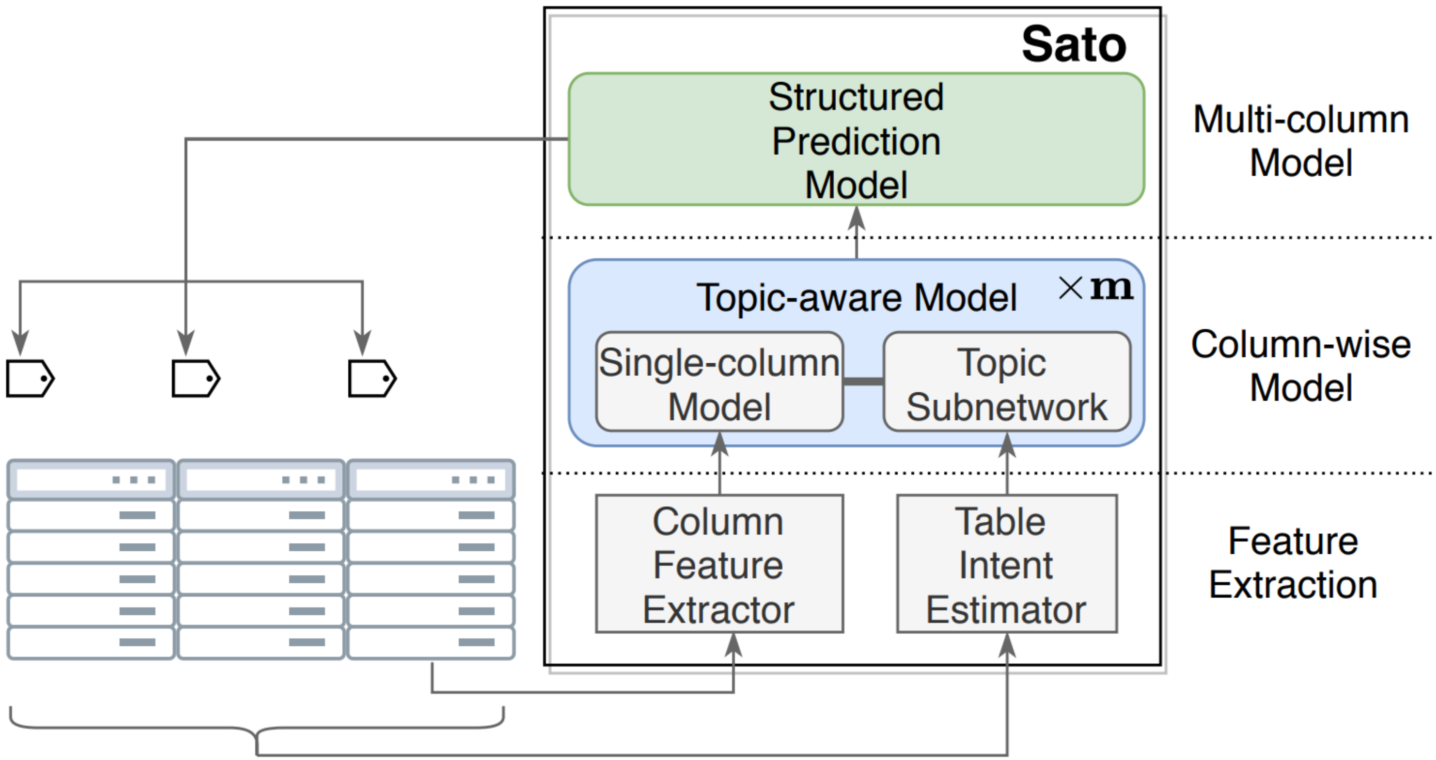
Table B: Cities in Europe

Sato: Contextual Semantic Type Detection in Tables. VLDB 2018

Word Embeddings for Data Preparation

☐ Column Type Annotation

- ➢ Multiple columns should be considered when annotating a column.



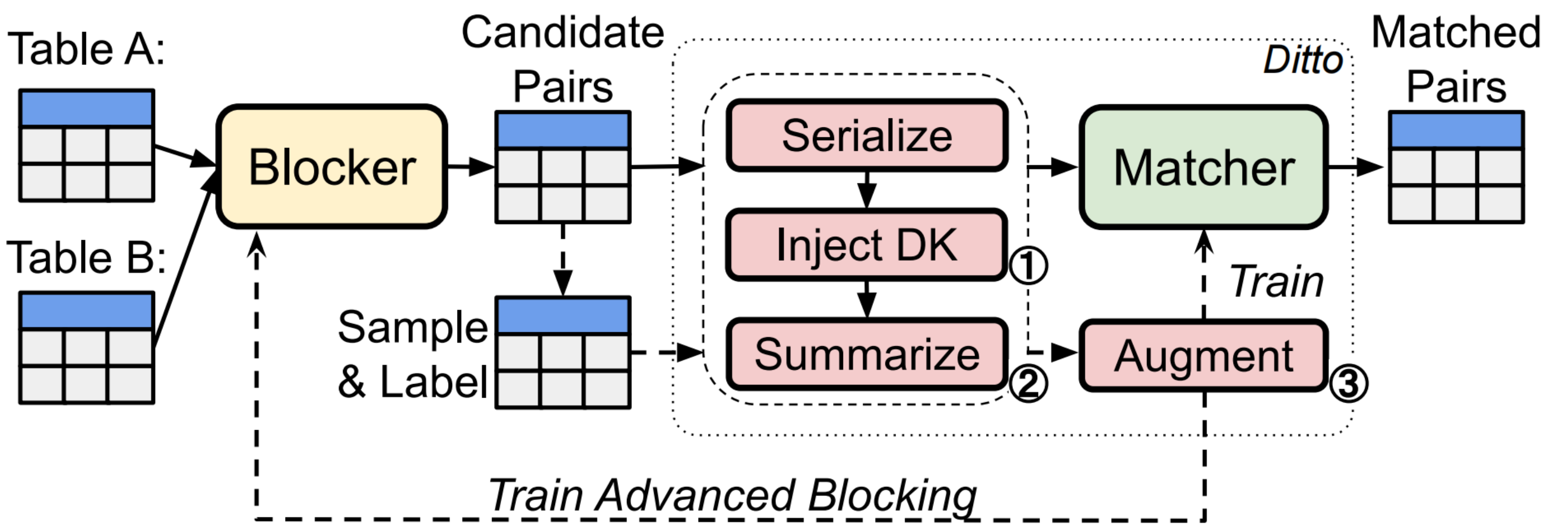
Word Embeddings for Data Preparation

	Multi-column tables $\mathcal{D}_{mult}$		All tables $\mathcal{D}$	
	Macro average F_1	Support-weighted F_1	Macro average F_1	Support-weighted F_1
Base	0.642 $\pm$0.015	0.879 $\pm$0.002	0.692 $\pm$0.007	0.867 $\pm$0.003
Sato	**0.735** $\pm$0.022 (14.4%↑)	**0.925** $\pm$0.003 (5.3%↑)	**0.756** $\pm$0.011 (9.3%↑)	**0.902** $\pm$0.002 (4.0%↑)
$\text{Sato}_{\text{noStruct}}$	0.713 $\pm$0.025 (11.0%↑)	0.909 $\pm$0.002 (3.5%↑)	0.746 $\pm$0.011 (7.8%↑)	0.891 $\pm$0.003 (2.8%↑)
$\text{Sato}_{\text{noTopic}}$	0.681 $\pm$0.016 (6.6%↑)	0.907 $\pm$0.002 (3.2%↑)	0.711 $\pm$0.006 (2.9%↑)	0.884 $\pm$0.002 (2.0%↑)

Outline

- **Overview**
  - **Motivation**
  - **Basic Concepts of PLMs**
  - **Word Embeddings for Data Preparation**
  - **Contextual Embeddings for Data Preparation**
  - Domain Adaptation
  - Unified Data Matching
- Challenges and Open Problems

Contextual Embeddings for Data Preparation



☐ Improvement

- ➢ Inject domain knowledge: Span typing, Span normalization
- ➢ Long entries summarization: Feed only the most informative tokens
- ➢ Data augmentation

Operator	Explanation
span_del	Delete a randomly sampled span of tokens
span_shuffle	Randomly sample a span and shuffle the tokens' order
attr_del	Delete a randomly chosen attribute and its value
attr_shuffle	Randomly shuffle the orders of all attributes
entry_swap	Swap the order of the two data entries e and e'

