EMBA: Entity Matching using Multi-Task Learning of BERT with Attention-over-Attention

Jing Zhang
Emory University
Atlanta, Georgia, USA
jing.zhang2@emory.edu

Huan Sun
The Ohio State University
Columbus, Ohio, USA
sun.397@osu.edu

Joyce C Ho
Emory University
Atlanta, Georgia, USA
joyce.c.ho@emory.edu

ABSTRACT

Entity matching is a crucial data integration process as it identifies whether two records refer to the same real-world object. Since shared identifiers are not always available, learning to match based on entity descriptions is an important task. While deep learning methods based on pre-trained transformers have been proposed to automate the entity matching process, these models only utilize the special token representation (i.e., [CLS]) to predict matches. However, this can ignore rich and nuanced contextual information in the descriptions, thereby yielding sub-optimal matching performance. We propose EMBA, a multi-task learning method with an attention-over-attention mechanism that leverages the individual token representations for the downstream tasks to better capture the information present in the descriptions of the two entities. Our evaluation across 7 entity matching benchmark datasets shows that EMBA achieves state-of-the-art performance, including up to an 8% improvement in F1 performance, over the existing dual-objective model. Our ablation study highlights the importance of using individual token representations. We also analyze the matching decision using both LIME explanations and attention score visualizations on a case study to illustrate the potential of EMBA.

1 INTRODUCTION

Entity matching (EM) is a data integration problem that identifies whether two data entries refer to the same real-world entity. It is an essential process for cleaning and fusing data across single or distributed data sources [11, 19, 23, 31, 39]. Matching entities accurately and quickly has enormous practical implications in commercial, scientific, and security applications [14] and is a longstanding problem in both data integration [9, 26] and data cleaning [1]. Figure 1a illustrates EM for different entries from two data sources. Unfortunately, the process of determining the pairs of matching entries can be time-consuming, especially in the presence of heterogeneous and large data sources. EM remains a challenging automation task because it requires a depth of language understanding and domain knowledge to match and distinguish entity information [25].

Existing EM approaches can be categorized by the level of comparisons made: attribute-centric, token-centric, and hybrid-centric [10]. An attribute-centric approach usually follows the alignment-comparison-summarization paradigm. This approach compares aligned attributes and aggregates the similarity vectors to determine the input for a binary classification system. Although these methods are generally successful, they may fail in real-world applications, when encountered with situations like schema heterogeneity, an extremely common occurrence. Accordingly, most recent research has been token-centric [24] or hybrid-centric [12], which incorporates token-level matching information into EM signals. Deep learning (DL) methods have become the de facto standard for tackling EM. By posing EM as semantic similarity matching, pre-trained natural language processing (NLP) models can serve as token-centric solutions to achieve impressive performance [7, 25, 27, 28, 32, 43]. These DL algorithms leverage popular transformer models such as BERT to automatically identify important entity description features using labeled examples without extensive engineering [37]. The pair-wise semantic similarity of two entity records, RECORD1 and RECORD2, is calculated by adapting BERT's input format (i.e., [CLS] RECORD1 [SEP] RECORD2 [SEP] as shown in Figure 1b) and using the [CLS] special token representation to predict the match.

While the vanilla transformer model has the potential to be useful for EM classification tasks, it suffers from several limitations. First, the self-attention mechanism was originally designed for capturing semantic interactions at the token level. Nevertheless, researchers usually construct entity descriptions by concatenating all attribute values, which introduces semantic discontinuity, deconcentrates the attention score, and impedes performance. One existing solution is serializing the entity with extra special tokens such as [COL] and [VAL] [25, 43]. Experimental results indicate that injecting special tokens for delimiting the attributes achieves better performance [31]. However simple fine-tuning will not fully utilize the data itself or may reach a sub-optimal point. Another limitation is that the masked language model (MLM) training objective optimizes token-level predictions but randomly masking some crucial information (i.e., the similar segments) can hamper the relatedness understanding for the entity pair. As such, introducing other sub-tasks can enrich the pragmatic knowledge encoded by BERT (as shown in [31]) and improve performance.

Multi-task learning techniques can obtain more general representations by complementing the main task objective with auxiliary training tasks [45]. One EM model has adopted the multi-task learning paradigm and proposed auxiliary multi-class classification problems to identify the individual entity identifier (or ID) from the descriptions to pair with the main EM classification problem [32]. While the model achieves state-of-the-art performance across some of the datasets, one major drawback is the failure to fully leverage the token representation power as only [CLS] special token is used for all 3 downstream tasks. Although the [CLS] token can be used to represent the meaning of the entire sentence, it may not be appropriate for all kinds of tasks (e.g., sequence tagging, or question answering). This ignores the rich semantic information from the individual tokens (e.g., the sub-word and character embeddings for the RECORD1) that potentially capture nuances in the entity description. Recent NLP work regarding sentence representation has highlighted the limitations of the special tokens [3, 20].

Title	PageCount	Rating	Description	Publisher
Betty Boothroyd: Autobiography	384	3.6	The enormous respect and affection ... views on the role of Parliament.	Random House UK
Autobiography of St. Teresa of Avila	352	4.11	In this landmark of Christian mysticism, ... of mystical literature.	Dover Publications
Benjamin Franklin: His Autobiography.	568	3.44	Based on Joseph Sabin's famed ... 25 cm	Gale Ecc

Title	Author	Salesrank	Price	Pages	PublicationDate
Betty Boothroyd Autobiography	Betty Boothroyd	-	-	434	10.28.02
The Life of St. Teresa of Jesus: Autobiography	Teresa of Jesus	-	-	-	01.13.14
The Autobiography of Benjamin Franklin: ... Diplomat	Benjamin Franklin	345,610	$9.95	182	12.28.10

(a) Examples of EM to determine the matching entries from two sources

Input Entity Pair		Serialized Entity Pair for BERT-based Models	Entity ID Prediction		Entity Matching		
Title + Description + Brand			JointBERT	EMBA	JointBERT	EMBA	Ground Truth
RECORD 1	buy online | samsung 850 evo 1tb ssd ... in india samsung 850 evo 1tb ssd mz-75e1t0bw	**[CLS]** RECORD 1 **[SEP]** RECORD 2 **[SEP]**	1696952	1696952	Match	Non-match	Non-match
RECORD 2	samsung 1tb 850 evo ...mz-n5e1t0bw | scan uk 1tb samsung 850 evo, m.2 (22x80) ssd, ...520mb/s, 97k/89k iops		1696952	899403			

(b) An example of the input to the BERT-based models and the prediction results from JointBERT and EMBA, where the entity IDs are computer cluster groups.

Figure 1: Illustrations of EM and results from the multi-task learning models

In this paper, we posit that individual token representations should be exploited in the multi-task formulation to improve the overall matching performance. We present EMBA, an entity matching multi-task learning model that uses the BERT individual tokens and attention-over-attention mechanism, to combine the binary EM and entity ID prediction. The main EM classification captures fine-grained relationships between the individual attribute values across the entity pair by utilizing an attention-over-attention (AOA) mechanism [5].

For the multi-class entity ID prediction module, EMBA learns the aggregation weights from the entity tokens, and provides flexibility to highlight task-specific aspects in the description. In this fashion, EMBA can identify the subword and character embeddings that are important for each task without requiring significant amounts of training data.

We compare our model against the existing multi-task learning EM model, JointBERT [32], a numerical-aware EM model, JointMatcher [43], and several transformer-based EM models [7, 25, 27, 28] on 7 EM benchmark datasets. Our results demonstrate that EMBA generally outperforms both models with multi-task objectives and those with single-task objectives with improvements ranging from 1-8%. We also conduct a detailed analysis of the attention weights to demonstrate the limitations of existing BERT-based EM models as the attention scores focus on a few words with contextual semantics that appear in both entity pairs. Our code is publicly available in GitHub.[1] In summary, our contributions are as follows:

- We propose EMBA, to utilize the individual BERT token representations for both the auxiliary entity ID prediction and main EM tasks.
- We align the individual token representations between the entity pairs using the AOA mechanism to capture cross-entity token interactions to better capture the similarity.
- We compare the performance of EMBA with existing state-of-the-art EM methods across 7 different datasets.
- We conduct a detailed ablation study to demonstrate the importance of using the individual token representations and our AOA module.
- We analyze matching decisions of EMBA and JointBert using LIME explanations [35] and attention visualizations to gain an understanding of the strengths of the token-based approach.

The paper is organized as follows. Section 2 introduces several previous related works about DL-based matchers. Section 3 illustrates our EMBA framework to tackle this matching problem. Section 4 describes the statistics of the datasets, the experiment design, and the results. It also contains the case study and ablation study to investigate the importance of the AOA mechanism and utilization of different token representations. Finally, in Section 5, we discuss conclusions from our results and potential future directions.

2 RELATED WORK

The three most common approaches to entity matching can be broadly categorized as rule-based [6], crowd-based [40], and machine-learning-based [15]. The traditional approach has been to handcraft string similarity metrics to produce similarity feature vectors and then utilize a classical off-the-shelf machine learning model such as SVM or random forest to classify them [22]. The two main drawbacks of this approach are the necessary manual tinkering and poor performance on dirty data. Recently, entity matching solutions used DL and achieved promising results [28, 47]. Since DL-based entity matchers offer state-of-the-art performance on the entity matching task, this paper focuses on these models.

