Distribution Agreement

In presenting this thesis or dissertation as a partial fulfillment of the requirements for an advanced degree from Emory University, I hereby grant to Emory University and its agents the non-exclusive license to archive, make accessible, and display my thesis or dissertation in whole or in part in all forms of media, now or hereafter known, including display on the world wide web. I understand that I may select some access restrictions as part of the online submission of this thesis or dissertation. I retain all ownership rights to the copyright of the thesis or dissertation. I also retain the right to use in future works (such as articles or books) all or part of this thesis or dissertation.

Signature:

_____ _____

Jing Zhang Date

Attention-enhanced Deep Learning Models for Data Cleaning and Integration

By

Jing Zhang
Doctor of Philosophy

Computer Science and Informatics

Joyce C Ho, Ph.D.
Advisor

Jinho D. Choi, Ph.D.
Committee Member

Huan Sun, Ph.D.
Committee Member

Li Xiong, Ph.D.
Committee Member

Accepted:

Kimberly Jacob Arriola, Ph.D.
Dean of the James T. Laney School of Graduate Studies

Date

Attention-enhanced Deep Learning Models for Data Cleaning and Integration

By

Jing Zhang
B.E., Hunan University of Science and Technology, China, 2011
M.Sc., Carnegie Mellon University, PA, 2016

Advisor: Joyce C Ho, Ph.D.

An abstract of
A dissertation submitted to the Faculty of the
James T. Laney School of Graduate Studies of Emory University
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy
in Computer Science and Informatics
2023

Abstract

Attention-enhanced Deep Learning Models for Data Cleaning and Integration
By Jing Zhang

Data cleaning and integration is an essential process for ensuring the accuracy and consistency of data used in analytics and decision-making. Schema matching and entity matching tasks are crucial aspects of this process to merge data from various sources into a single, unified view. Schema matching seeks to identify and resolve semantic differences between two or more database schemas whereas entity matching seeks to detect the same real-world entities in different data sources. Given recent deep learning trends, pre-trained transformers have been proposed to automate both the schema matching and entity matching processes. However, existing models only utilize the special token representation (e.g., [CLS]) to predict matches and ignore rich and nuanced contextual information in the description, thereby yielding suboptimal matching performance. To improve performance, we propose the use of the attention mechanism to (1) learn the schema matches between source and target schemas using the attribute name and description, (2) leverage the individual token representations to fully capture the information present in the descriptions of the entities, and (3) jointly utilize the attribute descriptions and entity descriptions to perform both schema and entity matching.

Attention-enhanced Deep Learning Models for Data Cleaning and Integration

By

Jing Zhang
B.E., Hunan University of Science and Technology, China, 2011
M.Sc., Carnegie Mellon University, PA, 2016

Advisor: Joyce C Ho, Ph.D.

A dissertation submitted to the Faculty of the
James T. Laney School of Graduate Studies of Emory University
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy
in Computer Science and Informatics
2023

Acknowledgments

I would like to thank my esteemed supervisor – Dr. Ho for her invaluable supervision, support and tutelage during the course of my Ph.D. degree. Additionally, I would like to express gratitude to Dr. Sun and Dr. Choi for their treasured support which was really influential in shaping my research methods and critiquing my results. I would also like to thank Dr. Xiong for her impressive discussions on my work and inspiring suggestions for this dissertation.

Thanks my lab mates, colleagues and research team for the cherished time spent together in the lab, and in social settings. My appreciation also extends out to my wife Elle Hao and friends (Zhexiong Liu and Shaojun Yu) for their encouragement and support through my studies.

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

1.1 Motivation

Data integration is an important process in data management and analysis. Data from multiple sources can be stored in different formats and structures, and are subject to different rules and constraints. Data integration can overcome the challenges related to heterogeneity and interoperability in data by combining multiple data sources into a single, coherent view. Furthermore, inconsistencies and errors in data can be identified and resolved by combining information across multiple sources. As a result, the data integration process can also enhance the quality and accuracy of data by aligning and harmonizing this data. Thus, data integration is essential to data warehousing, business intelligence, and data mining applications by ensuring the data is consistent and reliable.

Usually, data integration contains four steps: (1) source selection, (2) schema matching, (3) entity matching, and (4) data fusion as shown in Figure 1.1. Among these stages, schema matching and entity matching are the most time-consuming aspects of the process. Schema matching and entity matching aim to identify and match the similarities and differences between different data sources, and to find