Deep Entity Matching with Pre-Trained Language Models VLDB 2020

Contextual Embeddings for Data Preparation

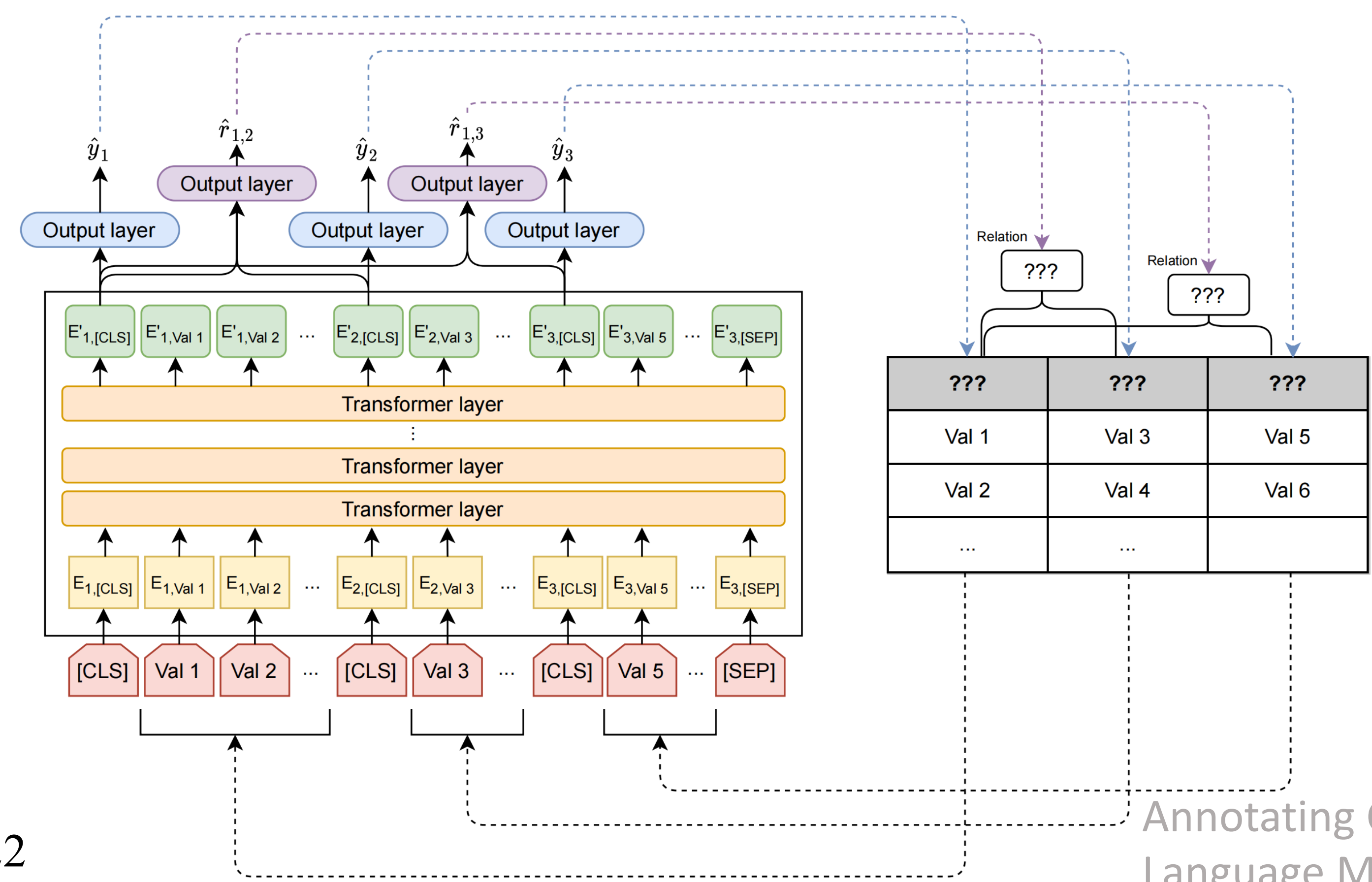
Table 5: F1 scores on the ER-Magellan EM datasets. The numbers of DeepMatcher+ (DM+) are the highest available found in [17, 23, 34] or re-produced by us.

Datasets	DM+	Ditto	Ditto (DA)	Ditto (DK)	Baseline	Size
Structured						
Amazon-Google	70.7	75.58 (+4.88)	75.08	74.67	74.10	11,460
Beer	78.8	94.37 (+15.57)	87.21	90.46	84.59	450
DBLP-ACM	98.45	98.99 (+0.54)	99.17	99.10	98.96	12,363
DBLP-Google	94.7	95.6 (+0.9)	95.73	95.80	95.84	28,707
Fodors-Zagats	100	100.00 (+0.0)	100.00	100.00	98.14	946
iTunes-Amazon	91.2	97.06 (+5.86)	97.40	97.80	92.28	539
Walmart-Amazon	73.6	86.76 (+13.16)	85.50	83.73	85.81	10,242
Dirty						
DBLP-ACM	98.1	99.03 (+0.93)	98.94	99.08	98.92	12,363
DBLP-Google	93.8	95.75 (+1.95)	95.47	95.57	95.44	28,707
iTunes-Amazon	79.4	95.65 (+16.25)	95.29	94.48	92.92	539
Walmart-Amazon	53.8	85.69 (+31.89)	85.49	80.67	82.56	10,242
Textual						
Abt-Buy	62.8	89.33 (+26.53)	89.79	81.69	88.85	9,575
Company	92.7	93.85 (+1.15)	93.69	93.15	41.00	112,632

Contextual Embeddings for Data Preparation

☐ Column Type Annotation

- ➢ Serialize the entire table into a sequence of tokens.
- ➢ Simultaneously identify column type and column relations.
  - Relation (person, location) → birthplace



Annotating Columns with Pre-trained Language Models. SIGMOD 2022

Contextual Embeddings for Data Preparation

Table 3: Performance on the WikiTable dataset.

Method	Col type			Col rel		
	P	R	F1	P	R	F1
Sherlock	88.40	70.55	78.47	–	–	–
TURL	90.54	87.23	88.86	91.18	90.69	90.94
Doduo	**92.69**	**92.21**	**92.45**	**91.97**	**91.47**	**91.72**
TURL+metadata	92.75	92.63	92.69	92.90	93.80	93.35
Doduo+metadata	93.25	92.34	92.79	91.20	94.50	92.82

Table 4: Performance on the VizNet dataset.

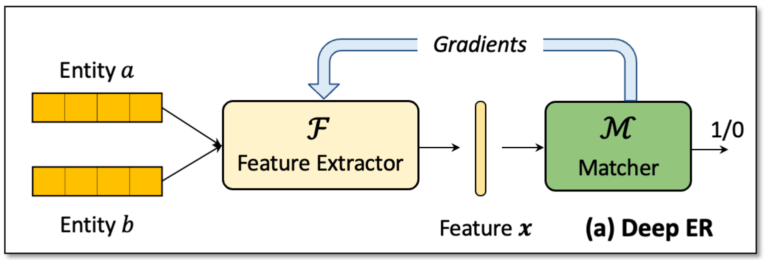
Method	Full		Multi-column only	
	Macro F1	Micro F1	Macro F1	Micro F1
Sherlock	69.2	86.7	64.2	87.9
Sato	75.6	88.4	73.5	92.5
Doduo	**84.6**	**94.3**	**83.8**	**96.4**

Outline

- Overview
  - Motivation
  - Basic Concepts of PLMs
  - Word Embeddings for Data Preparation
  - Contextual Embeddings for Data Preparation
  - Domain Adaptation
  - Unified Data Matching
- Challenges and Open Problems

DL-based Entity Matching(Deep EM)

- The Framework of Deep EM :
  - **Feature Extractor** converts entity pair (a,b) into d-dimensional vector-based representation (feature).
  - **Matcher** takes the feature of entity pair as input, and predicts whether they match or not.



$$\hat{y} = \mathcal{M}(\mathbf{x}) = \mathcal{M}(\mathcal{F}(a, b))$$

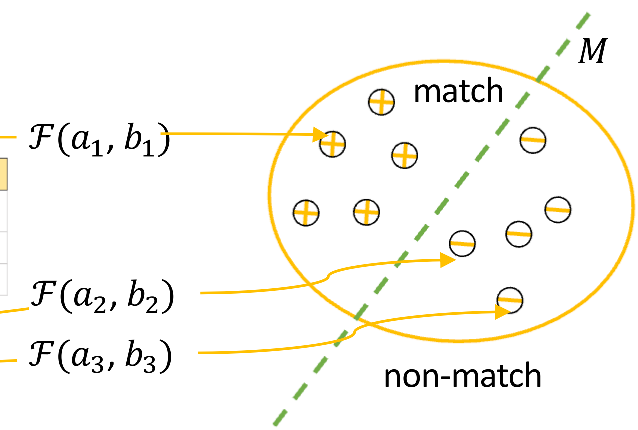
Table A: entity a

id	title	category	brand	price
a_1	balt wheasel ...	stationery ...	balt	239.88
a_2	kodak esp ...	printers	NULL	58.0
a_3	hp q3675a ...	printers	hp	194.84

$(a_1, b_1, 1)$
$(a_2, b_2, 0)$
$(a_3, b_3, 0)$

Table B: entity b

id	title	category	brand	price
b_1	balt inc. ...	laminating ...	mayline	134.45
b_2	kodak esp 7 ...	kodak	NULL	149.29
b_3	hewlett ...	cleaning repair	hp	NULL



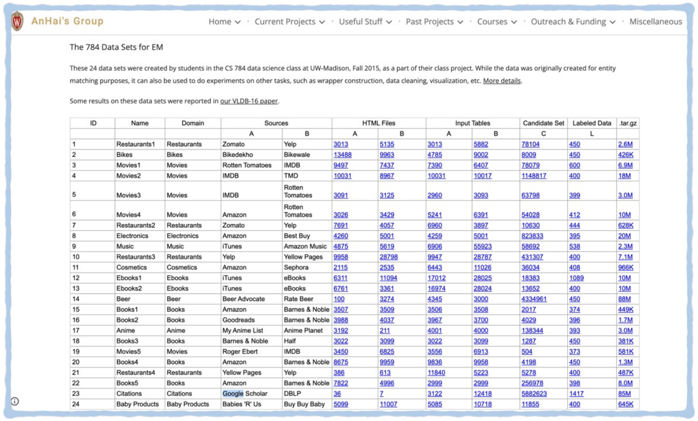
Problem: DL-based methods need a large amount of labeled training data.