2.1 Single-Task Models for EM

Most of the DL-based EM systems approach matching as a binary classification problem. DeepER [10] trains EM models based on the LSTM [18] neural network architecture with word embeddings such as GloVe [33]. DeepMatcher takes two data entries of the same quality as input and aligns their attributes before passing them on to the matching algorithm [28]. In contrast to

[1] https://github.com/JZCS2018/EMBA

DeepMatcher, the BERT [7] and RoBERTa-based [27] EM models eliminate both the attribute embedding and similarity representation components of the architecture in favor of a single pre-trained language model and offer a simpler design. As shown in Figure 1b, the pair-wise semantic similarity of two entity records, RECORD1 and RECORD2, is calculated by serializing both entries into a single input using the [CLS] and [SEP] tokens. The [CLS] is then used to determine whether the two entities match [2, 37, 46]. DITTO further builds on this idea by introducing two structural tags, [COL] and [VAL], to tackle the semantic discontinuity problem. Auto-EM [47] improves DL-based EM models by pre-training the model on an auxiliary task of entity type detection. JointMatcher [43] separates the records into two inputs to embed the features using a pre-trained language model and then introduces two encoders to identify similar and number-contained segments of the entity pair.

2.2 Multi-Task Learning Models for EM

The performance gains of multi-task learning in NLP illustrate the potential benefit of adopting this paradigm for the EM task. The idea is to integrate multiple tasks into the training of the DL architecture to yield improved, more general representations. JointBERT [32] introduces a dual-objective function to train the model and achieve a better performance compared with the single-task models. JointBERT uses the [CLS] token for both the main EM task and a multi-class classification problem to predict the entity identifier of each of the two entity descriptions. Although JointBERT can achieve better performance in some cases than the single-task, it uses the [CLS] token for its multi-class objective which can be suboptimal. Instead, we utilize the individual tokens to better differentiate the representation between the first entity description and the second entity description. Related to multi-task learning for EM is the multi-label setting where entities may have multiple intents which is necessary for user-personalization. To tackle the multiple intents entity resolution, FlexER [13] utilizes a graph neural network to learn intents through their relationships and improves upon the current state-of-the-art for universal entity resolution.

3 EMBA: MULTI-TASK LEARNING OF BERT WITH ATTENTION-OVER-ATTENTION

3.1 Problem Definition

Given two entity records, RECORD1 and RECORD2, their respective entity IDs, ID_{e_1} and ID_{e_2}, and their respective descriptions $D_{e_1} = \{D^1_{e_1}, D^2_{e_1}, \cdots, D^m_{e_1}\}$ and $D_{e_2} = \{D^1_{e_2}, D^2_{e_2}, \cdots, D^n_{e_2}\}$, where $D^1_{e_1}, D^2_{e_1} \cdots, D^m_{e_1}$ are the attributes, the multi-task learning paradigm seeks to learn (1) whether RECORD1 and RECORD2 refer to the same object (i.e., EM binary classification task) based on the descriptions (i.e., D_{e_1} and D_{e_2}) and (2) predict the entity ID (ID_{e_i}) based on the description D_{e_i}. The latter auxiliary tasks are known as multi-class classification problems where the entity ID serves as the class. The entity ID is user-specified and represents a grouping of objects within the dataset. Examples of entity ID prediction tasks include the publisher of the book (Figure 1a) or the computer cluster group (Figure 1b). It is important to note the two entities are not required to share the same schema. As shown in Figure 1a, RECORD1 contains the attributes title, page count, rating, description, and publisher whereas RECORD2 has the attributes title, author, sales rank, price, pages, and publication date.

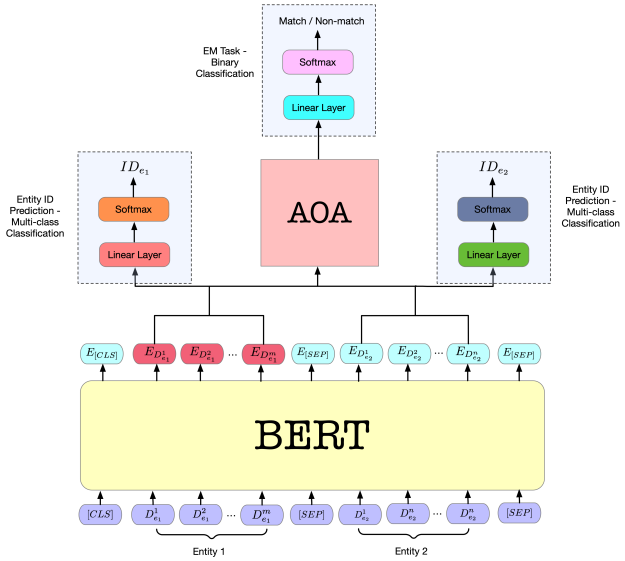

Figure 2: EMBA framework

3.2 BERT-based Token Embedding

EMBA follows the common BERT input format used for EM. The two entity descriptions are concatenated together as follows: [CLS] $\{D^1_{e_1}, D^2_{e_1}, \cdots, D^m_{e_1}\}$ [SEP] $\{D^1_{e_2}, D^2_{e_2}, \cdots, D^n_{e_2}\}$ [SEP].

In JointBERT, the pooled output representation of the [CLS] token is used to train both the EM binary classification task and the auxiliary entity ID prediction tasks. However, since the [CLS] token denotes the representation for the sequence pair, it is hard to untangle the representation of the two individual entities. Moreover, the [CLS] token may not properly capture the interactions between the pair of entities. While [CLS] is commonly used in NLP for many downstream classification tasks, recent work suggests that it may not always yield the best performance for all downstream tasks. Choi et al. illustrated the limitation of [CLS] for sentence embeddings [3]. Another recent work demonstrated that simply averaging the individual tokens provides better representations than the [CLS] token for semantic textual similarity tasks [20]. Furthermore, enforcing the same shared representation for multiple tasks can be beneficial with limited training data, but restricts the weights to be the same for all tasks. This scenario is suboptimal especially if the same token is used to predict two different entity identifiers, as the second entity description may not be fully reflected when using [CLS]. Therefore, EMBA uses the individual BERT token representations and learns task-specific weights.

3.3 Entity ID Prediction

A major motivation for moving away from the [CLS] token is recent NLP work that suggests that aggregating the token embeddings themselves may offer better sentence embeddings [3, 20]. One naïve approach is to use a different special token (e.g., [SEP] token) for the second entity ID prediction task, as the original [CLS] special token may not fully capture this entity description. However, as we will demonstrate in the ablation study, this offers marginal improvement as the [CLS] token remains a suboptimal representation for the first entity. Thus, for the entity ID prediction task, we propose the use of the token embeddings themselves as the input representation for the cross-entropy loss, as shown in Figure 2.

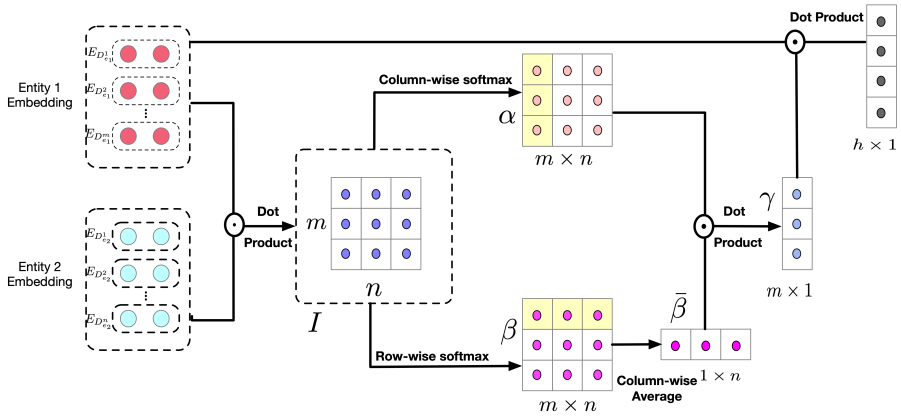

Figure 3: AOA module

Let E_{e_i} denote the output representations of the different entity tokens of the last encoder layer from BERT such that $E_{e_1} = \{E_{D^1_{e1}}, \cdots, E_{D^m_{e1}}\}$ and $E_{e_2} = \{E_{D^1_{e2}}, \cdots, E_{D^n_{e2}}\}$. EMBA uses the token embeddings from the entity description, E_{e_i}, directly for both auxiliary tasks. The token embeddings are passed to a linear layer that learns the task-specific weights to aggregate the representation and feeds it to the softmax layer. In this manner, each task can identify the subset of tokens that are indicative of the entity identifier. Since each entity description has different lengths, m and n for entities 1 and 2, respectively, the learned weights will be task-specific. We briefly note the definition of the entity ID task can impact the main EM task performance. An appropriate auxiliary task should be used where the classes are approximately balanced and there are sufficient samples for each class to train a model (i.e., predicting the primary key of an entity is unlikely to be a good auxiliary task).

3.4 AOA for EM

For the EM task, we again use the entity token representations, E_{e_1} and E_{e_2}. These representations are fed to an attention-over-attention (AOA) module to model the token-level interactions between these two pairs. AOA was first proposed for the question-answering task, as placing another attention over the primary attention can capture the importance of the original attention weights [5]. This is important as the dot product or the difference between the two entity representations can fail to capture the fine-grained relations between the individual attribute values. As such, the AOA module introduces mutual attention to simultaneously capture the relationships between the specific values of the first entity description to other values of the second entity description.

Our AOA module, illustrated in Figure 3, captures the correlations between the entity description using two mechanisms. Notice that $E_{e1} \in R^{m \times h}$ denotes the RECORD1 representation, where m is the first entity token length and h is the BERT token dimension. Similarly, $E_{e2} \in R^{n \times h}$ denotes the RECORD2 representation, where n is the second entity token length. The module first calculates the pair-wise interaction matrix $I = E_{e1} \cdot E^T_{e2}$, where the value of each entry represents the correlation of each token pair between RECORD1 and RECORD2. A column-wise softmax is applied to the interaction matrix I to obtain α, a probability distribution for each column, where each column represents the individual token-level level distribution for RECORD2 when considering the RECORD1. A row-wise softmax is applied to interaction matrix I to obtain β, the attention from the second entity description to the first entity description. Thus for the k^{th} token embedding from RECORD1 and the t^{th} token embedding from RECORD2, the associated attentions $\alpha(t)$ and $\beta(k)$ are:

$$\alpha(t) = softmax(I(1,t), I(2,t), \cdots, I(m,t)) \quad (1)$$

$$\beta(k) = softmax(I(k,1), I(k,2), \cdots, I(k,n)) \quad (2)$$

Then, the averaged second entity attention $\bar{\beta}$ is calculated using a column-wise averaging of β. Finally, the AOA $\gamma \in R^m$ is obtained as a weighted sum of the averaged second entity attention, $\bar{\beta}$, to α. By considering the contribution of each token explicitly, the AOA module learns the important weights for each token in the two different embeddings.

$$\bar{\beta} = \frac{1}{n} \sum_k = 1^n \beta(k)$$

$$\gamma = \alpha \cdot \bar{\beta}^T$$

The resulting AOA vector, γ is then multiplied with the entity 1 representation, E_{e1}, to yield a vector representation, $x \in R^{h \times 1}$ that is sent to the final classification layer which consists of a linear layer and a softmax layer to predict whether or not RECORD1 and RECORD2 refer to the same object.