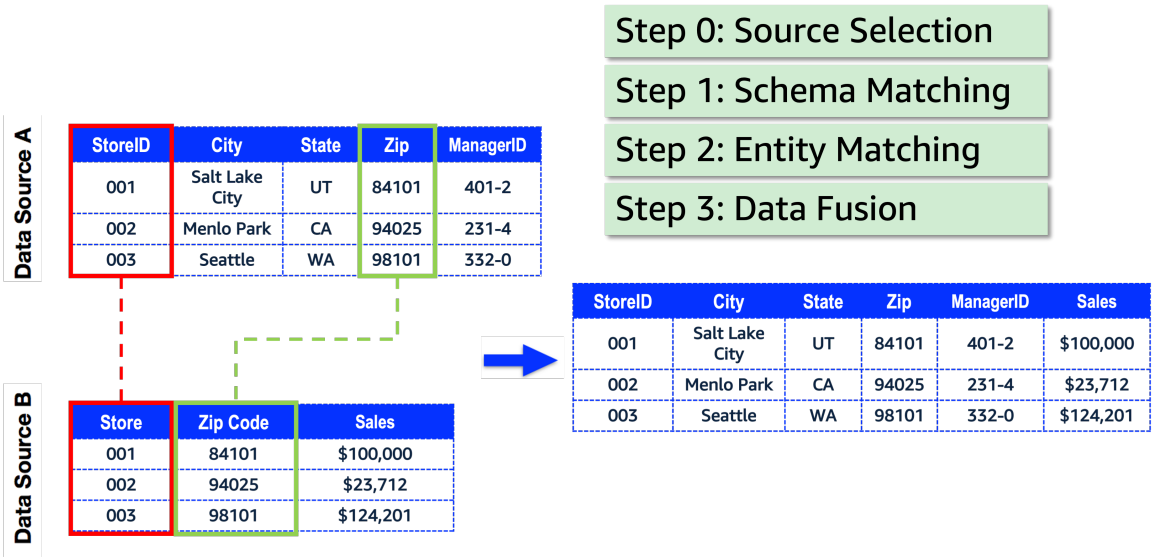


Figure 1.1: Data integration process: multiple data sources to one single view [27]

correspondences between entities in the data sources, respectively.

By identifying and matching common elements in the data sources, these methods allow data to be shared, queried, and analyzed across multiple sources. Such work is necessary for any downstream analytic tasks to extract value from the data. While schema matching and entity matching have been extensively studied in the database community, most solutions are often ad-hoc and require substantial effort to annotate the data or generate features [74]. As such, data scientists still spend more than 80% of their time curating the data [17].

Various automated schema matching methods have been proposed, including constraint-based approaches [28, 65, 72] and linguistic-based approaches [36, 39, 46, 83]. While the existing methods have achieved high performance in different domains, they suffer from several limitations. The constraint-based approaches analyze the element contents, which are not always guaranteed to be the same across the two schemas. Moreover, it assumes the data on both sides can be queried, which can violate privacy constraints. For the linguistic approaches, the relations are hand-coded between the two schemas or may not properly capture the similarity between the field

descriptions. Numerous matching tools (or matchers) can generate correspondences between pairs of schemas [9, 28]. Yet they rely on heuristic techniques. Recently, a deep learning (DL)-based model, ADnEV, was proposed to utilize similarity from existing matchers and post-process the results to work across domains [70]. However, the model is limited by the capability of existing matchers and may not generalize to all domains.

Similarly, recent entity matching models have posed the problem as semantic similarity matching. As a result, pre-trained natural language processing (NLP) models can serve as token-centric solutions to achieve impressive performance [18, 52, 54, 56, 62, 85]. These algorithms, leveraging the popular transformer models such as BERT [18], can automatically identify important entity description features using labeled examples without extensive engineering [73]. While the vanilla transformer model can be useful for entity matching tasks, there are several limitations. First, the model was designed to capture semantic interactions at the token level. However, existing entity matching models construct entity descriptions by concatenating all attribute values thereby introducing semantic discontinuity and impeding the overall performance. Second, while fine-tuning the transformer can be effective, it may not utilize the nuanced contextual information in the entity description. Finally, the masked language model 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 [61]) and improve the performance. Unfortunately, only using the special token representation from BERT can unnecessarily constrain the entity representation.

1.2 Research Contributions

Given the limitations of existing matching models to fully utilize the rich and nuanced contextual information in the attribute and entity descriptions, we propose new matching models that leverage the attention mechanism to improve performance. Attention has been proposed to focus the model on specific parts of the input when making predictions, rather than considering the entire input equally [8]. Attention has been shown to improve the model's performance on certain tasks, particularly when dealing with long sequences of data such as natural language processing and machine translation. It can also help to reduce the computational complexity of a model by allowing it to focus on the most relevant parts of the input, rather than processing the entire input equally. This can make the model more efficient and easier to train. In addition, attention mechanisms can improve the interpretability of a model by providing a way to visualize which parts of the input the model is focusing on when making predictions. This can be useful to gain insight into the model's behavior and to identify potential areas for improvement. In this dissertation, we introduce several attention variants that are specifically designed for schema matching and entity matching, as shown in Figure 1.2. Our main contributions are briefly summarized below.

1.2.1 Attention-over-Attention Deep Learning Schema Matching Model

We posit the schema matching process can be viewed as inferring the relatedness (or similarity) between the source and target fields to leverage recent advances in natural language processing and sentiment analysis. We propose `SMAT`, a DL-based model that uses attention-over-attention (AOA) mechanism [16] to capture the interactions between attribute names and their descriptions to identify the field-to-field mapping.