Opportunity of Reusing Well-Labeled EM Datasets

➢ There are many well-labeled entity matching datasets, either public on the Web or available in enterprises

- E.g., Magellan datasets and WDC datasets

Magellan datasets



AnHai's Group Home · Current Projects · Useful Stuff · Past Projects · Courses · Outreach & Funding · Miscellaneous

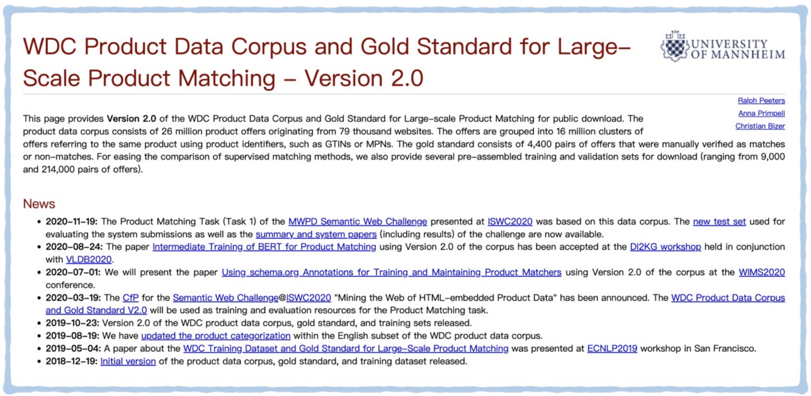
The 784 Data Sets for EM

These 24 data sets were created by students in the CS 784 data science class at UW-Madison, Fall 2015, as a part of their class project. While the data was originally created for entity matching purposes, it can also be used to do experiments on other tasks, such as wrapper construction, data cleaning, visualization, etc. More details.

Some results on these data sets were reported in our VLDB-16 paper.

ID	Name	Domain	Sources		HTML Files		Input Tables		Candidate Set	Labeled Data	.tar.gz
			A	B	A	B	A	B	C	L	
1	Restaurants1	Restaurants	Zomato	Yelp	3013	5135	3013	5882	78104	450	2.6M
2	Bikes	Bikes	Bikedekho	Bikewale	13488	9963	4785	9002	8009	450	426K
3	Movies1	Movies	Rotten Tomatoes	IMDB	9497	7437	7390	6407	78079	600	6.9M
4	Movies2	Movies	IMDB	TMD	10031	8967	10031	10017	1148817	400	18M
5	Movies3	Movies	IMDB	Rotten Tomatoes	3091	3125	2960	3093	63798	399	3.0M
6	Movies4	Movies	Amazon	Rotten Tomatoes	3026	3429	5241	6391	54028	412	10M
7	Restaurants2	Restaurants	Zomato	Yelp	7691	4057	6960	3897	10630	444	628K
8	Electronics	Electronics	Amazon	Best Buy	4260	5001	4259	5001	823833	395	20M
9	Music	Music	iTunes	Amazon Music	4875	5619	6906	55923	58692	538	2.3M
10	Restaurants3	Restaurants	Yelp	Yellow Pages	9958	28798	9947	28787	431307	400	7.1M
11	Cosmetics	Cosmetics	Amazon	Sephora	2115	2535	6443	11026	36034	408	966K
12	Ebooks1	Ebooks	iTunes	eBooks	6311	11094	17012	28025	18383	1089	10M
13	Ebooks2	Ebooks	iTunes	eBooks	6761	3361	16974	28024	13652	400	10M
14	Beer	Beer	Beer Advocate	Rate Beer	100	3274	4345	3000	4334961	450	88M
15	Books1	Books	Amazon	Barnes & Noble	3507	3509	3506	3508	2017	374	449K
16	Books2	Books	Goodreads	Barnes & Noble	3988	4037	3967	3700	4029	396	1.7M
17	Anime	Anime	My Anime List	Anime Planet	3192	211	4001	4000	138344	393	3.0M
18	Books3	Books	Barnes & Noble	Half	3022	3099	3022	3099	1287	450	381K
19	Movies5	Movies	Roger Ebert	IMDB	3450	6825	3556	6913	504	373	581K
20	Books4	Books	Amazon	Barnes & Noble	8675	9959	9836	9958	4198	450	1.3M
21	Restaurants4	Restaurants	Yellow Pages	Yelp	386	613	11840	5223	5278	400	487K
22	Books5	Books	Amazon	Barnes & Noble	7822	4996	2999	2999	256978	398	8.0M
23	Citations	Citations	Google Scholar	DBLP	36	7	3122	12418	5882623	1417	85M
24	Baby Products	Baby Products	Babies 'R' Us	Buy Buy Baby	5099	11007	5085	10718	11855	400	645K

WDC datasets



WDC Product Data Corpus and Gold Standard for Large–Scale Product Matching – Version 2.0

UNIVERSITY OF MANNHEIM

Ralph Peeters
Anna Primpeli
Christian Bizer

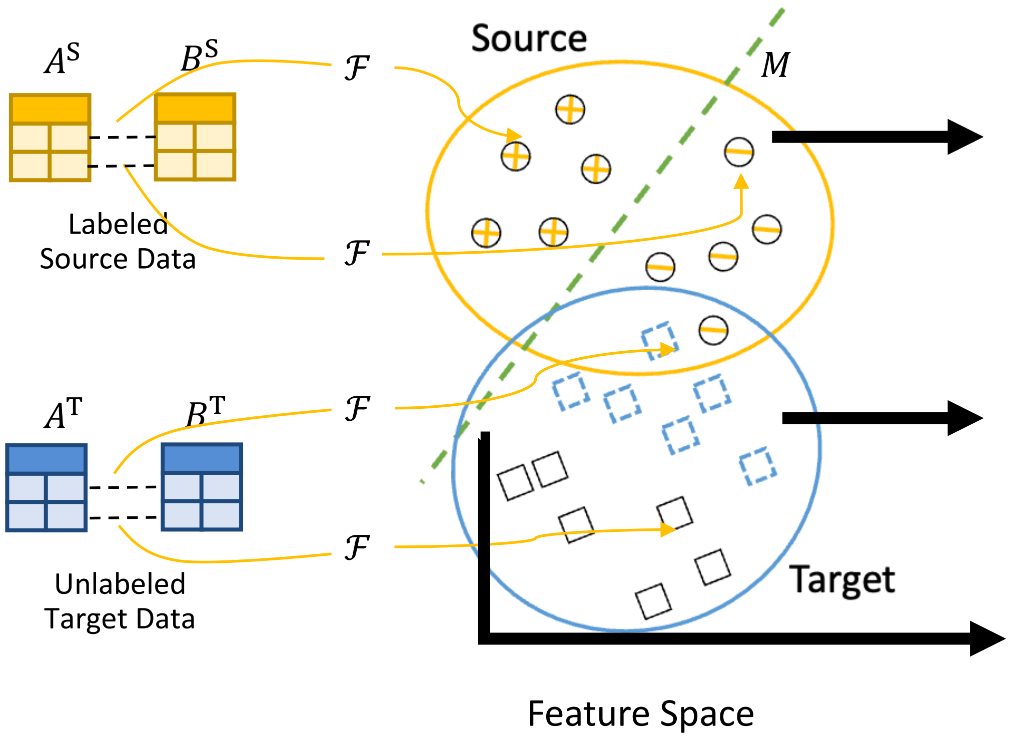
This page provides **Version 2.0** of the WDC Product Data Corpus and Gold Standard for Large–scale Product Matching for public download. The product data corpus consists of 26 million product offers originating from 79 thousand websites. The offers are grouped into 16 million clusters of offers referring to the same product using product identifiers, such as GTINs or MPNs. The gold standard consists of 4,400 pairs of offers that were manually verified as matches or non–matches. For easing the comparison of supervised matching methods, we also provide several pre–assembled training and validation sets for download (ranging from 9,000 and 214,000 pairs of offers).