Algorithm 1 Multi-task learning for EMBA

```
Initialize:
    Model parameters θ:
    a. Shared layer parameters by BERT;
    b. Task-specific layer parameters randomly;
Generate B by merging mini-batches for each dataset;
while epoch < Epoch_Num do
    Shuffle B;
    for Element in B do
        Compute loss L from Eq. (3);
        Compute gradient: ∇(θ);
        Update model: θ = θ − η∇(θ);
    end for
end while
```

3.5 Dual Objective Training

EMBA uses the binary cross-entropy loss (BCEL) for the main EM task and the cross-entropy loss (CEL) for the entity ID prediction tasks. Let $y_{em_i}, y_{e1_i}, y_{e2_i}$ denote the EM label, and the two entity

Table 1: Statistics about the datasets

Dataset	Size	# Pos. Pairs	# Neg. Pairs	LRID	# Classes	# Test Set
WDC computers	Xlarge	9690	58771	0.399	745	1100
	Large	6146	27213	0.189		
	Medium	1762	6332	0.135		
	Small	722	2112	0.149		
WDC cameras	Xlarge	7178	35099	0.764	562	1100
	Large	3843	16193	0.403		
	Medium	1108	4147	0.223		
	Small	486	1400	0.228		
WDC watches	Xlarge	9264	52305	0.518	615	1100
	Large	5163	21864	0.287		
	Medium	1418	4995	0.185		
	Small	580	1675	0.186		
WDC shoes	Xlarge	4141	38288	0.372	562	1100
	Large	3482	19507	0.261		
	Medium	1214	4591	0.208		
	Small	530	1533	0.194		
abt-buy	Default	822	6837	0.791	1013	1916
dblp-scholar	Default	4277	18688	4.548	52	5742
companies	Default	22560	67569	0.653	28200	22503
baby products	Default	108	292	1.008	132	40
bikes	Default	130	320	2.314	21	45
books	Default	92	305	1.865	2882	40

identifiers respectively, then we define the loss L as follows,

$$L_i = BCEL(y_{em_i}, \hat{y}_{em_i}) + CEL(y_{e1_i}, \hat{y}_{e1_i}) + CEL(y_{e2_i}, \hat{y}_{e2_i}) \quad (3)$$

where i stands for each pair. Algorithm 1 illustrates the process of applying multi-task learning to EMBA in which all layers in the model are refined. As a first step, similar to JointBERT, we initialize the parameters of the pre-trained BERT model and then randomly initialize the parameters of the task-specific layers, including EM classification, first entity ID prediction, and second entity ID prediction. During the training stage, both objectives are jointly optimized, so that the EM task will be improved by other two multi-class classification tasks training simultaneously.

4 EXPERIMENTS

We designed the experiments to answer three key questions: (1) How *accurate* is EMBA in automating the entity matching? (2) How *important* are the different components of EMBA? (3) What are the important words that are learned for the matching decisions?

4.1 Datasets

We compare the performance of EMBA with several existing baseline methods on 7 existing EM benchmark datasets. These datasets have already been split into non-overlapping training, validation, and test sets. The dataset statistics are provided in Table 1.

4.1.1 WDC datasets. The WDC Product Data Corpus for Large-scale Product Matching [34], was built by extracting product offers from Common Crawl. The WDC datasets serve as a popular benchmark and have been used for evaluation in DITTO, JointBERT, and the Semantic Web Challenge on Mining the Web of HTML-embedded Product Data at ISWC2020 [46]. WDC contains the titles, descriptions, and product identifiers from the e-shops' HTML pages. We utilize the same training, validation, and test configuration as JointBERT across four categories: computers, cameras, shoes, and watches. The training sets are available in four sizes, labeled small, medium, large and xlarge, ranging from 2,000 to 70,000 product offer pairs. All entities that are contained in the test sets are also represented with different entity descriptions in the training set.

For our experiments, we used the attributes brand, title, description, and specTableContent which are predominantly text and contain long sequences of words. The attribute values were gathered from the Web and may contain noise as a result of extraction errors. We limit the number of words used for each attribute to the 512-token maximum length limit for BERT-based transformer models. We predict the encoded product IDs, such as GTIN or MPN numbers, using both entity descriptions in a pair for the task of entity ID prediction.

4.1.2 abt-buy, dblp-scholar, companies datasets. Each dataset represents a match between two mostly deduplicated datasets from products (abt-buy), scientific texts (dblp-scholar), and companies. We use the same preprocessed splits as in JointBERT and DeepMatcher. Unlike the WDC datasets, these 3 EM datasets have a limited number of entity descriptions for the described entities (e.g., < 5 number of samples for many of the classes of the auxiliary task). The datasets only include matching labels for each pair of entities. We leverage these labels along with the transitive relationships that emerge from the pairs marked as matches to allocate unique identifiers to the descriptions of all the entities in each pair. For illustration purposes, if (A, B) and (B, C) are matches, then the group will include A, B, C. A unique cluster identifier is assigned for each group. As dblp-scholar includes specific information on venue and year, we use these attributes as the target for predicting entity IDs.

4.1.3 Magellan datasets. We use 3 more EM datasets, baby products, bikes, and books, from [22]. Baby products contain the baby products from Babies 'R' Us and Buy Buy Baby website and apply the same schema to the two tables title, ext_id, SKU, colors, and category. Bikes contain bike re-sale information from India's leading sources Bikedekho and Bikewale. The common schema contains color, bike_name, price, and km_driven. Books contain the book information from Goodreads and Barnes & Noble. The common schema of these two tables contains the page count, title, publisher, ISBN13, and format. Our dataset configuration excludes the ISBN13 attribute since it can be considered as the unique ID of each entity. The entity ID label is assigned for each of the datasets as category, brand, and publisher, respectively.

4.1.4 Likelihood ratio imbalance degree. Standard learning algorithms often assume relatively balanced class distributions. However, imbalanced data with unequal class distributions, can pose significant practical costs when the minority classes are incorrectly classified [48]. The class-imbalance extent is commonly measured using the imbalance ratio [16]. Unfortunately, the imbalance ratio is unable to capture detailed information for multi-class data as it relies only on the largest majority class and the smallest minority class for calculation, neglecting the nuances present in the remaining class distribution. The imbalance degree has been proposed to address the limitations for multi-class data [30]. However, improper use of distance metrics in these calculations can have harmful effects on the results.

Likelihood ratio imbalance degree (LRID) was proposed in [48] to measure imbalanced data:

$$LRID = -2 \sum_{c=1}^{C} n_c \ln(\frac{N}{Cn_c})$$

where C is the number of classes, n_c is the observations of class c, and N is the data size. A balanced multi-class dataset has an LRID of 0 and larger values of LRID denote higher degrees of imbalance. In Table 1, we present the LRID for the entity ID prediction tasks.

From the statistics, we observe that WDC datasets are almost balanced for different classes and dblp-scholar dataset is the most imbalanced.

4.2 Baseline Models

EMBA is evaluated against six baseline models and 3 pre-trained embedding variants. This section summarizes each of the models, along with their specific training settings. For the models using BERT (e.g., EMBA, DITTO, JointBERT), we generally use the pre-trained BERT-base model which consists of 12 layers and 768 dimensions. Any deviation from BERT-base is specified in the model description.

- **DeepMatcher** [28]: An EM solution that customizes the Recurrent Neural Network (RNN) architecture to aggregate the attribute values and then compares the aggregated representations of attribute values. It fixes the batch size at 16 and sets the positive-negative ratio, which controls the class weighting, to the actual distribution of each training set. It keeps the default values for all other hyper-parameters and uses fastText embeddings pre-trained on the English Wikipedia as input.
- **BERT-based Models**: Both uncased BERT and RoBERTa models are presented as in [32]. The attributes of each entity description are concatenated into a single string with any further preprocessing omitted and left to the tokenizer of the respective models. Both BERT and RoBERTa models use the full input length of 512 tokens.
- **DITTO** [25]: A state-of-the-art EM model that cast the problem as a sequence-pair classification and fine-tunes RoBERTa, a pre-trained Transformer-based language model [27]. We report the results from [32] which injected domain knowledge via the offered spans for the product or general domain according to the datasets. To make it comparable with JointBERT, the authors use the pre-trained BERT model rather than RoBERTa and set the batch size to 8 due to memory constraints with warmup.
- **JointMatcher** [43]: It is a novel EM method that forces the transformer model to learn the contextual information from the textual records. It contains a relevance-aware encoder and the numerically-aware encoder to pay more attention to similar segments and segments with numbers, respectively. It does not inject any domain knowledge when small or medium size training sets are used. Since its implementation is not accessible publicly, we summarize the results on the WDC datasets.
- **JointBERT** [32]: A dual-objective training method for BERT that combines binary matching and multi-class classification. The model uses the [CLS] token to predict the entity identifier based on each entity description in a training pair in addition to the matching decision. It achieved state-of-the-art results on the WDC datasets in large and xlarge settings.
- **EMBA (FT)**: fastText [21] was developed by Facebook's AI Research lab as an efficient tool for learning word embeddings and conducting text classification. By leveraging subword details, fastText can create precise word vectors for infrequently occurring or even unknown words. We pre-trained a fastText model using all of the 7 EM datasets. The BERT-based embedding was then replaced with our pre-trained fastText model.
- **EMBA (SB)**: There are several different embedding dimensions and layers offered by the BERT framework. BERT-small is a more compact version with 4 layers and 512 dimensions. This scaled-down version demands less computational power than the base and allows it to operate effectively on less robust hardware. We replaced the existing BERT-based component (BERT-base) with a pre-trained BERT-small model.
- **EMBA (DB)**: Similar to the EMBA (SB) variant, we investigate the use of distilBERT [36] instead of the pre-trained BERT-base model. distilBERT reduced the model by 40% using knowledge distillation and contains 6 layers and has an embedding size of 768 dimensions.