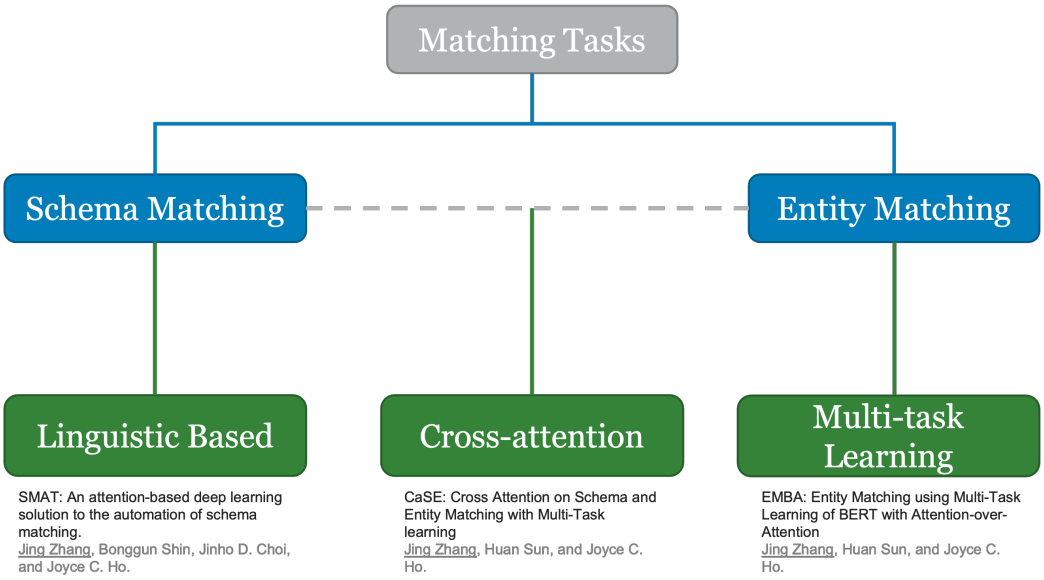


Figure 1.2: Dissertation Contributions

Our contributions include:

- A new DL model to automatically capture the semantic correlation from the source schema elements and their attributes to the target schema elements and their attributes based on the element and attribute descriptions. The model does not rely on existing matchers nor requires encoding prior domain knowledge.

- A new benchmark schema matching dataset for the healthcare domain, `OMAP`, that annotates several source to target conversions for sharing electronic health records. Existing schema matching models perform poorly on `OMAP` and illustrate the lack of generalizability to a variety of domains.

1.2.2 Multi-Task Learning with Attention-over-Attention for Entity Matching

We propose an AOA mechanism for the entity matching task to better capture the relationships across the pair of entity token representations. Rather than rely on

a single special token to combine the dual-objective of binary matching and entity identifier prediction as proposed in [62], we introduce `EMBA` to learn the aggregation weights from the individual entity tokens. This provides flexibility for each entity classification task to identify the important aspects of the description without requiring significant amounts of training data. In summary, our contributions are as follows:

- We propose to utilize the BERT token representations for both the auxiliary (entity identifier prediction) and main (entity matching) tasks. We align the token representations using the AOA mechanism to capture cross-entity token interactions to better capture the semantic similarity.
- We illustrate the benefits of AOA by visually analyzing the matching decisions of our model with existing state-of-the-art entity matching methods and empirically assessing the individual components of our model.

1.2.3 Cross-Attention Multi-task Learning for Schema and Entity Matching

We introduce a new multi-task paradigm that jointly captures both schema matching and entity matching simultaneously. To combine these two tasks, we utilize cross attention [91], which allows the model to quickly and accurately switch the focus between different viewpoints or perspectives. Existing research has shown that cross-view attention is associated with increased cognitive flexibility and adaptability, as well as improved social skills and decision-making abilities. Our new model, `CaSE`, can deal with realistic entity matching problems where attribute values may be missing or misaligned. Our contributions are as follows:

- A new cross attention-based DL model that can jointly model the attribute names and values to deal with incorrectly entered values or differing column

names.

- A newly curated benchmark dataset that combines both schema matching and entity matching tasks into a single model.

1.2.4 Organization

The remainder of this dissertation is organized as follows. Section 2 summarizes existing work in schema matching, entity matching, and various attention mechanisms. Section 3 introduces `SMAT`, the AOA-based model to automate the schema matching process. Section 4 uses the AOA mechanism to improve the entity matching performance using the vanilla transformer model. Section 5 introduces the new multi-task learning objective that combines schema matching and entity matching and our new cross attention model to deal with real-world databases. Section 6 concludes the dissertation and discusses the future directions of our work.

Chapter 2

Background

2.1 Schema Matching

Across many domains, data is collected using a wide variety of database systems with customized schemas developed for each company or purpose. As a result, similar collections of data can be stored using different physical formats, terminologies, or even logical organizations. Customized databases can hinder data exchange and data integration. Moreover, the process of standardizing different data formats into one common standard enables better downstream processing of the data, including large-scale analytics. Schema matching aims to establish the correspondence between the fields of a source and target database schema – a decisive initial step in the migration of different databases. Automation in schema matching has received steady attention in the database and artificial intelligence communities over the years. It has also been adopted as a practical and principled tool to improve the modeling and implementation of data exchange and data integration [3, 5, 22, 40, 47]. Yet, this problem remains largely unsolved. Existing solutions are not suitable for real-world database schemas [6], since field and table names can be cryptic and involve domain-specific abbreviations and acronyms. Although systems exist to support the creation

of schema matching [9, 20, 67], designing the matching process requires significant manual labor.