News

- **2020–11–19:** The Product Matching Task (Task 1) of the MWPD Semantic Web Challenge presented at ISWC2020 was based on this data corpus. The new test set used for evaluating the system submissions as well as the summary and system papers (including results) of the challenge are now available.
- **2020–08–24:** The paper Intermediate Training of BERT for Product Matching using Version 2.0 of the corpus has been accepted at the DI2KG workshop held in conjunction with VLDB2020.
- **2020–07–01:** We will present the paper Using schema.org Annotations for Training and Maintaining Product Matchers using Version 2.0 of the corpus at the WIMS2020 conference.
- **2020–03–19:** The CfP for the Semantic Web Challenge@ISWC2020 "Mining the Web of HTML–embedded Product Data" has been announced. The WDC Product Data Corpus and Gold Standard V2.0 will be used as training and evaluation resources for the Product Matching task.
- **2019–10–23:** Version 2.0 of the WDC product data corpus, gold standard, and training sets released.
- **2019–08–19:** We have updated the product categorization within the English subset of the WDC product data corpus.
- **2019–05–04:** A paper about the WDC Training Dataset and Gold Standard for Large–Scale Product Matching was presented at ECNLP2019 workshop in San Francisco.
- **2018–12–19:** Initial version of the product data corpus, gold standard, and training dataset released.

Can we reuse these $\underbrace{\text{labeled EM datasets}}_{\text{Source}}$ for a new $\underbrace{\text{unlabeled EM dataset}}_{\text{Target}}$?

Directly Reusing Feature Extractor and Matcher Trained on Labeled Source?



Step1: Getting the **Feature Extractor** $\boldsymbol{\mathcal{F}}$ and the **Matcher** $\boldsymbol{M}$ trained by labeled source data.

Step2: Mapping the unlabeled target data into the feature space.

Step3: Predicting the target data with $\boldsymbol{M}$ directly.

⚠️ The $\boldsymbol{M}$ fails to predict the target data due to the distribution change or domain shift of feature space.

Distribution Change or Domain Shift

- Similar domains
  - Source (Citation): DBLP-ACM (Title, Authors, Venue, Year)
  - Target (Citation): DBLP-Scholar (Title, Authors, Venue, Year)
- Different domains
  - Source (Music): iTunes-Amazon (Album Name, Artist Name, Song Name, Album Price, …)
  - Target (Citation): DBLP-Scholar (Title, Authors, Venue, Year)

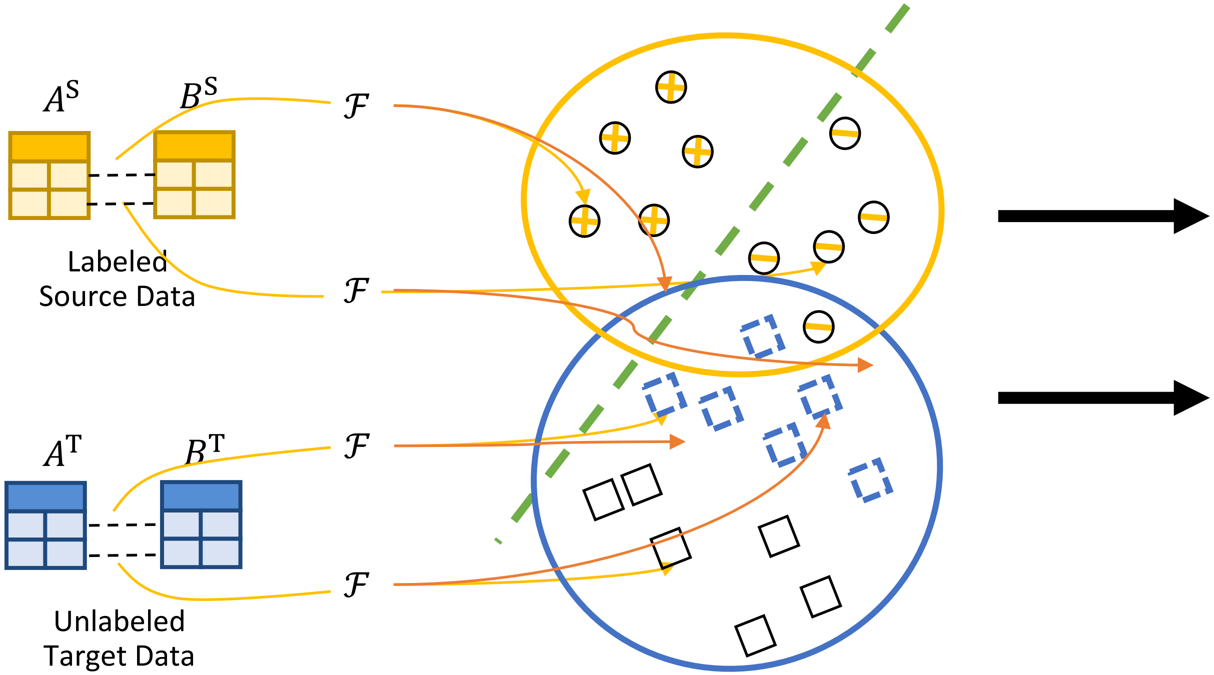
Training only with source vs. Training with target (F1)

	Source	Target	Training only with source	Training with target
Similar domains	DBLP-ACM	DBLP-Scholar	77.8	95.6
Different domains	iTunes-Amazon	DBLP-Scholar	68.2	95.6

Can we better reuse the source?

Domain Adaptation (DA) for Deep EM

➢ Learn domain-invariant and discriminative features.



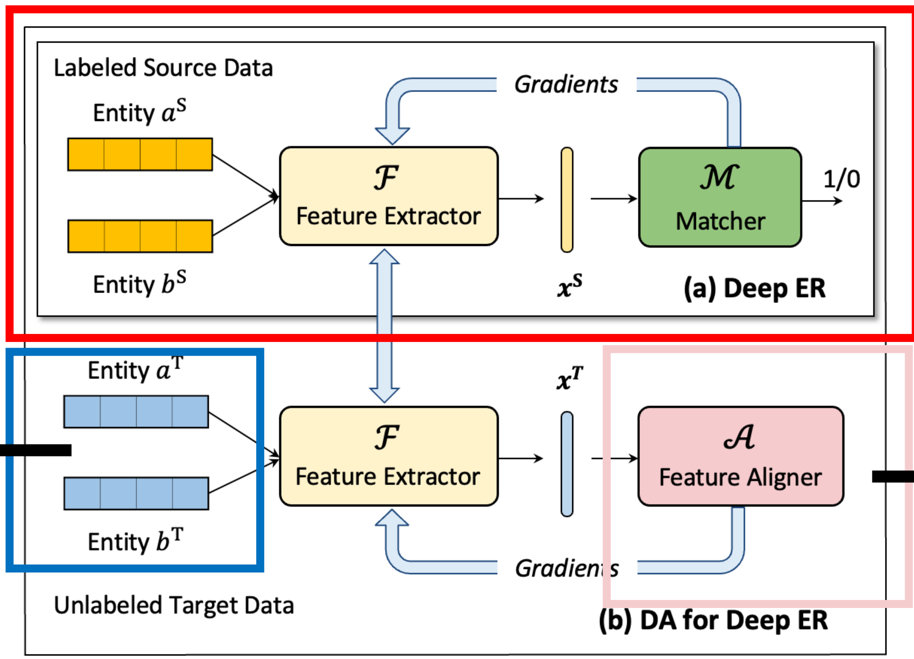
Domain-invariant : reducing distribution change.

Discriminative : extracting discriminative information.

Whether DA can be used for EM tasks?

DADER Framework

- ➢ Feature Extractor and Matcher
- ➢ Feature Aligner: the key module to realize domain adaptation.



① Training $\mathcal{F}$ and M with labeled source data.

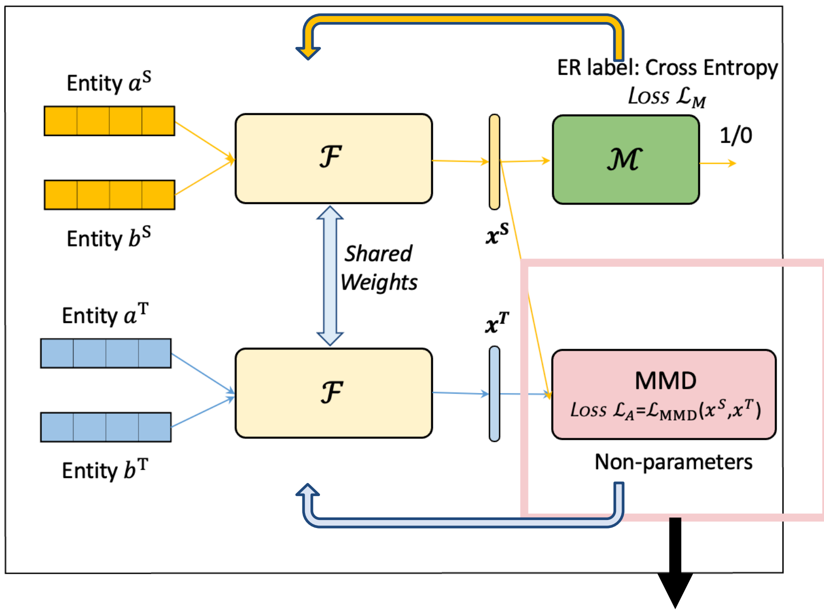
② Utilizing unlabeled target data.