We train all models on a single NVIDIA Tesla V100 GPU with 16GB VRAM. The attributes of each entity description are concatenated into a single string. Any further preprocessing is omitted and left to the tokenizer of the respective models. EMBA and JointBERT use the full input length of 512 tokens. We fix the batch size at 32 and use the Adam optimizer to train the models for 50 epochs using a linearly decaying learning rate with one epoch warmup. A learning rate sweep is done over the range [1e-5, 3e-5, 5e-5, 8e-5. 1e-4]. We also apply the early stopping strategy if the model performance on the validation set does not increase over 10 consecutive epochs.

For all 7 datasets, EMBA and JointBERT are trained 5 times and we report the average performance with its standard deviation. For WDC, abt-buy, dblp-scholar, and companies, we present the best result for the other 5 models from either the JointMatcher or JointBERT paper [32, 43]. For the remaining 3 Magellan EM datasets, we report the average across the 5 trails for all but JointBERT and EMBA to maintain consistency.

4.3 Predictive Performance

4.3.1 EM Task. Table 2 summarizes the F1 results for the main EM binary task across all models and datasets. EMBA generally achieves the best performance on the computers, cameras, watches, and shoes categories for WDC. The lone exceptions are for the small training size setting where JointMatcher and RoBERTa achieve a higher F1 score. Our model also offers a performance improvement over the single-objective models such as BERT and RoBERTa by 1-11% and DITTO by 1-8% in the medium to xlarge settings.

For the other 6 smaller benchmark datasets, EMBA achieves the best performance on the companies and baby products datasets and the second-highest performance on the abt-buy dataset. RoBERTa offers the best F1 performance for abt-buy, dblp-scholar, and bikes, and second-best for baby products and books. DITTO achieves the best on books and second-best on bikes. As can be seen from Table 1, dblp-scholar, bikes, and books have a higher LRID (> 1.5) and limited number of entity descriptions for each entity ID class. This suggests that improperly designed auxiliary tasks can hinder the performance of the main EM task.

Among the variants of EMBA (e.g., FT, SB, DB), we observed variability in the performance. EMBA (SB), where the number of trainable parameters is only a quarter of EMBA, offers comparable performance and in some cases outperforms the original (BERT-case) version for some small datasets. This suggests that employing a simpler embedding model (with fewer layers and dimensions) can potentially lead to improved performance on datasets with limited training data. However, we note that this is

Table 2: Comparison of F1 performance on the EM task for the different datasets. The best performance is bolded and the second best performance underlined. Statistical significance analysis of the F1 performance between EMBA and JointBERT. The mean and standard deviation (in parenthesis) are shown, as well as the result of the t-test. * denotes if $p < 0.05$, ** if $p < 0.01$, * if $p < 0.001$, **** if $p < 0.0001$, and ns if $p \geq 0.05$.**

Dataset	Size	JointBERT	EMBA	EMBA (FT)	EMBA (SB)	EMBA (DB)	DeepMatcher	BERT	RoBERTa	DITTO	JointMatcher
WDC computers	xlarge	95.88(±0.96)	**98.44**(±0.82)**	87.21	96.61	67.91	88.95	94.57	94.73	96.53	95.73
	large	94.16(±1.49)	**97.73**(±0.37)**	84.46	94.11	71.11	84.32	92.11	94.68	93.81	94.03
	medium	86.00(±0.98)	**93.03**(±0.27)****	71.78	90.23	58.88	69.85	89.31	91.90	88.97	90.10
	small	75.66(±0.97)	81.89(±2.06)***	65.91	80.47	51.44	61.22	80.46	86.37	81.52	**86.95**
WDC Cameras	xlarge	95.31(±2.00)	**99.16**(±0.47)**	79.47	95.78	67.83	84.88	91.42	94.39	94.74	93.57
	large	93.02(±0.91)	**97.84**(±0.01)***	76.15	93.52	65.89	82.16	91.02	93.91	94.41	92.00
	medium	84.40(±1.94)	**91.90**(±0.79)***	68.41	89.95	59.18	69.34	87.02	90.20	87.97	89.26
	small	76.88(±0.86)	80.69(±0.81)****	60.26	78.74	48.70	59.65	77.47	**85.74**	78.67	84.15
WDC watches	xlarge	96.23(±1.38)	**99.11**(±0.15)**	82.15	97.19	68.22	88.34	95.76	94.87	97.05	96.61
	large	95.59(±2.08)	**98.97**(±0.30)*	80.32	95.87	63.12	86.03	95.23	93.93	97.17	95.89
	medium	84.72(±1.96)	**92.63**(±1.87)****	62.34	90.11	47.51	67.92	89.00	92.28	89.16	93.18
	small	72.79(±2.72)	83.28(±1.33)***	56.68	81.99	42.84	54.97	78.73	87.16	81.32	**91.31**
WDC shoes	xlarge	93.75(±3.48)	**98.47**(±0.49)*	80.23	96.63	61.60	86.74	87.44	88.88	93.28	90.22
	large	90.78(±3.14)	**97.16**(±0.96)**	78.21	95.13	62.18	83.17	87.37	86.60	90.07	89.01
	medium	77.48(±2.08)	**88.47**(±0.32)***	69.29	83.55	55.58	74.40	79.82	81.12	83.20	85.63
	small	67.42(±2.39)	73.42(±2.68)**	65.84	75.47	54.08	64.71	74.49	**80.29**	75.13	78.42
abt-buy	default	82.13(±1.11)	84.81(±1.37)**	63.12	79.36	62.29	62.80	84.64	**91.05**	82.11	-
dblp-scholar	default	93.25(±1.73)	94.71(±0.23)ns	87.48	93.28	58.17	94.70	95.27	**95.29**	94.47	-
companies	default	90.98(±0.70)	**92.74**(±0.39)**	81.25	91.17	73.20	92.70	91.70	91.81	90.68	-
baby products	default	73.00(±1.13)	74.21(±1.70)ns	68.14	**75.19**	61.07	70.12	72.25	73.49	72.12	-
bikes	default	67.18(±1.40)	71.76(±0.73)***	73.52	73.26	59.23	72.00	76.17	**77.21**	77.03	-
books	default	68.21(±2.35)	73.47(±0.93)**	72.19	75.25	61.39	74.00	74.42	75.17	**76.81**	-

not always the case as EMBA (DB) and EMBA (FT) generally experience a performance drop. Surprisingly, distilBERT, which has fewer layers but the same dimension as BERT-case, performs even worse than fastText. The performance variations among these three BERT variants and fastText highlight the impact of the underlying language model.

4.3.2 Comparison with JointBERT. Across the 7 datasets, EMBA always offers a better performance than JointBERT. The performance improvement ranges from 1-8%. We briefly note that since we evaluate JointBERT as the average of 5 runs, the F1 score is lower for the WDC datasets, abt-buy, dblp-scholar, and companies than those reported in [32]. However, EMBA still provides better performance compared to the scores in the original paper. This illustrates that using the [CLS] token for all three tasks is suboptimal, as it restricts the representation power of the embedding. By adopting the token-based representation for all three tasks, EMBA has more flexibility to learn a better overall representation without constraining the [CLS] token to generalize to all three tasks.

We also conduct an analysis to determine whether EMBA provides a statistically significant improvement over JointBERT and assess the stability of the two models. The null hypothesis (H_0) and alternative (H_a) hypotheses are as follows:

$$H_0 : \mu_{\text{EMBA}} \leq \mu_{JointBERT}$$
$$H_a : \mu_{\text{EMBA}} > \mu_{JointBERT}$$

Table 2 presents the results of the one-tailed t-tests between EMBA and JointBERT using the * symbol next to EMBA. We notice that for all but *dblp-scholar* and *baby-products*, we can reject the null hypothesis suggesting that EMBA provides statistically significant improvements over JointBERT. For both datasets, the null hypothesis cannot be rejected as the largest F1 score from JointBERT is greater than the smallest of five F1 scores from EMBA.

Table 2 also illustrates the stability of EMBA. As the WDC training size increases, we can observe that there is less variation (i.e., standard deviation shown in parenthesis is smaller) in the performance of our model. However, this trend is not necessarily observed in JointBERT as can be seen by the standard deviation for the camera category and the xlarge training size setting. Moreover, EMBA consistently has smaller standard deviations than JointBERT, which suggests a more stable performance.

4.3.3 Entity ID Tasks. We further explore the performance of JointBERT and EMBA on the auxiliary entity ID prediction tasks. Table 3 shows the results rounded to two decimal points for accuracy and micro F1. Acc1 and Acc2 represent the accuracy for the first and second entity ID prediction, respectively. EMBA and EMBA (SB) outperform JointBERT over all datasets. And we notice EMBA (SB) also outperforms EMBA on small datasets. Although for the companies dataset, JointBERT seemingly does not identify anything, this due to rounding. JointBERT correctly matches a small number, with scores around ~ 0.002. When focusing on the smaller benchmark datasets, EMBA improves the results at most 67% compared with JointBERT. The results demonstrates the benefit of allowing flexibility to learn the task-specific weights from the individual tokens instead of the single [CLS] token for the two entity ID prediction tasks.