One line of schema matching work is the constraint-based approach. Most schemas contain constraints to define the attributes such as data types and value ranges, uniqueness, optionality, relationship types, and cardinalities [65]. Similarity can be measured by data types and domains, key characteristics (e.g., unique, primary, foreign), and relationship cardinality [2, 29, 55]. Recently, [4] proposed a hybrid of the constraint-based approach using key characteristics and the instance itself to create the meta-schema. Unfortunately, such approaches cannot readily handle the n:1 scenario that can be found in schema matching. For example, if the source schema contains "starttime" and "endtime" and the target schema contains "Duration", the meta-schema mapping can not generate and convert the two attributes into a single target.

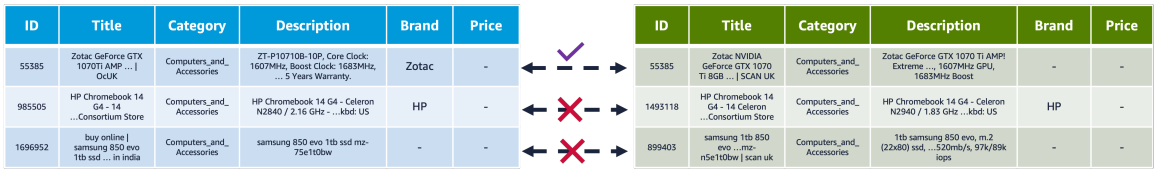
An alternative method is the linguistic content-based approach, which utilizes names and text to explore semantically similar schema elements. There are two primary linguistic data mapping techniques: name matching and description matching. The idea behind these techniques is to calculate similarity based on either the name of the fields or the description of the fields, respectively. In name matching, the similarity of names can be defined and measured through equality of names, equality of synonyms, similarity of names based on common substrings, and user-provided name matches. Examples include [38] which helps database designers visualize similarity and dissimilarity based on attribute names and [86] which uses a prescribed dictionary to obtain the aggregation among attributes. However, consulting a synonym lexicon has limitations since it is common to use abbreviations for attribute names (e.g., DOB for date of birth, SSN for Social Security number, etc.) and may not identify the relationships.

Description matching is based on the idea that schemas usually contain comments

or descriptions in natural language to express the intended semantics of schema elements. The process involves the identification of two semi-related data objects and the creation of mappings between them. In a recent work [46], the authors utilized the UMBC EBIQUITY-CORE technique [37] to obtain the similarity of the comments of schemas. Yet, it may not capture the similarity between the descriptions. For example, the similarity score between "the comment of the book" and "the review of the article" is 0.39 where 1 is highly similar and 0 is different. Another work used word embeddings to link datasets [30]; however, they only embedded the table name which may not yield sufficient information. [57] proposed a probabilistic graphical model and achieved a good score on precision and recall. Recently, ADnEV was proposed to utilize a DL technique to post-process the matching results from other matchers and performed better than the work in [21, 33, 57, 75]. However, the quality of the matchers limits the potential of the model.

2.2 Entity Matching

Entity matching is a crucial data integration problem that identifies whether two data entries refer to the same real-world entity. It is an essential process for cleaning and integrating data across single or distributed data sources [26, 42, 49, 61, 77]. In a variety of fields including e-commerce and medicine, entity matching serves as a longstanding critical problem in data integration [24, 53] and data cleaning [1]. Figure 2.1 illustrates the entity matching for different entries from different data sources. Matching entities accurately and quickly has enormous practical implications in commercial, scientific, and security applications [34]. However, the process of determining the pairs of matching entries can be time-consuming, especially in the presence of heterogeneous and large data sources. It still remains a challenging task for automated approaches because it requires a depth of language understanding and



ID	Title	Category	Description	Brand	Price
55385	Zotac GeForce GTX 1070Ti AMP ... | OcUK	Computers_and_ Accessories	ZT-P10710B-10P, Core Clock: 1607MHz, Boost Clock: 1683MHz, ... 5 Years Warranty.	Zotac	-
985505	HP Chromebook 14 G4 - 14 ...Consortium Store	Computers_and_ Accessories	HP Chromebook 14 G4 - Celeron N2840 / 2.16 GHz - ...kbd: US	HP	-
1696952	buy online | samsung 850 evo 1tb ssd ... in india	Computers_and_ Accessories	samsung 850 evo 1tb ssd mz-75e1t0bw	-	-

ID	Title	Category	Description	Brand	Price
55385	Zotac NVIDIA GeForce GTX 1070 Ti 8GB ... | SCAN UK	Computers_and_ Accessories	Zotac GeForce GTX 1070 Ti AMP! Extreme ..., 1607MHz GPU, 1683MHz Boost	-	-
1493118	HP Chromebook 14 G4 - 14 Celeron ...Consortium Store	Computers_and_ Accessories	HP Chromebook 14 G4 - Celeron N2940 / 1.83 GHz ...kbd: US	HP	-
899403	samsung 1tb 850 evo ...mz-n5e1t0bw | scan uk	Computers_and_ Accessories	1tb samsung 850 evo, m.2 (22x80) ssd, ...520mb/s, 97k/89k iops	-	-

Figure 2.1: Examples of EM to determine the matching entries from two sources

domain knowledge to match and distinguish entity information [52].