③ Adjusting $\mathcal{F}$ and M to align distributions of source and target, such that $\mathcal{F}$ and M can work on target.

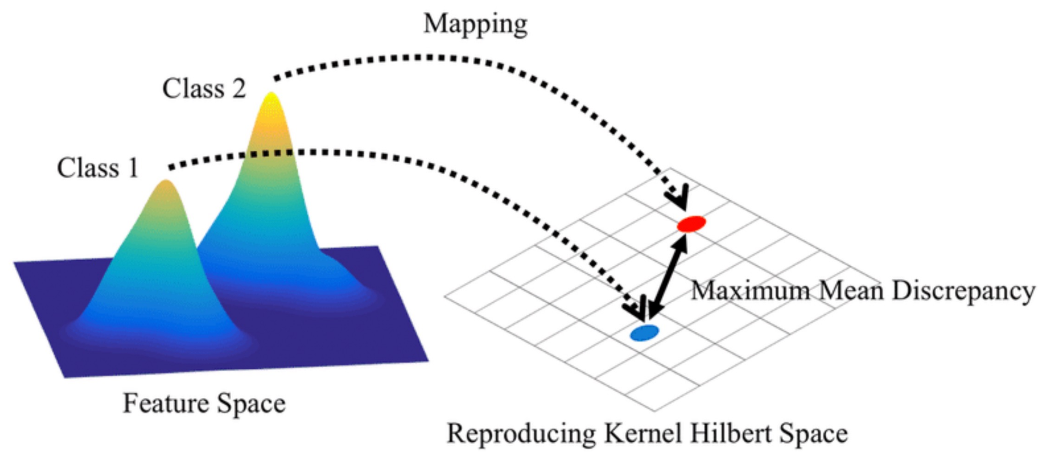
DADER: Hands-Off Entity Resolution with Domain Adaptation. VLDB 2022

Representative Method: MMD (Discrepancy-based)

➤ Feature Aligner is a function to measure **maximum mean discrepancy**.



During training, the Maximum Mean Discrepancy of source and target feature spaces is **computed** and **reduced**. The smaller the MMD, the more similar the distributions.

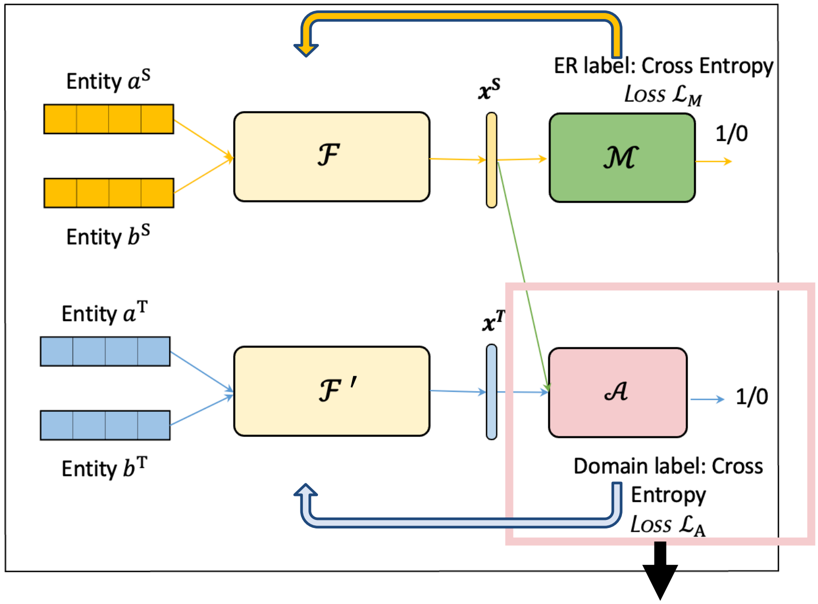


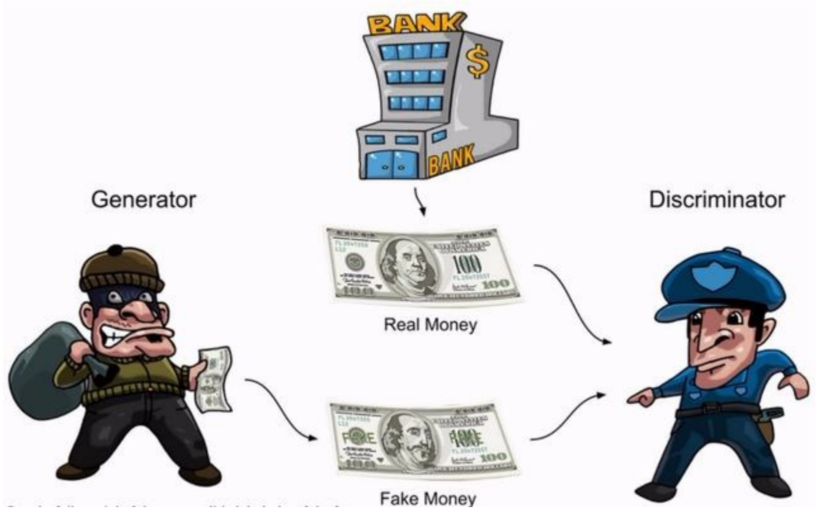
$$\mathcal{L}_{\mathrm{MMD}} = \sup_{\|\phi\|_H \le 1} \|E_{\mathbf{x}^{\mathrm{S}} \sim p_{\mathrm{S}}}[\phi(\mathbf{x}^{\mathrm{S}})] - E_{\mathbf{x}^{\mathrm{T}} \sim p_{\mathrm{T}}}[\phi(\mathbf{x}^{\mathrm{T}})]\|_H^2$$

Representative Method: InvGAN (Adversarial-based)

- Feature Aligner is a binary domain classifier to discriminate source/target dataset.

During training, the optimization objective of Feature Aligner is to **minimize the domain classification loss**, while Feature Extractor is to generate the **indistinguishable features** that confuse Feature Aligner.





$$\min_{\mathcal{F},\mathcal{M}} \max_{\mathcal{A}} V(\mathcal{F}, \mathcal{M}, \mathcal{A}) = \mathcal{L}_M(\mathcal{F}, \mathcal{M}) + \beta \mathcal{L}_A(\mathcal{F}, \mathcal{A}),$$

$$\mathcal{L}_A = E_{x^S \sim \mathcal{D}^S} \log \mathcal{A}(\mathcal{F}(x^S)) + E_{x^T \sim \mathcal{D}^T} \log(1 - \mathcal{A}(\mathcal{F}(x^T))),$$

Experiment

	Datasets		NoDA	Discrepancy-based		Adversarial-based			Reconstruction-based	ΔF1
	Source	Target		MMD	K-order	GRL	InvGAN	InvGAN+KD	ED	
Similar Domain	Walmart-Amazon	Abt-Buy	65.8	72.6	68.3	68.4	56.0	69.6	39.4	6.8
	Abt-Buy	Walmart-Amazon	56.9	71.1	62.0	66.3	47.5	63.5	45.7	14.2
	DBLP-Scholar	DBLP-ACM	97.2	97.2	96.2	96.9	97.1	97.2	96.8	0.0
	DBLP-ACM	DBLP-Scholar	77.8	91.5	88.9	84.2	92.1	92.3	90.5	14.5
	Zomato-Yelp	Fodors-Zagats	85.4	92.2	87.7	89.1	94.5	93.5	78.0	9.1
	Fodors-Zagats	Zomato-Yelp	47.6	64.5	72.6	49.6	29.7	75.0	0.0	27.4
Different Domain	RottenTomatoes-IMDB	Abt-Buy	40.6	43.6	41.4	42.7	23.8	53.9	13.8	13.3
	RottenTomatoes-IMDB	Walmart-Amazon	38.4	41.5	41.9	49.0	35.1	49.4	30.7	11.0
	iTunes-Amazon	DBLP-ACM	80.3	94.5	86.9	92.1	57.7	94.4	77.5	14.1
	iTunes-Amazon	DBLP-Scholar	68.2	86.9	80.4	85.4	59.6	89.1	42.8	20.9
	Book2	Fodors-Zagats	49.6	91.5	78.2	84.2	93.5	93.4	78.1	43.9
	Book2	Zomato-Yelp	67.4	73.0	68.0	54.0	63.3	81.8	19.7	14.4