Table 3 also illustrates the potential limitation of multi-task learning when there are insufficient samples to appropriately train the auxiliary prediction models. The entity ID F1 performance on the small sizes for the WDC datasets, abt-buy, and bikes are considerably lower than the other values and in these situations, the single-task models perform better on the main EM task. This is exacerbated as the two auxiliary tasks together provide more weight on the dual objective than the main EM task. The lone exception occurs for companies, where the low entity ID prediction model still improves the overall EM F1 score. This is potentially because companies dataset is larger and more

Table 3: Comparison of accuracy (Acc) and micro F1 on the entity ID prediction tasks for the different datasets. The best performance is bolded.

Dataset	Size	JointBERT			EMBA			EMBA (SB)			EMBA (DB)			EMBA (FT)		
		Acc1	Acc2	F1	Acc1	Acc2	F1	Acc1	Acc2	F1	Acc1	Acc2	F1	Acc1	Acc2	F1
WDC computers	xlarge	95.09	90.73	91.82	**98.73**	98.82	**98.82**	**98.73**	**98.90**	98.48	98.71	96.45	95.18	97.31	96.56	96.12
	large	94.18	88.27	87.01	**98.91**	**99.09**	**98.66**	98.73	98.91	98.48	96.91	94.55	92.25	96.28	93.18	90.01
	medium	51.36	44.09	52.31	**97.18**	**96.55**	**96.87**	96.36	94.55	93.40	84.18	80.09	73.13	86.32	80.14	83.02
	small	7.45	5.27	3.16	54.09	43.00	44.11	**62.73**	**59.82**	**62.11**	50.27	40.82	42.53	51.90	50.41	49.27
WDC cameras	xlarge	95.45	91.45	92.58	**100.00**	**98.18**	**99.67**	99.82	**98.18**	99.33	99.36	95.00	97.28	98.78	95.19	96.34
	large	92.09	87.55	87.13	**99.91**	**97.73**	**99.33**	99.45	97.09	98.49	96.82	92.64	93.81	84.23	86.65	82.11
	medium	49.27	44.09	53.45	**96.55**	**94.00**	**96.10**	90.10	87.82	88.22	78.82	75.45	74.58	79.54	76.12	76.30
	small	3.91	7.55	17.56	71.27	60.00	62.82	**71.82**	**62.00**	**68.11**	57.45	43.27	48.44	55.41	57.12	45.56
WDC watches	xlarge	96.00	86.64	89.53	**100.00**	**98.27**	**99.33**	99.91	97.45	99.00	95.91	91.91	91.26	89.16	87.90	89.23
	large	92.64	85.82	87.33	**99.91**	**98.64**	**99.34**	**100.00**	97.91	99.01	94.00	88.45	85.97	88.96	84.17	86.68
	medium	56.09	46.64	53.94	92.91	**91.64**	88.16	**93.73**	91.18	**88.37**	77.27	70.91	59.63	75.63	73.78	71.26
	small	4.09	4.18	2.30	21.18	18.27	19.20	**66.82**	**57.18**	**61.91**	40.64	37.09	30.00	35.96	30.01	32.12
WDC shoes	xlarge	97.73	90.35	94.59	99.91	**98.09**	**99.17**	**100.00**	97.73	98.67	94.54	90.81	90.51	89.71	84.22	85.46
	large	95.36	88.44	89.43	**99.91**	**98.18**	**98.52**	99.72	97.36	98.38	92.99	89.63	89.57	87.28	84.13	84.02
	medium	40.67	24.84	46.74	**95.45**	**90.63**	**91.22**	94.18	89.17	89.07	44.68	45.95	47.13	58.73	65.26	64.19
	small	0.18	0.55	0.66	11.92	12.10	9.83	**73.34**	**55.69**	**63.33**	9.10	6.10	4.76	9.87	5.62	6.13
abt-buy	default	10.44	14.70	15.01	84.19	79.28	57.53	**95.41**	**95.20**	**77.96**	89.66	85.65	67.02	86.26	82.41	85.63
dblp-scholar	default	10.14	2.61	10.80	**94.81**	**53.13**	**73.68**	92.13	47.81	70.26	80.13	76.19	79.28	82.12	79.28	81.90
companies	default	0.00	0.00	0.00	12.13	19.28	17.97	**15.27**	**26.12**	**25.69**	9.28	11.21	11.33	11.23	15.27	15.32
baby products	default	90.12	92.89	92.56	98.26	96.11	96.93	**98.74**	**96.92**	**97.18**	89.56	87.45	88.64	91.14	90.10	90.45
bikes	default	47.07	42.23	44.92	67.45	69.18	68.73	**72.13**	**75.98**	**76.56**	61.30	65.36	65.12	70.21	71.37	73.56
books	default	67.17	62.36	66.89	82.84	86.53	87.21	**84.98**	**88.24**	**89.15**	76.82	70.35	72.80	79.69	78.15	79.42

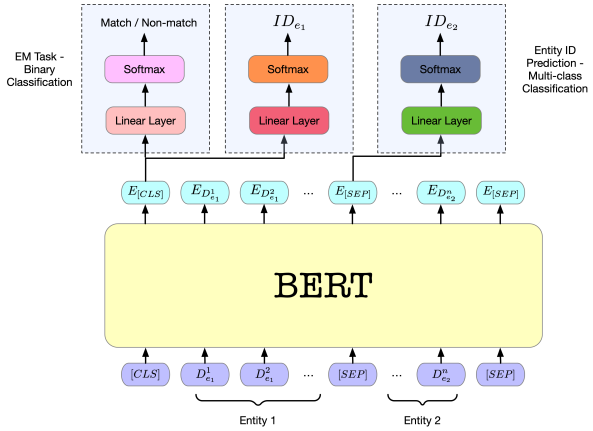

Figure 4: JointBERT-S where the [SEP] token used for the second entity ID prediction task and the [CLS] token is used for the binary classification and first entity ID prediction.

balanced than other small datasets, which can help mitigate a bad auxiliary task.

4.4 Ablation Study

To gain further insights of the various components in EMBA, we conducted an ablation study. In particular, we examined the effectiveness and contributions of the AOA module, and token representation strategy for the auxiliary (i.e., first and second entity ID prediction) and main (i.e., entity matching) tasks across the four benchmark datasets.

- *JointBERT with [SEP] token (**JointBERT-S**)*: This is the naïve extension of JointBERT to use a different special token, [SEP], for the second entity ID prediction task as shown in Figure 4. Note that the [CLS] token is used for the entity matching and first entity ID prediction task.
- *JointBERT with word-tokens representations (**JointBERT-T**)*: We utilize the average token representations for all the tasks. For the entity ID prediction task, the average of the token representations from the entity description is passed to a softmax layer. Similarly, for the entity matching task, we average the two entity token representation.
- *JointBERT with [CLS] token and word-tokens representations (**JointBERT-CT**)*: We utilize the word-token representations for the two auxiliary tasks (same average token representation as JointBERT-T) but keep the [CLS] special token for the entity matching task.
- EMBA *only with [CLS] token (**EMBA-CLS**)*: The [CLS] special token is used for the two auxiliary tasks but the AOA module is used for the binary matching problem.
- EMBA*with SurfCon [41] (**EMBA-SurfCon**)*: The SurfCon framework proposed an encoding component and a context-matching component to capture sequence-level and token-level similarity. We substitute the AOA module with the SurfCon framework while maintaining the same configuration as EMBA for all the other parts.

The results of the ablation study are summarized in Table 4. Unsurprisingly, EMBA outperforms the other models, suggesting that all the components are needed for better matching performance. We can observe that simply swapping the representation to the [SEP] token for the second entity ID prediction task (JointBERT-S) improves the performance and in some cases provides the second-best performance. This demonstrates that using the [CLS] token for all three tasks is suboptimal, as it restricts the representation power of the special token embedding to fit 3 tasks.

Table 4 also highlights the importance of using individual token representations. Even using a simple average of the tokens

Table 4: Comparison of F1 performance on EM for the ablation experiments. The best performance is bolded and the second best performance underlined.

Dataset	Size	JointBERT	JointBERT-S	JointBERT-T	JointBERT-CT	EMBA-CLS	EMBA-SurfCon	EMBA
WDC computers	xlarge	96.37	98.83	97.49	97.65	97.48	96.86	**99.03**
	large	94.81	97.83	96.68	97.50	95.52	97.33	**97.96**
	medium	86.55	92.33	89.86	90.65	89.48	89.34	**93.06**
	small	76.15	81.74	76.47	80.18	77.31	67.52	**83.15**
WDC cameras	xlarge	96.34	98.32	98.00	99.01	98.19	98.60	**99.33**
	large	93.55	97.66	95.44	97.04	96.03	97.34	**97.84**
	medium	85.36	91.13	86.46	88.44	86.11	84.07	**91.88**
	small	77.33	80.24	74.66	75.80	78.12	57.92	**80.98**
WDC watches	xlarge	96.99	98.32	98.35	98.84	98.01	97.79	**99.18**
	large	96.66	98.84	97.87	98.33	98.02	97.84	**99.05**
	medium	85.66	93.23	89.03	91.22	87.44	84.42	**93.8**
	small	74.16	83.77	75.10	79.65	79.37	57.38	**83.91**
WDC shoes	xlarge	95.49	98.67	97.81	97.99	96.99	97.46	**98.72**
	large	92.40	97.50	97.84	96.88	96.11	93.07	**97.83**
	medium	78.73	85.67	80.65	87.50	81.63	71.74	**88.65**
	small	68.84	73.73	68.89	69.94	71.64	57.20	**74.79**
abt-buy	default	82.76	85.17	81.35	81.72	83.29	79.86	**85.42**
dblp-scholar	default	94.12	94.58	94.40	93.17	94.13	94.01	**94.83**
companies	default	91.39	91.94	91.54	91.15	89.17	90.69	**92.73**
baby products	default	73.63	74.17	73.98	73.63	72.49	73.89	**75.12**
bikes	default	67.96	69.21	68.47	68.92	65.71	68.14	**72.23**
books	default	69.58	72.32	69.93	71.24	67.28	67.21	**74.07**

(JointBERT-T) does not hinder the performance and even in some cases provides a slight benefit, which is consistent with the findings for semantic textual similarity in [20]. This suggests that the [CLS] token may not provide the best entity representation. This phenomenon can also be observed when comparing results between EMBA and EMBA-CLS. Notably, the main difference is that [CLS] token is used in the latter model for the two auxiliary tasks. However, this change results in a significant drop in performance for EMBA-CLS, especially for the smaller training sizes.