There are three categories of existing entity matching work: attribute-centric, token-centric, and hybrid-centric, which are defined by where the comparison level [25]. The first approach usually follows the alignment-comparison-summarization paradigm which involves comparing aligned attributes and aggregating the similarity vectors to determine the input for a binary classification system. Although these methods are generally successful, they fail in common real-world occurrences like schema heterogeneity (e.g., schemas are different across the two entities). Accordingly, most recent research has been token-centric [50], which compares individual attributes (e.g., tokens) and then aggregates the token-level comparison features into entity matching signals, or hybrid-centric [31], which aligns the tokens according to the attributes from two tables.

2.2.1 Single Task Deep Learning Models

Recent state-of-the-art entity matchers are DL based and approach entity matching as a binary classification problem. DeepER [25] trains entity matching models based on the LSTM [41] neural network architecture with word embeddings such as GloVe [63]. DeepER also proposed a blocking technique to represent each entry by the LSTM's output. DeepMatcher [56] extends recurrent neural networks with an attention mechanism and takes two data entries of the same quality as input and aligns their attributes before passing them on to the matching algorithm.

The vanilla transformer model (e.g., BERT [18] and RoBERTa [54]) has also been

proposed where the self-attention mechanism is used to carry out pair-wise semantic similarity of tokens between the two candidate records by using BERT's input format: [CLS] ENTITY1 [SEP] ENTITY2 [SEP].

More recent works utilize a pre-trained transformer and leverage the [CLS] token to determine whether two entities match [7, 73, 89]. DITTO further builds on BERT by serializing both data entries as one input while introducing structural tags [52]. As an example from Figure 2.1, it serializes each entity in the pair as e = [COL] title [VAL] Zotac $\cdots$ ocUK [COL] Brand [VAL] Zotac, and generates the pair as [CLS] e [SEP] e' [SEP]. Auto-EM [90] improves DL-based entity matching models by pre-training the entity matching model on an auxiliary task of entity type detection.

2.2.2 Multi-Task Deep Learning Models

Tangential to attribute, token, and hybrid-centric entity matching is the multi-task learning paradigm. Multi-task learning techniques have been used in concert with NLP to obtain more general representations, by complementing the main task objective with auxiliary training tasks. The idea is to improve the learning of a model for task t by using the knowledge contained in the tasks where all or a subset of auxiliary tasks are related [88]. By learning across related tasks, the learned representation can outperform a single task.

JointBERT [62] introduces the multi-task learning paradigm to the entity matching task to achieve a better performance compared with the single-task models. The model uses the [CLS] token in BERT for both the entity matching task and a multi-class object to predict the entity identifier of each of the two entity descriptions. As JointMatcher [85] incorporates the joint learning paradigm, we also categorize it as multi-task learning. JointMatcher customizes the sequences of the entity description by introducing two special tokens, [COL] and [VAL] to help identify similar segments and develop sensitivity to numerical values. JointBERT achieves higher performance

than JointMatcher on larger datasets, yet utilizes the [CLS] token for its multi-class objective which is suboptimal.

2.3 Common Deep Learning Models used for Embedding Schema and Entities

We briefly introduce two DL models that have been used for embedding entities or attributes.

2.3.1 Bidirectional LSTM Network

Given the variable length of text, long short-term memory (LSTM) networks were introduced to represent sequences of data [41] and are used to avoid the gradient vanishing problem in recurrent neural networks. Given an input word at the tth location in the sentence, e_t, the LSTM network can be formalized as follows:

$$f_t = \sigma\left(\vec{W}_f \cdot \left[\vec{h}_{t-1}, \vec{e}_t\right] + \vec{b}_f\right) \tag{2.1}$$

$$i_t = \sigma\left(\vec{W}_i \cdot \left[\vec{h}_{t-1}, \vec{e}_t\right] + \vec{b}_i\right) \tag{2.2}$$

$$o_t = \sigma\left(\vec{W}_o \cdot \left[\vec{h}_{t-1}, \vec{e}_t\right] + \vec{b}_o\right) \tag{2.3}$$

$$g_t = \tanh\left(\vec{W}_g \cdot \left[\vec{h}_{t-1}, \vec{e}_t\right] + \vec{b}_g\right) \tag{2.4}$$

$$\vec{c}_t = f_t * \vec{c}_{t-1} + i_t * g_t \tag{2.5}$$

$$\vec{h}_t = o_t * \tanh\left(\vec{c}_t\right), \tag{2.6}$$

where α is the sigmoid activation function, i_t, f_t, o_t are the input gate, forget gate, and output gate, respectively. The weights of the network are $\overrightarrow{W}_i$, $\overrightarrow{W}_f$, $\overrightarrow{W}_o, \overrightarrow{W}_g \in \mathbb{R}^{d\times(d+d_v)}$ and the bias of each gate are $\vec{b}_i, \vec{b}_f, \vec{b}_o, \vec{b}_g \in \mathbb{R}^d$, where d is the hidden dimension size and d_v is the size of input. Equation (2.1) represents the forget gate

and decides what information should be thrown away or kept. Equation (2.5) decides whether to update the cell state based on the previous hidden state or the input word. Finally the output gate, or Equation (2.6) decides what the next hidden state should be.