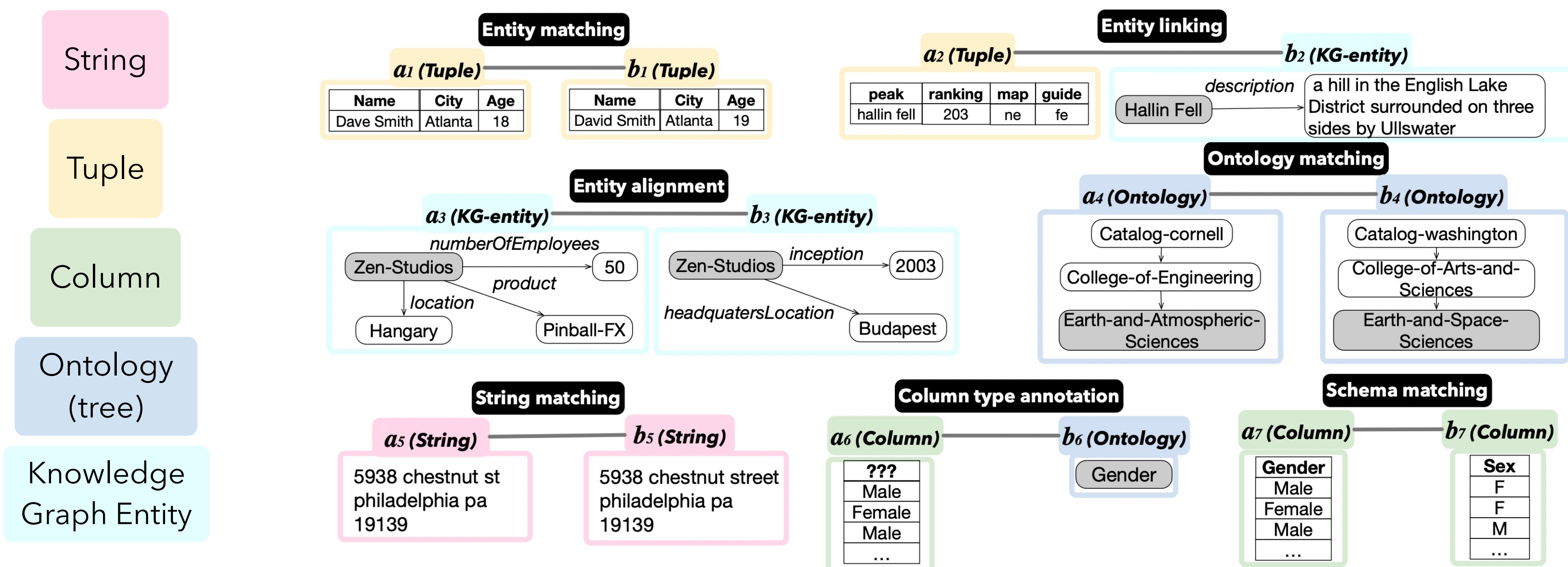
Outline

- **Overview**
  - Motivation
  - Basic Concepts of PLMs
  - Word Embeddings for Data Preparation
  - Contextual Embeddings for Data Preparation
  - Domain Adaptation
  - Unified Data Matching
- **Challenges and Open Problems**

Data Matching Tasks

- Data matching generally refers to the process of deciding whether two data elements are the same (a.k.a. a “match”)

Data Elements ***Seven Common Data Matching Tasks***



Unicorn: A Unified Multi-tasking Model for Supporting Matching Tasks in Data Integration. SIGMOD 2023

Existing Solutions

- Due to their importance, almost all matching tasks have been studied for decades, and remain to be important research topics.
  - DeepMatcher[1] and Ditto[2] for entity matching, Hybrid[3] and TURL[4] for entity linking , HNN+P2Vec[5] for column type annotation, etc.
  - Current solutions are task-specific or even dataset-specific
- Limitations of the specific models
  - The learned knowledge cannot be shared across different models
  - One model has to be learned for each task or dataset, which is inefficient and has a high monetary cost

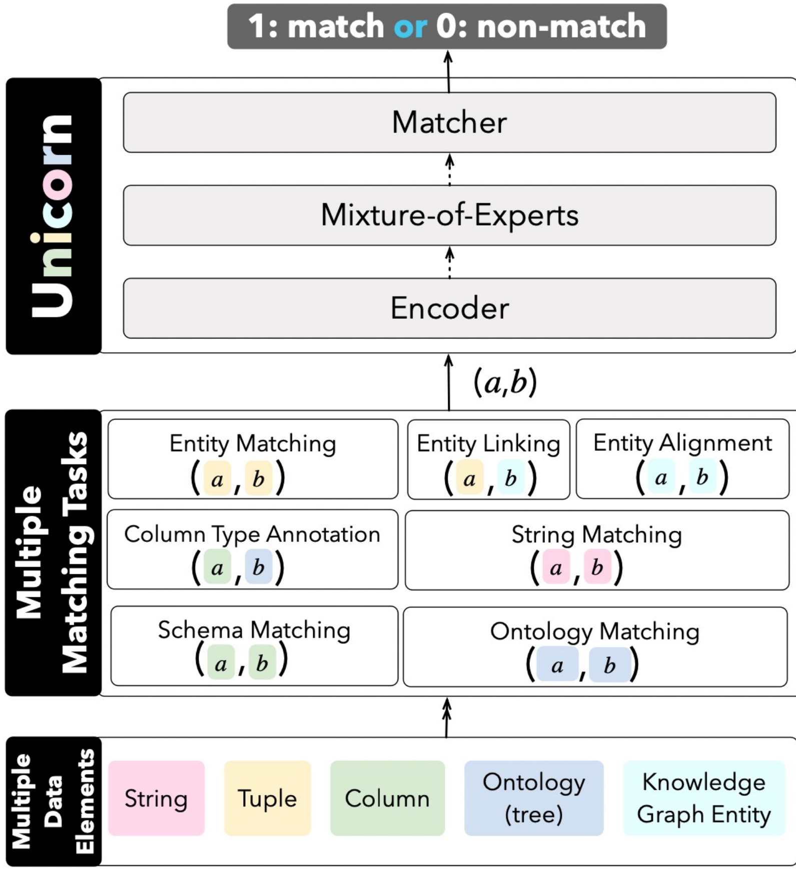
Task Type	Data	Previous SOTA (Labels)
Entity Matching	DBLP-Scholar	95.6 (22,965)
String Matching	Product	67.18 (1,020)
Entity Alignment	SRPRS: DBP-WD	99.6 (4,500)

[1] Mudgal S, Li H, et al. Deep learning for entity matching: A design space exploration. SIGMOD 2018.
[2] Li Y, Li J, et al. Deep entity matching with pre-trained language models. VLDB 2020.
[3] Efthymiou V, Hassanzadeh O, et al. Matching web tables with knowledge base entities: from entity lookups to entity embeddings. ISWC 2017.
[4] Deng X, Sun H, et al. Turl: Table understanding through representation learning. SIGMOD 2022.
[5] Chen J, Jiménez-Ruiz E, et al. Learning semantic annotations for tabular data. IJCAI 2019.

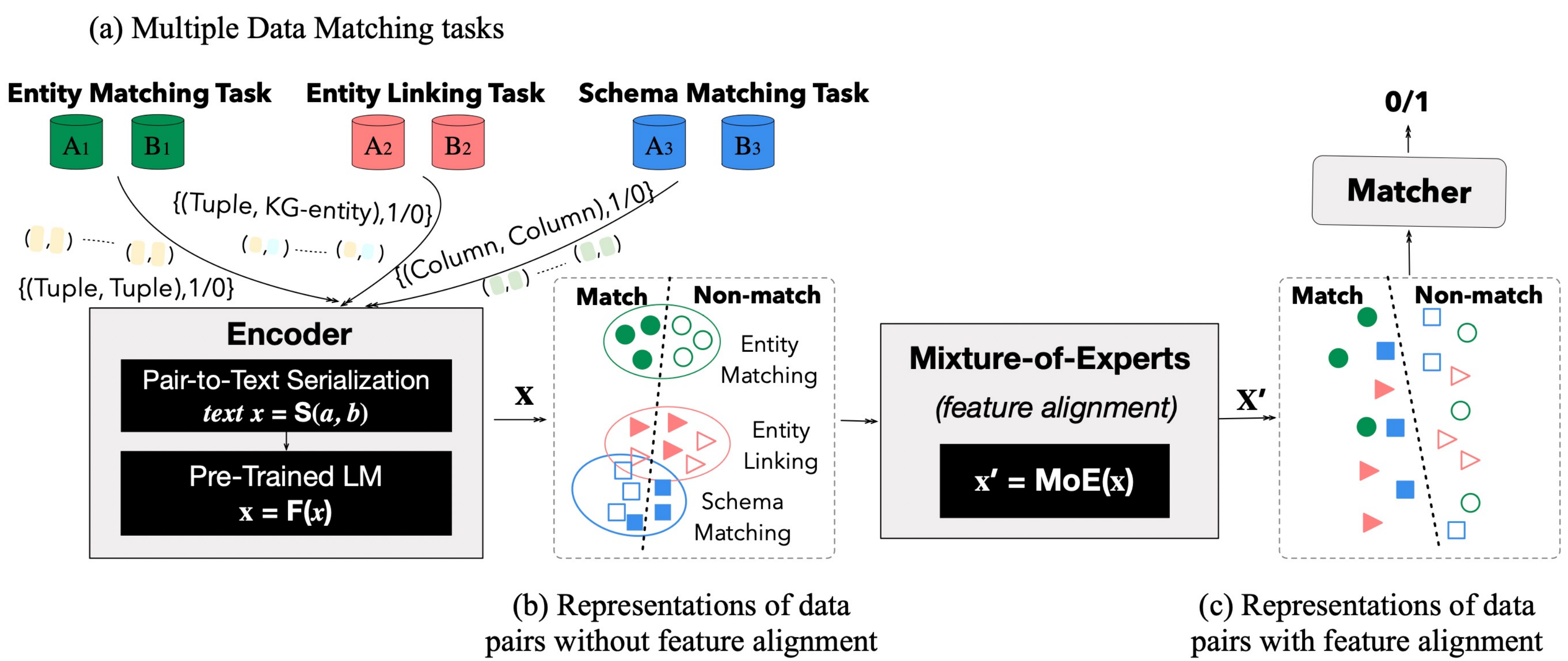
Can we build <u>a unified model</u> that learns from multiple tasks/datasets?