To identify the impact of the AOA module, we first compare the results of JointBERT with EMBA-CLS. We observe that AOA alone is often insufficient without using the token representation for the auxiliary tasks. However, in conjunction with the token representation (i.e., EMBA), the AOA module can better tease out important attention weights as the embedding is fine-tuned to better reflect the task. The results also illustrate that the AOA module is necessary as swapping it out for the average or even the SurfCon framework does not yield better results than EMBA.

In addition, we compared the model performance on the other multi-class classification tasks (i.e., entity ID prediction). The accuracy and micro F1 for JointBERT-S, JointBERT-T, and JointBERT-CT are presented in Table 5. These are the models that exhibit superior performance. When using [SEP] token for second entity ID prediction (i.e., JointBERT-S), or averaging each entity token for 1st/2nd entity ID prediction respectively, F1 scores on small datasets are improved by 30%, while on large datasets they are improved by 20% compared with JointBERT. We also observe that a simple average of the tokens, JointBERT-T, often provides an improvement over JointBERT with the lone exceptions on baby products and books.

We do note that since the lengths of entity pairs are different, it is hard to simply batch the outputs from BERT. We apply the sample-wised computation to the AOA module, which will be slower than batched computation. Based on this, we also tried a simple padding strategy to enable batching of the outputs from BERT, which will expedite the computation of AOA module. However, traditional padding is applied before the model, so that it can learn the zero paddings to avoid the skewness. We experiment on small and xlarge datasets of WDC computers, and the F1 scores on small and xlarge datasets are 79.16 and 96.68, which is much lower than those in EMBA. It means the intermediate padding for the AOA will skew the representation for the downstream tasks.

4.5 Impact of Imbalanced Datasets

We explored the impact of class imbalance on the EM models. Based on Table 1, the WDC datasets do not exhibit a severe class imbalance in terms of the ratio between the positive and negative pairs. Thus, we created three variations of the WDC computers dataset by sampling the number of positive samples from 9690 to 6146, 1762, and 722 while leaving the negative samples unchanged for the xlarge dataset. In this manner, the overall dataset size is not significantly altered by reducing the positive samples as it is one of the largest samples. Table 6 presents the F1 scores for the 5 of the models. We omit JointMatcher, DeepMatcher, RoBERTa, and EMBA (DB) based on their performance in Table 2 for the WDC computers xlarge dataset. We observe that EMBA and EMBA (SB) did not experience large performance drops in comparison with the other models. However, there is still a performance drop as the class imbalance becomes more noticeable. This suggests that imbalanced datasets are not necessarily problematic but may

Table 5: Comparison of accuracy (Acc) and micro F1 on the entity ID prediction tasks for the different datasets. The dataset size order matches that of the others. The superior performance compared to EMBA is highlighted in bold, and it is underscored as the best among the three models presented.

Dataset	JointBERT-S			JointBERT-T			JointBERT-CT		
	Acc1	Acc2	F1	Acc1	Acc2	F1	Acc1	Acc2	F1
WDC computers	98.36	98.45	98.49	98.09	98.27	98.13	98.72	98.63	98.48
	98.09	96.27	96.27	96.81	94.91	93.74	98.36	99.09	98.47
	96.18	93.25	92.76	93.90	94.28	93.33	96.81	94.91	93.74
	21.54	21.18	30.84	11.36	15.91	15.09	48.36	37.00	44.12
WDC cameras	100.00	98.27	99.83	100.00	97.91	99.33	100.00	**98.36**	**100.00**
	99.63	97.73	99.16	99.91	97.45	99.33	**99.79**	**98.18**	99.08
	90.72	87.82	86.93	93.45	91.82	93.98	92.90	91.91	92.31
	62.45	55.27	56.08	10.27	12.00	8.59	59.00	49.36	47.62
WDC watches	99.09	98.09	98.51	98.17	96.98	96.53	99.12	**99.09**	98.73
	99.63	98.82	**99.50**	99.22	98.54	99.14	99.81	98.63	**99.61**
	94.18	**92.64**	**89.15**	95.54	91.45	88.26	**96.11**	**92.62**	**89.19**
	19.64	12.82	12.65	15.54	12.73	13.28	20.28	13.64	15.52
WDC shoes	99.78	97.82	99.00	99.81	98.09	99.00	**100.00**	97.91	99.17
	99.91	**98.36**	**98.68**	99.91	98.27	98.52	99.91	97.99	98.36
	95.18	**92.08**	91.20	93.54	89.35	89.97	92.72	90.71	90.88
	10.85	9.04	6.38	8.38	6.47	5.34	11.77	10.94	8.62
abt-buy	78.31	71.12	52.86	43.53	37.11	16.62	79.62	70.98	44.23
dblp-scholar	92.71	51.94	67.05	46.49	32.91	31.50	90.18	50.11	62.56
companies	0.47	0.22	0.63	0.14	0.57	0.35	0.15	1.31	0.82
baby products	92.45	94.51	94.87	90.19	92.36	92.18	93.12	92.87	90.26
bikes	52.63	55.96	50.28	49.87	52.56	49.12	50.50	52.62	18.79
books	69.24	65.69	69.58	62.78	63.90	60.26	69.17	64.56	60.23

Table 6: Results for unbalanced datasets experiments on F1 for the EM task. The number in parenthesis, (Δ), denotes the change in F1 when compared with the result on the WDC computers xlarge dataset.

Pos./Neg. Ratio	JointBERT	EMBA	EMBA (SB)	BERT	DITTO
0.104	92.23 (-3.65)	98.12 (-0.32)	95.39 (-1.22)	91.44 (-3.13)	94.71 (-1.82)
0.030	89.37 (-6.51)	96.56 (-1.88)	92.63 (-3.98)	87.10 (-7.47)	91.69 (-4.84)
0.012	86.12 (-9.76)	93.41 (-5.03)	91.87 (-4.74)	86.23 (-8.34)	90.15 (-6.38)

require other mechanisms to improve robustness for datasets with smaller training samples.

4.6 Computational Efficiency

Given the dependence on DL frameworks for the EM tasks, it is crucial to assess their computational efficiency given the increasing model complexities and parameter spaces. To gauge the computation requirements across various models, we assess their performance speeds during both the training and inference phases. Table 7 summarizes the computational speed of the EM models using the metric of entity pairs (or items) processed per second. EMBA (FT) emerges as the front-runner, managing 44 items per second in training and an impressive 121 items per second in inference. DITTO also performs well, particularly in the inference stage, processing 33 items per second, a boost

Table 7: Computational efficiency of the different EM models (pairs/second) for the two stages, training and inference.

Model	Training	Inference
JointBERT	10	20
EMBA	9	19
EMBA (FT)	44	121
EMBA (SB)	28	52
EMBA (DB)	16	30
BERT	10	24
RoBERTa	8	20
DITTO	12	33

attributed to its use of mixed precision optimization. The base version of EMBA demonstrates efficiency on par with models like JointBERT, BERT, and RoBERTa. Notably, EMBA (SB) offers better speed as only the fastText variant is faster, excels on smaller datasets, and still maintains comparable performance relative to the state-of-the-art EMBA. These results suggest that if memory and computational speed are important factors for deployment, EMBA (SB) is more suitable.

4.7 Case Study

To better understand the potential benefit of EMBA in terms of explaining the matching decision, we investigate the word and token importance between our model and JointBERT. We use an example where a non-match is classified incorrectly by JointBERT but correctly by EMBA to illustrate the differences. The entity descriptions for two entities are shown in Figure 5a. As can be seen, the brand names of these two entities are different and thus should not match. However, we can also observe that they share many similar attribute values such as 4gb, 50p, cf, CompactFlash, card, and retail.

4.7.1 LIME Explanations. We first analyze the word importance using the same methodology used for JointBERT [32]. In particular, we utilize the Mojito framework [8] which is based on the LIME algorithm [35] and has been used to explain deep matching decisions [32]. LIME perturbs all pairs of entity descriptions by randomly dropping words and then labels for all perturbed instances are queried from the model. A surrogate linear regression model is then trained using this set of instance/label pairs and serves as a local approximation for the original model. The resulting linear regression coefficients then provide the importance of the individual word in determining the matching decision.

Figure 5 illustrates the LIME explanations generated with Mojito for a matching decision by JointBERT (see Figure 5b) and by EMBA (see Figure 5c). Orange-colored words push the model toward a non-match whereas blue-colored words have the opposite effect (pushing toward a match). As can be seen from the figure, JointBERT considers the brand *transcend* as a match signal, while EMBA identifies the same attribute as a non-match. We can also observe that the non-match words identified from EMBA have a higher negative weight (darker orange color), whereas the match words identified by JointBERT display a higher positive weight (darker blue color). The figure highlight some of the benefits of using the individual token representations to make the entity prediction and matching decision, as too much similarity between the entity descriptions can drown out the non-match signal from a small but important subset of attribute values.