However, the standard LSTM can neglect the future contexts, which can negatively impact the predictive performance as the meaning of words may only make sense in the future context [68]. Bidirectional LSTM (BiLSTM) networks add a second layer, where the data flows in the opposite order of the first layer. Thus, the BiLSTM model can utilize both past and future information. The backward LSTM follows a similar process and is concatenated along with the forward LSTM. For any $t \in [1, 2, ..., n]$ where n is the sentence length, the concatenation of the forward and backward LSTM h_t is obtained as follows:

$$\overrightarrow{h_t} = \overrightarrow{LSTM_t}(e_1, ..., e_n) \tag{2.7}$$

$$\overleftarrow{h_t} = \overleftarrow{LSTM_t}(e_1, ..., e_n) \tag{2.8}$$

$$h_t = [\overrightarrow{h_t}, \overleftarrow{h_t}] \in \mathbb{R}^{2d} \tag{2.9}$$

BiLSTM has been used to convert each entity tuple to a distributed representation (or vector) which can be used to capture similarities between tuples [25].

2.3.2 Transformers

Transformers-based entity matching models [7, 18, 52, 54, 73, 89] learn the semantics of words better than previous entity matching solutions that were trained using word embeddings (e.g. word2Vector, GloVe, and FastText) and recurrent neural network architectures tailored to the domain. This is primarily due to the fact that the transformer calculates token embeddings for all tokens in an input sequence, and as a result, the embeddings it generates are highly contextual and capture seman-

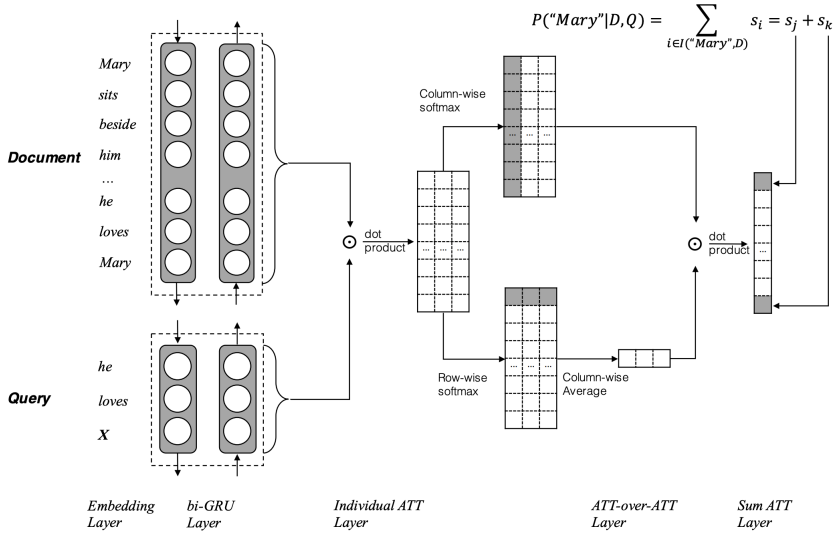


Figure 2.2: Illustration of AOA [16].

tic and contextual information. According to [52], the entity records often contain abbreviations (e.g., deluxe & dlux, and 2.0 & 2) and transformer-based models can embed *deluxe* and *dlux* similarly given their same respective contexts. This language understanding capability can improve the entity matching performance.

2.4 Attention Mechanisms

Attention was introduced to focus the model on specific parts of the input rather than considering the entire input equally [8]. The idea was to model the human cognitive function that only selectively pays attention to specific parts as needed. Attention has been widely used across a variety of application domains in conjunction with deep learning [58]. Here, we briefly introduce the two forms of attention relevant to this dissertation.

2.4.1 Attention-over-Attention (AOA)

AOA was first proposed for the question-answering task [16] shown in Figure 2.2. It allows the model to attend to multiple levels of abstraction within the input sequence

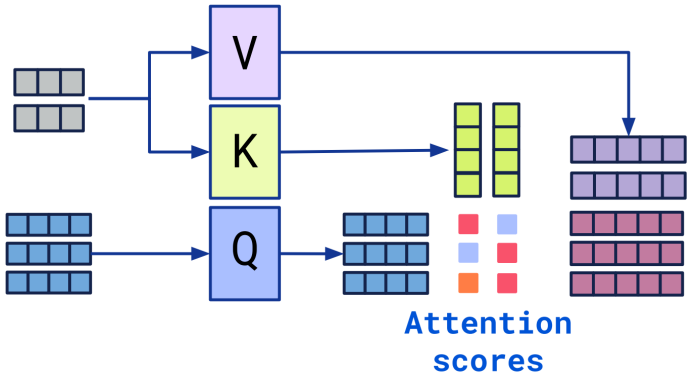


Figure 2.3: Illustration of cross attention [44].

or features through similarity generation (e.g., dot product). It captures the interactions of the query term and document-level sequences by summing each individual attention across both column and row. Since we can formulate the schema matching and entity matching tasks as sequence relatedness tasks, therefore, this mechanism can help to explore the relations of tokens between each matching pair.