Unicorn: A Unified Model for Data Matching

- Task unification: A unified model to serve a variety of data matching tasks
- Multi-task learning: Enabling knowledge sharing across multiple data matching, which may even outperform specific models
- Zero-shot prediction: Making predictions for a new task or a new dataset with zero labeled matching/non-matching pairs
- Building such a unified model is hard
  - **Heterogeneous formats**: Data elements have different data formats
  - **Unique matching semantics**: Tasks have different data matching semantics



A General Framework of Unicorn



- **Encoder** converts any pair of elements with heterogeneous formats into a learned representation **x** based on Pair-to-Text Serialization and Pre-trained Language Model
- **Mixture-of-Experts (MoE)** layer enhances the representation **x** into a better representation **x'** with feature alignment
- **Matcher** predicts either 1/0 (match/non-match) by taking the above representation as input

Experiment

Table 4: The overall Result for Unified Prediction. `Unicorn` w/o MoE is a variant of `Unicorn` that has no MoE layer. `Unicorn` is our proposed framework with `Encoder`, MoE and `Matcher`. `Unicorn` ++ is improved with MoE optimization for Expert Routing.

Task Type	Task	Metric	Unicorn w/o MoE	Unicorn	Unicorn ++	Previous SOTA (Paper)
Entity Matching	Walmart-Amazon	F1	85.12	86.89	**86.93**	86.76 (Ditto [25])
	DBLP-Scholar	F1	95.38	95.64	**96.22**	95.6 (Ditto [25])
	Fodors-Zagats	F1	97.78	**100**	97.67	**100** (Ditto [25])
	iTunes-Amazon	F1	94.74	96.43	**98.18**	97.06 (Ditto [25])
	Beer	F1	90.32	90.32	87.5	**94.37** (Ditto [25])
Column Type Annotation	Efthymiou	Acc.	98.08	98.42	**98.44**	90.4 (TURL [7])
	T2D	Acc.	98.81	99.14	**99.21**	96.6 (HNN+P2Vec [4])
	Limaye	Acc.	96.11	96.75	**97.32**	96.8 (HNN+P2Vec [4])
Entity Linking	T2D	F1	79.96	91.96	**92.25**	85 (Hybrid I [16])
	Limaye	F1	83.12	86.78	**87.9**	82 (Hybrid II [16])
String Matching	Address	F1	97.81	98.68	99.47	**99.91** (Falcon [35])
	Names	F1	86.12	91.19	**96.8**	95.72 (Falcon [35])
	Researchers	F1	96.59	97.66	**97.93**	97.81 (Falcon [35])
	Product	F1	84.61	82.9	**86.06**	67.18 (Falcon [35])
	Citation	F1	96.34	96.27	**96.64**	90.98 (Falcon [35])
Schema Matching	FabricatedDatasets	Recall	81.19	**89.6**	89.35	81 (Valentine [22])
	DeepMDatasets	Recall	66.67	96.3	96.3	**100** (Valentine [22])
Ontology Matching	Cornell-Washington	Acc.	90.64	**92.34**	90.21	80 (GLUE [11])
Entity Alignment	SRPRS: DBP-YG	Hits@1	99.46	99.67	99.49	**100** (BERT-INT [44])
	SRPRS: DBP-WD	Hits@1	97.11	97.22	97.28	**99.6** (BERT-INT [44])
AVG			90.8	94.21	**94.56**	91.84
Model Size			139M	147M	147M	996M

Outline

- ☐ Overview
  - Motivation
  - Basic Concepts of PLMs
  - Word Embeddings for Data Preparation
  - Contextual Embeddings for Data Preparation
  - Domain Adaptation
  - Unified Data Matching

- ☐ **Challenges and Open Problems**

Open Problems of PLMs

- **Automatic domain knowledge injection**
  - *Can we* ***automatically identify*** *and* ***collect domain knowledge*** *in the wild?*

- **Data cleaning**
  - *Can the* ***contextual embeddings*** *generated by PLMs* ***benefit*** *various* ***data cleaning tasks****?*

- **Domain-adaptive data augmentation**
  - *Can we* ***synthesize labeled data*** *by considering the* ***domain adaption problem****?*

Thanks

DADER Design Space

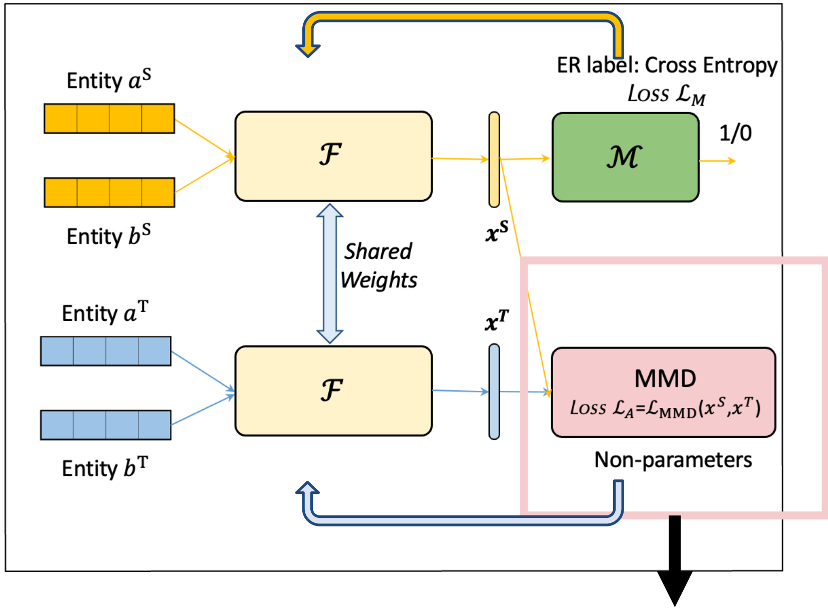
- Feature Extractor: RNN, LMs
- Matcher: MLP
- Feature Aligner: Discrepancy-based, Adversarial-based, Reconstruction-based

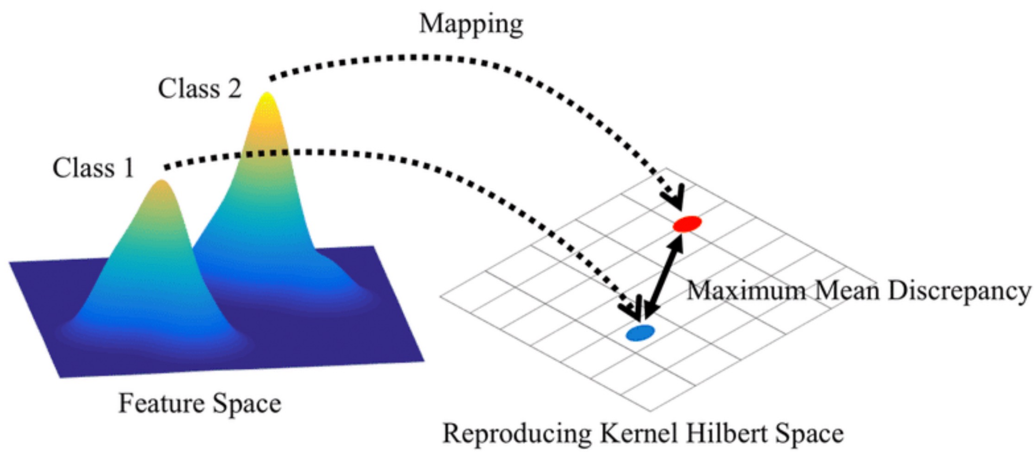
Modules	Categorization	
Feature Extractor ($\mathcal{F}$)	(I) Recurrent neural network (RNN) (II) Pre-trained language models (LMs)	
Matcher ($\mathcal{M}$)	Multi-layer Perceptron (MLP)	
Feature Aligner ($\mathcal{A}$)	(1) Discrepancy-based	(a) MMD (b) K-order
	(2) Adversarial-based	(c) GRL (d) InvGAN (e) InvGAN+KD
	(3) Reconstruction-based	(f) ED

Representative Method: MMD (Discrepancy-based)

➢ Feature Aligner is a function to measure **maximum mean discrepancy**.