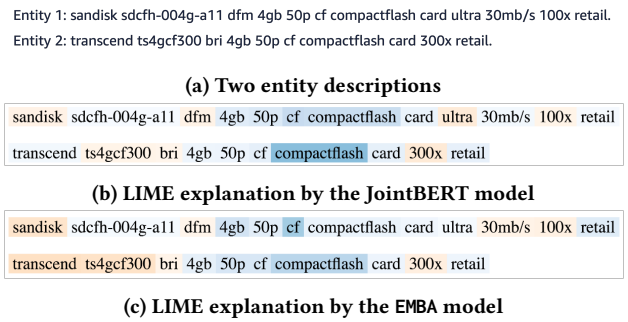

Figure 5: LIME explanations for a non-match classified incorrectly by the JointBERT and correctly by the EMBA.

4.7.2 Attention Visualizations. To demonstrate the intuitive benefits of token-level representation, we visualize the variation in the attention score of similar segments in the same entity pair using JointBERT and EMBA. Figure 6 illustrates the attention scores of each word in the entity description. We note that in some cases, the record pair is split into token sequences by the WordPiece tokenizer to deal with out-of-vocabulary words like "sdcfh-004g-a11". For a split-up word, we sum the attention scores over its tokens based on the multi-head attention in the last layer as suggested by [42]. It can be seen that in JointBERT, most of the attention scores focused on a few words with contextual semantics, such as "compactflash" and "sdcfh-004g-a11" in entity 1, and "compactflash" and "ts4gcf300" in entity 2. The high attention to "compactflash" in both entities lead JointBERT to incorrectly conclude that there is a match. The brand name "sandisk" in entity 1, and "transcend" in entity 2 did not obtain enough attention in JointBERT. Also, JointBERT gives low attention scores for several alignments on the parameters, such as "4gb 50p" and "300x", which could provide other evidence of a non-match. In contrast, EMBA enhanced the attention scores of the brand name, “transcend" and “sandisk". Moreover, both "sdcfh-004g-a11" and "ts4gcf300" have higher attention along with some of the other attributes. These higher weights help EMBA focus on the small subset of attributes to achieve the correct label for the entity pairs.

We hypothesize that one potential reason why the attention loses focus on some important words in JointBERT is that the [CLS] token denotes the representation for the sequence pair. As such, it is hard to untangle the representation of the two individual entities, and there are no strong signals to give feedback to optimize the model parameters. In contrast, EMBA feeds the token representations rather than special tokens to the tasks and obtains appropriate feedback from different tasks to optimize the attention weights. Therefore, the attention weights can better focus on the crucial tokens such as the brand names and model numbers and improve the results.

We also explored the case where EMBA incorrectly predicts a non-match but JointBERT correctly predicts a match. For example, consider the two entities, 1. *corsair cmso4gx3m1a1333c9 4gb ddr3 1333mhz sodimm unbuff cl9 for laptops laptops for $38.54.*; 2. *corsaer 4gb (1x4gb) ddr3 1333 mhz (pc3 10 666) laptop memory blank media - page 2 | all tech toys.* The golden standard indicates that these two are the same entity. In the datasets, if the entity pair is the same, their pre-defined entity IDs are also the same. When we analyze its entity ID prediction tasks, both of them belong to the same pre-defined entity ID, and JointBERT predicts them right, but the results of EMBA are different. We posit this is because we aggregate the word token of each entity, which can integrate noisy information especially when the entity contains a long description. This suggests that there are cases where aggregating over long token sequences can be harmful in which case the [CLS] special token offers a better representation.

5 CONCLUSION

Our paper highlights that using the [CLS] token for both the auxiliary entity ID prediction and main EM tasks is suboptimal, as it restricts the representation power of the embedding. Instead, we introduce EMBA, to fully utilize the BERT token representations for the multi-task formulation. We align the individual token representations between the pairs of entities using the AOA mechanism to capture cross-entity token interactions to better capture the semantic similarity. We also propose to learn the aggregation weights from the individual tokens for the entity ID prediction task. The experiments on 7 benchmark datasets demonstrate that EMBA can achieve state-of-the-art performance for EM. The results also demonstrate that our model provides a statistically significant performance improvement when compared with the other dual-objective model, JointBERT. The experiments highlight the importance of appropriate auxiliary tasks to achieve better performance. We explored a different auxiliary task for dblp-scholar (i.e., venue instead of venue and year for the ID prediction) which improved the performance.

Our detailed ablation study illustrates that both the individual token representation and the AOA module are necessary to achieve the best performance. We also perform a case study analysis to better understand the general word importance from the two dual-objective models. Using LIME, we observe that JointBERT can fail when there is too much similarity between entity descriptions which can drown out the non-match signal from a small but important subset of attribute values. In contrast, our AOA module identifies these essential words as demonstrated by analyzing the attention weights from the individual token embeddings. In addition, our results suggest that involving other sequence representation strategies to deal with the long textual descriptions can potentially improve the subtask performance.

Since most of the current models yield excellent performance on the larger training sizes, it may be desirable to explore other strategies to improve on the smaller datasets or zero-shot learning settings (i.e., no training samples). For example, a semi-supervised approach that uses a small portion of the training labels can be explored. Similarly, self-learning or contrastive learning approaches may yield generalizable representations that improve EM performance with fewer or no labeled data. As entity matching spans various domains, we have the option to employ domain adaptation techniques [17, 44] to enhance the resilience of our models. Another potential avenue for further improvement is incorporating the attribute description and attribute name with EM. Our preliminary results indicate introducing description structures instead of relying on special tokens (e.g., [COL]) can improve the robustness and performance of the EM model. Since the performance is sufficiently high (F1 > 95) for our experiments, we will explore the development of more challenging datasets, especially in the healthcare domain.

At last, large language models (LLMs) are capable of interpreting and generating sequences across a wide range of domains, including natural language, computer code, and protein sequences. There also are numerous LLMs for question and answering, such

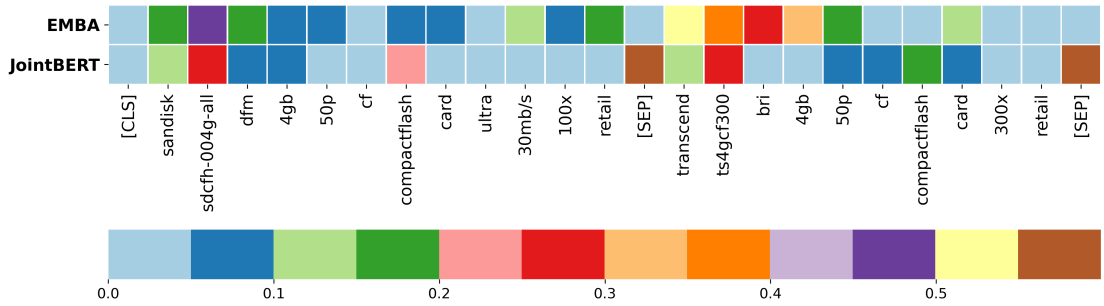

Figure 6: Attention visualization of an entity pair

as FLAN-T5 [4], LLaMA [38], and GPT-4 (backbone model of ChatGPT) [29]. We briefly explored larger language models such as ChatGPT for EM. However, given that the performance is already quite high for BERT-based models, the use of ChatGPT and GPT-4.0 has yet to yield significant improvements, especially in light of their higher computational requirements. We plan to further assess these LLMs for EM problems, such as how to apply the sequences to the LLMs efficiently, the utilization of the tokenization, and prompt design.

ACKNOWLEDGEMENTS

This work was supported by the National Science Foundation award IIS-2145411.

REFERENCES

[1] Ziawasch Abedjan, Xu Chu, Dong Deng, Raul Castro Fernandez, Ihab F Ilyas, Mourad Ouzzani, Paolo Papotti, Michael Stonebraker, and Nan Tang. 2016. Detecting data errors: Where are we and what needs to be done? *Proceedings of the VLDB Endowment* 9, 12 (2016), 993–1004.

[2] Ursin Brunner and Kurt Stockinger. 2020. Entity matching with transformer architectures-a step forward in data integration. In *23rd International Conference on Extending Database Technology, Copenhagen, 30 March-2 April 2020*. OpenProceedings.

[3] Hyunjin Choi, Judong Kim, Seongho Joe, and Youngjune Gwon. 2021. Evaluation of BERT and ALBERT sentence embedding performance on downstream NLP tasks. In *2020 25th International conference on pattern recognition (ICPR)*. IEEE, 5482–5487.

[4] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *ArXiv preprint* abs/2210.11416 (2022). https://arxiv.org/abs/2210.11416

[5] Yiming Cui, Zhipeng Chen, Si Wei, Shijin Wang, Ting Liu, and Guoping Hu. 2017. Attention-over-Attention Neural Networks for Reading Comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Vancouver, Canada, 593–602. https://doi.org/10.18653/v1/P17-1055

[6] Nilesh N. Dalvi, Vibhor Rastogi, Anirban Dasgupta, Anish Das Sarma, and Tamás Sarlós. 2013. Optimal hashing schemes for entity matching. In *22nd International World Wide Web Conference, WWW '13, Rio de Janeiro, Brazil, May 13-17, 2013*, Daniel Schwabe, Virgílio A. F. Almeida, Hartmut Glaser, Ricardo Baeza-Yates, and Sue B. Moon (Eds.). International World Wide Web Conferences Steering Committee / ACM, 295–306. https://doi.org/10.1145/2488388.2488415

[7] 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, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, Minneapolis, Minnesota, 4171–4186. https://doi.org/10.18653/v1/N19-1423

[8] Vincenzo Di Cicco, Donatella Firmani, Nick Koudas, Paolo Merialdo, and Divesh Srivastava. 2019. Interpreting deep learning models for entity resolution: an experience report using LIME. In *Proceedings of the Second International Workshop on Exploiting Artificial Intelligence Techniques for Data Management*. 1–4.

[9] Xin Luna Dong and Divesh Srivastava. 2013. Big data integration. In *2013 IEEE 29th international conference on data engineering (ICDE)*. IEEE, 1245–1248.

[10] 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.