2.4.2 Cross attention

Cross attention (illustrated in Figure 2.3) is an attention mechanism in Transformer architecture that mixes two different embedding sequences, and it is widely used in multi-modal and multi-scale architecture, such as image-text classification [44, 51, 10], machine translations [78, 35], and video recognition [84]. To capture the relatedness between the two sequences, cross attention combines one of the sequences as a query input, while the other as a key and value inputs. With the mechanism, it will amplify the attention weights that are similar with the query terms [51]. With cross attention mechanism, we can not only explore and visualize the relation between tokens, but it also brings the computational efficiency.

Chapter 3

Attention-over-Attention Deep Learning Schema Matching Model

Schema matching is an integral task for data exchange and data integration. As a motivating example, we consider the Observational Health Data Sciences and Informatics (OHDSI) community which seeks to bring out the value of health data through open-source, large-scale analytics and evidence gathering. Since healthcare data is collected using different systems and formats, the OHDSI community adopted the Observational Medical Outcomes Partnership (OMOP) Common Data Model (CDM) standard, to harmonize the heterogeneous data into a common data standard. Thus, each institution that participates in the community needs to perform the time-consuming task of manually mapping their original health data schema into the OMOP CDM, as data can not be readily shared due to patient privacy concerns. Further complications can arise from multiple attribute-to-attribute matching. Figure 3.1 illustrates a schema matching example for the MIMIC dataset to the OMOP CDM standard.

Given the rising importance of schema integration involving sensitive data, such as in healthcare, we focus on schema-level matching rather than instance-level or hybrid

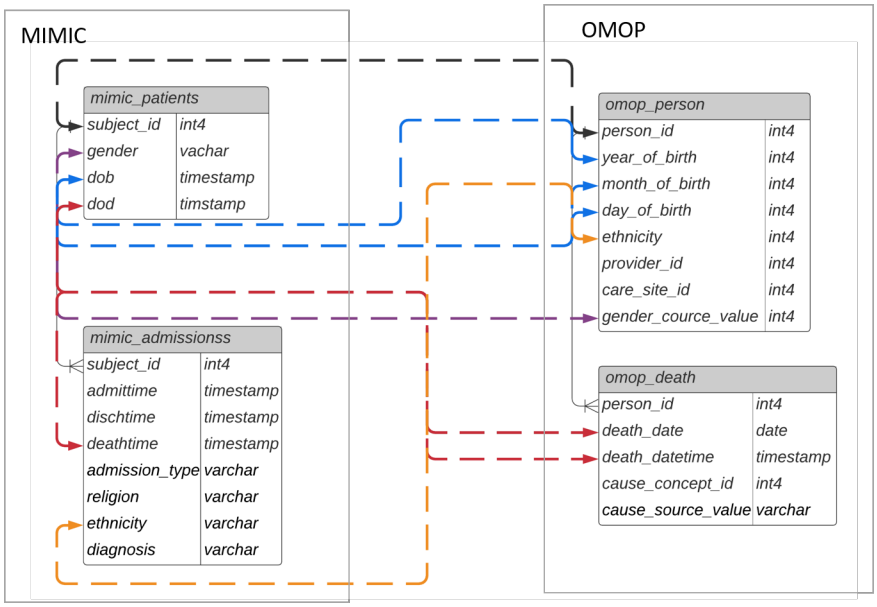


Figure 3.1: The schema matching design to convert the MIMIC dataset into the OMOP CDM standard [59]. For simplicity, only two elements from MIMIC (patients and admissions) are matched to OMOP (person and death). A match is given by double-arrow dashed edges.

schema matching. We posit that the schema matching process (i.e., source schema elements to target schema elements and its attributes matching) can be viewed as inferring the relatedness (or similarity) between the source and target fields. We propose `SMAT`, a DNN-based model with attention that extends recent advances in NLP and sentiment analysis. Our model can be used to automatically generate the matching between the source and target schemas without encoding domain knowledge. We perform an extensive evaluation of `SMAT` on a new benchmark dataset and three other popular schema matching datasets. The empirical results demonstrate the potential of `SMAT`. We also present a case study to gain further insight into the automation process.

3.1 Approach

We introduce SMAT, an attention-based DNN model to automate the schema matching between the source and target schemas. Under the paradigm where schema matching is viewed as inferring relatedness, the data dictionaries containing the elements and attributes descriptions/contents can be used to automatically capture the semantic correlation between the two fields without requiring explicit domain knowledge. In this section, we first formulate the problem and then introduce the various components of SMAT.

3.1.1 Problem Formulation

A schema contains a set of elements, such as relational tables and columns or XML elements and attributes. For our model, we extract tables and columns with their contents or descriptions to construct 'sentences'. SMAT assumes the semantic nature of the elements and attributes descriptions/contents can be used to learn universal sentence representations in a supervised manner. Since only the descriptions/contents of the fields themselves may not be sufficient to reveal the mappings, SMAT utilizes both the table description/content and column description/content to reveal the relations between them.