During training, the Maximum Mean Discrepancy of source and target feature spaces is **computed** and **reduced**. The smaller the MMD, the more similar the distributions.



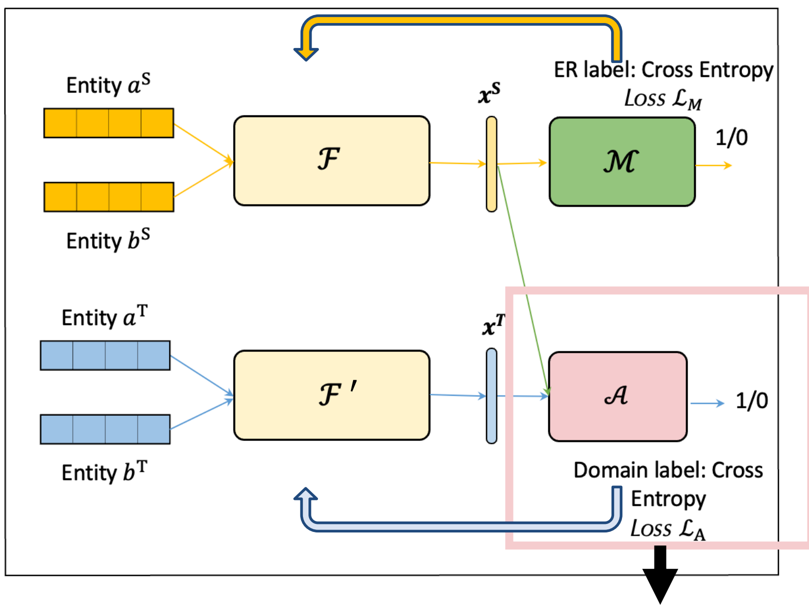


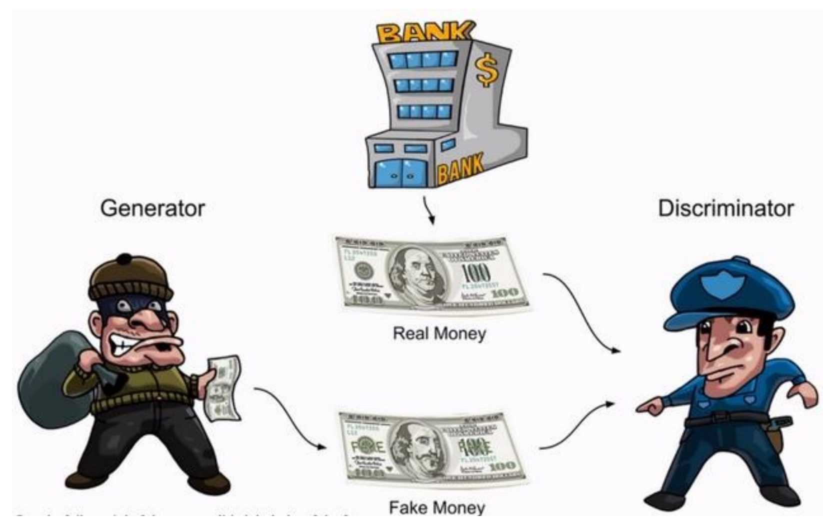
$$\mathcal{L}_{\mathrm{MMD}} = \sup_{\|\phi\|_H \le 1} \|E_{\mathbf{x}^{\mathrm{S}} \sim p_{\mathrm{S}}}[\phi(\mathbf{x}^{\mathrm{S}})] - E_{\mathbf{x}^{\mathrm{T}} \sim p_{\mathrm{T}}}[\phi(\mathbf{x}^{\mathrm{T}})]\|_H^2$$

Representative Method: InvGAN (Adversarial-based)

➢ Feature Aligner is a binary domain classifier to discriminate source/target dataset.

During training, the optimization objective of Feature Aligner is to **minimize the domain classification loss**, while Feature Extractor is to generate the **indistinguishable features** that confuse Feature Aligner.





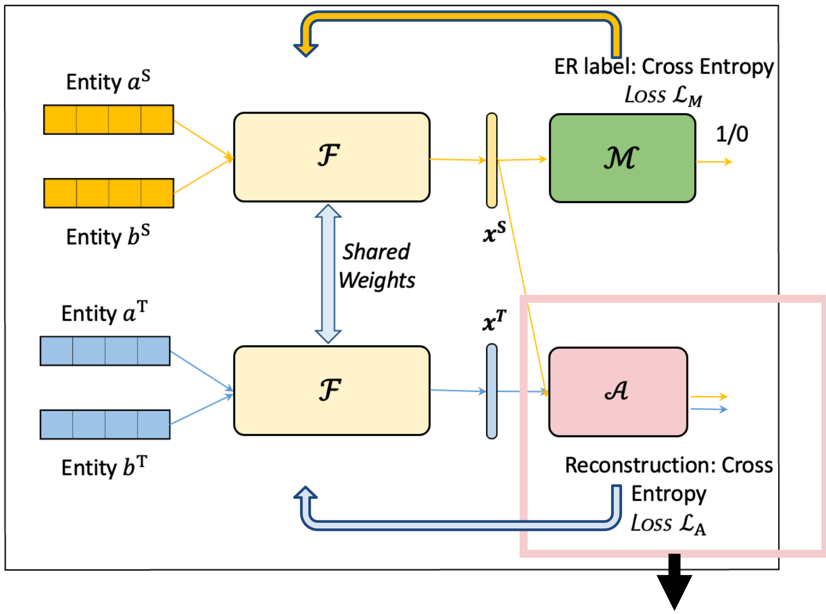
$$\min_{\mathcal{F},\mathcal{M}} \max_{\mathcal{A}} V(\mathcal{F}, \mathcal{M}, \mathcal{A}) = \mathcal{L}_M(\mathcal{F}, \mathcal{M}) + \beta \mathcal{L}_A(\mathcal{F}, \mathcal{A}),$$

$$\mathcal{L}_A = E_{x^{\mathrm{S}} \sim \mathcal{D}^{\mathrm{S}}} \log \mathcal{A}(\mathcal{F}(x^{\mathrm{S}})) + E_{x^{\mathrm{T}} \sim \mathcal{D}^{\mathrm{T}}} \log(1 - \mathcal{A}(\mathcal{F}(x^{\mathrm{T}}))),$$

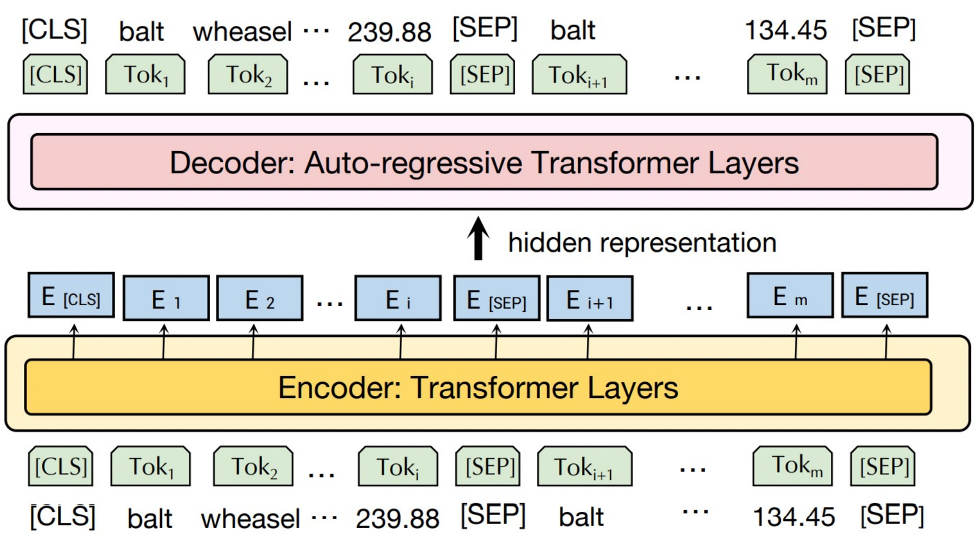
Representative Method: ED (Reconstruction-based)

➢ Feature Aligner is a decoder to reconstruct the initial data for source and target.

During training, the **auxiliary reconstruction task** can ensure the shared Feature Extractor (encoder) to extract important and shared information from both domains.



One example of Encoder-Decoder (ED) Architecture: Bart



$$\mathcal{L}_{\mathrm{REC}} = E_{x \sim \mathcal{D}^{\mathrm{S}} \cup \mathcal{D}^{\mathrm{T}}} \left[\mathcal{L}_{CE}(\mathcal{A}(\mathcal{F}(x)), x) \right]$$