[11] Ahmed K Elmagarmid, Panagiotis G Ipeirotis, and Vassilios S Verykios. 2006. Duplicate record detection: A survey. *IEEE Transactions on knowledge and data engineering* 19, 1 (2006), 1–16.

[12] Cheng Fu, Xianpei Han, Jiaming He, and Le Sun. 2020. Hierarchical Matching Network for Heterogeneous Entity Resolution. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, Christian Bessiere (Ed.). ijcai.org, 3665–3671. https://doi.org/10.24963/ijcai.2020/507

[13] Bar Genossar, Roee Shraga, and Avigdor Gal. 2023. FlexER: Flexible Entity Resolution for Multiple Intents. *Proceedings of the ACM on Management of Data* 1, 1 (2023), 1–27.

[14] Lise Getoor and Ashwin Machanavajjhala. 2012. Entity resolution: theory, practice & open challenges. *Proceedings of the VLDB Endowment* 5, 12 (2012), 2018–2019.

[15] Chaitanya Gokhale, Sanjib Das, AnHai Doan, Jeffrey F. Naughton, Narasimhan Rampalli, Jude W. Shavlik, and Xiaojin Zhu. 2014. Corleone: hands-off crowdsourcing for entity matching. In *International Conference on Management of Data, SIGMOD 2014, Snowbird, UT, USA, June 22-27, 2014*, Curtis E. Dyreson, Feifei Li, and M. Tamer Özsu (Eds.). ACM, 601–612. https://doi.org/10.1145/2588555.2588576

[16] Haibo He and Edwardo A Garcia. 2009. Learning from imbalanced data. *IEEE Transactions on knowledge and data engineering* 21, 9 (2009), 1263–1284.

[17] Huan He, Owen Queen, Teddy Koker, Consuelo Cuevas, Theodoros Tsiligkaridis, and Marinka Zitnik. 2023. Domain Adaptation for Time Series Under Feature and Label Shifts. *arXiv preprint arXiv:2302.03133* (2023).

[18] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[19] Jiacheng Huang, Wei Hu, Zhifeng Bao, Qijin Chen, and Yuzhong Qu. 2022. Deep entity matching with adversarial active learning. *The VLDB Journal* (2022), 1–27.

[20] Junjie Huang, Duyu Tang, Wanjun Zhong, Shuai Lu, Linjun Shou, Ming Gong, Daxin Jiang, and Nan Duan. 2021. WhiteningBERT: An Easy Unsupervised Sentence Embedding Approach. In *Findings of the Association for Computational Linguistics: EMNLP 2021*. Association for Computational Linguistics, Punta Cana, Dominican Republic, 238–244. https://doi.org/10.18653/v1/2021.findings-emnlp.23

[21] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of Tricks for Efficient Text Classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*. Association for Computational Linguistics, Valencia, Spain, 427–431. https://aclanthology.org/E17-2068

[22] Pradap Konda, Sanjib Das, AnHai Doan, Adel Ardalan, Jeffrey R Ballard, Han Li, Fatemah Panahi, Haojun Zhang, Jeff Naughton, Shishir Prasad, et al. 2016. Magellan: toward building entity matching management systems over data science stacks. *Proceedings of the VLDB Endowment* 9, 13 (2016), 1581–1584.

[23] Hanna Köpcke and Erhard Rahm. 2010. Frameworks for entity matching: A comparison. *Data & Knowledge Engineering* 69, 2 (2010), 197–210.

[24] Bing Li, Wei Wang, Yifang Sun, Linhan Zhang, Muhammad Asif Ali, and Yi Wang. 2020. GraphER: Token-Centric Entity Resolution with Graph Convolutional Neural Networks. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*. AAAI Press, 8172–8179. https://aaai.org/ojs/index.php/AAAI/article/view/6330

[25] Yuliang Li, Jinfeng Li, Yoshihiko Suhara, AnHai Doan, and Wang-Chiew Tan. 2020. Deep entity matching with pre-trained language models. *Proceedings of the VLDB Endowment* 14, 1 (2020), 50–60.

[26] Yuliang Li, Jinfeng Li, Yoshihiko Suhara, Jin Wang, Wataru Hirota, and Wang-Chiew Tan. 2021. Deep entity matching: Challenges and opportunities. *Journal of Data and Information Quality (JDIQ)* 13, 1 (2021), 1–17.

[27] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *ArXiv preprint* abs/1907.11692 (2019). https://arxiv.org/abs/1907.11692

[28] Sidharth Mudgal, Han Li, Theodoros Rekatsinas, AnHai Doan, Youngchoon Park, Ganesh Krishnan, Rohit Deep, Esteban Arcaute, and Vijay Raghavendra. 2018. Deep Learning for Entity Matching: A Design Space Exploration. In *Proceedings of the 2018 International Conference on Management of Data, SIGMOD Conference 2018, Houston, TX, USA, June 10-15, 2018*, Gautam Das, Christopher M. Jermaine, and Philip A. Bernstein (Eds.). ACM, 19–34. https://doi.org/10.1145/3183713.3196926

[29] OpenAI. 2023. GPT-4 Technical Report. arXiv:cs.CL/2303.08774

[30] Jonathan Ortigosa-Hernández, Inaki Inza, and Jose A Lozano. 2017. Measuring the class-imbalance extent of multi-class problems. *Pattern Recognition Letters* 98 (2017), 32–38.

[31] Matteo Paganelli, Francesco Del Buono, Andrea Baraldi, Francesco Guerra, et al. 2022. Analyzing how BERT performs entity matching. *Proceedings of the VLDB Endowment* 15, 8 (2022), 1726–1738.

[32] Ralph Peeters and Christian Bizer. 2021. Dual-objective fine-tuning of BERT for entity matching. *Proceedings of the VLDB Endowment* 14, 10 (2021), 1913–1921.

[33] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Doha, Qatar, 1532–1543. https://doi.org/10.3115/v1/D14-1162

[34] Anna Primpeli, Ralph Peeters, and Christian Bizer. 2019. The WDC training dataset and gold standard for large-scale product matching. In *Companion Proceedings of The 2019 World Wide Web Conference*. 381–386.

[35] Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, Balaji Krishnapuram, Mohak Shah, Alexander J. Smola, Charu C. Aggarwal, Dou Shen, and Rajeev Rastogi (Eds.). ACM, 1135–1144. https://doi.org/10.1145/2939672.2939778

[36] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *ArXiv preprint* abs/1910.01108 (2019). https://arxiv.org/abs/1910.01108

[37] Kai Sheng Teong, Lay-Ki Soon, and Tin Tin Su. 2020. Schema-Agnostic Entity Matching using Pre-trained Language Models. In *CIKM '20: The 29th ACM International Conference on Information and Knowledge Management, Virtual Event, Ireland, October 19-23, 2020*, Mathieu d'Aquin, Stefan Dietze, Claudia Hauff, Edward Curry, and Philippe Cudré-Mauroux (Eds.). ACM, 2241–2244. https://doi.org/10.1145/3340531.3412131

[38] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *ArXiv preprint* abs/2302.13971 (2023). https://arxiv.org/abs/2302.13971

[39] Jianhong Tu, Xiaoyue Han, Ju Fan, Nan Tang, Chengliang Chai, Guoliang Li, and Xiaoyong Du. 2022. DADER: Hands-off Entity Resolution with Domain Adaptation. *Proc. VLDB Endow.* 15, 12 (2022), 3666–3669. https://doi.org/10.14778/3554821.3554870

[40] Jiannan Wang, Tim Kraska, Michael J Franklin, and Jianhua Feng. 2012. CrowdER: Crowdsourcing Entity Resolution. *Proceedings of the VLDB Endowment* 5, 11 (2012).

[41] Zhen Wang, Xiang Yue, Soheil Moosavinasab, Yungui Huang, Simon M. Lin, and Huan Sun. 2019. SurfCon: Synonym Discovery on Privacy-Aware Clinical Data. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019*, Ankur Teredesai, Vipin Kumar, Ying Li, Rómer Rosales, Evimaria Terzi, and George Karypis (Eds.). ACM, 1578–1586. https://doi.org/10.1145/3292500.3330894

[42] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Association for Computational Linguistics, Online, 38–45. https://doi.org/10.18653/v1/2020.emnlp-demos.6

[43] Chen Ye, Shihao Jiang, Hua Zhang, Yifan Wu, Jiankai Shi, Hongzhi Wang, and Guojun Dai. 2022. JointMatcher: Numerically-aware entity matching using pre-trained language models with attention concentration. *Knowledge-Based Systems* (2022), 109033.

[44] Zhiqi Yu, Jingjing Li, Zhekai Du, Lei Zhu, and Heng Tao Shen. 2023. A Comprehensive Survey on Source-free Domain Adaptation. *arXiv preprint arXiv:2302.11803* (2023).

[45] Yu Zhang and Qiang Yang. 2021. A survey on multi-task learning. *IEEE Transactions on Knowledge and Data Engineering* (2021).

[46] Ziqi Zhang, Christian Bizer, Ralph Peeters, and Anna Primpeli. 2020. MWPD2020: Semantic Web Challenge on Mining the Web of HTML-embedded Product Data.. In *MWPD@ ISWC*.

[47] Chen Zhao and Yeye He. 2019. Auto-EM: End-to-end Fuzzy Entity-Matching using Pre-trained Deep Models and Transfer Learning. In *The World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019*, Ling Liu, Ryen W. White, Amin Mantrach, Fabrizio Silvestri, Julian J. McAuley, Ricardo Baeza-Yates, and Leila Zia (Eds.). ACM, 2413–2424. https://doi.org/10.1145/3308558.3313578

[48] Rui Zhu, Ziyu Wang, Zhanyu Ma, Guijin Wang, and Jing-Hao Xue. 2018. LRID: A new metric of multi-class imbalance degree based on likelihood-ratio test. *Pattern Recognition Letters* 116 (2018), 36–42.