Formally, given two table descriptions/contents S_{T1} and S_{T2}, two attributes names N_{F1} and N_{F2}, and their descriptions/contents S_{F1} and S_{F2} from the source and target schema respectively, we construct two sets of sentences. Sentence set $S_{TF1} = \{S_{T1}, S_{F1}\} = \{w_1, w_2, ..., w_n\}$ consists of n words (e.g., **Des 1** in Table. 3.1), and sentence set $S_{TF2} = \{S_{T2}, S_{F2}\} = \{v_1, v_2, ..., v_{n'}\}$ consists of n' words (e.g., **Des 2** in Table. 3.1). Moreover, for the training data, there is an annotated label $L(S_{TF1}, S_{TF2})$ (e.g., **Label** in Table. 3.1), where $L(S_{TF1}, S_{TF2}) = 0$ denotes two fields are not related (i.e., mapped to each other), and $L(S_{TF1}, S_{TF2}) = 1$ denotes two sentences are related

(i.e., corresponding attribute-to-attribute matching). The task objective is then to classify the semantic relation of each sentence pair to reveal the attribute-to-attribute matching.

3.1.2 Overview

We observe that the task of determining the relatedness between two attributes' descriptions/contents is similar to inferring the similarity of two sentence pairs in NLP tasks. Since DNNs can be trained end-to-end without any prior knowledge [71] (i.e., no need to implement feature engineering), DNN models are utilized for text similarity tasks. InferSent utilized this approach to encode sentences with a downstream sentiment classification task and achieved higher performance than existing sentiment analysis models [15]. Yet there are two major limitations to adopting the InferSent model for the schema matching task. First, the element and attribute description may not contain sufficient information to distinguish it from others. Second, the descriptions may have abbreviations or words that have unknown word representations.

To address the above limitations of InferSent, `SMAT` consists of 4 major modules. First, the input embedding of the sentences utilizes a hybrid encoding to deal with large vocabularies for any input text. Second, a BiLSTM network (see Section 2.3.1) is used to capture the hidden semantics of the words in the description and the column name separately. Third, `SMAT` adopts the AOA mechanism (discussed in Section 2.4.1) to capture the correlation between the column name and its description. The final prediction layer uses the sentence representations to make an accurate classification. The overall architecture of `SMAT` is shown in Figure 3.3. We also introduce two techniques to deal with the class imbalance problem that is present in schema matching tasks.

3.1.3 Input Embedding

Existing word embedding models such as GloVe [63] are limited by vocabulary size or the frequency of word occurrences. As a result, rare words especially abbreviations like *ICD-9* are not captured and result in unknown tokens. Byte-Pair Encoding (BPE) is a hybrid between character- and word-level representations which could deal with the large vocabularies common in natural language corpora [69]. Instead of full words, BPE relies on sub-words units, which are extracted by performing statistical analysis of the training corpus. Using bytes, it is possible to learn a sub-word vocabulary of modest size (50K units) without introducing any "unknown" tokens. Thus, `SMAT` uses BPE as the initial tokenizing technique.

After text tokenizing with BPE, in order to map each word/sub-word w_i in the sentence $S_1 = \{w_1, w_2, ..., w_n\}$ to a high dimensional vector e_i, `SMAT` employs the embedding matrix $M \in \mathbb{R}^{d_v \times |V|}$ for word searching, where d_v is the word vector dimension and V is the fixed-sized vocabulary. Each word w_i is converted to its embedding e_i with the formula:

$$e_i = Mx_i \tag{3.1}$$

where x_i is a vector with size $|V|$, which is only 1 at the index e_i and 0 for all other positions. Each sentence, $S_1 = \{w_1, w_2, ..., w_n\}$, is then transformed into a real-valued vectors $emb_{S_1} = \{e_1, e_2, ..., e_n\}$. In addition to embedding the sentence descriptions, the column name is also embedded as a separate representation $emb_{N_{F1}}$. Our model uses GloVe embeddings [63] due to its popularity, but any word embedding representation can be used.

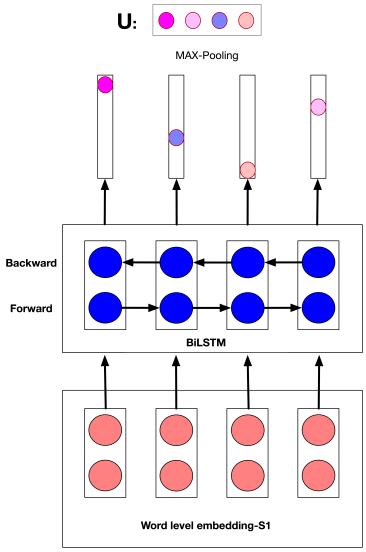


Figure 3.2: Bidirectional LSTM with max-pooling

3.1.4 BiLSTM

After obtaining the word representations, $emb_{N_{F1}}, emb_{S_{TF1}}, emb_{N_{F2}}, emb_{S_{TF2}}$, we feed these four sets of words into four BiLSTM networks respectively. The final representation from the BiLSTM is a concatenation of the hidden representations from the forward and backward LSTM. Therefore, each input (i.e. word) results in a vector representation of size $2d$. In order to capture all the information in the sequence, a common mechanism is to use the max-pooling to compress the sequence into a single vector as described in [14]. Our BiLSTM model is illustrated in Figure 3.2.

3.1.5 Attention-over-Attention

One limitation of using the representation from max-pooling is the inability to capture interactions between the attribute name and its description. Another approach to deal with the hidden semantic representations from BiLSTM is calculating the attention weights for the text via an AOA module. Our AOA module in Figure 3.3 uses mutual attention to simultaneously capture the relationships between attribute name to description and description to attribute name.

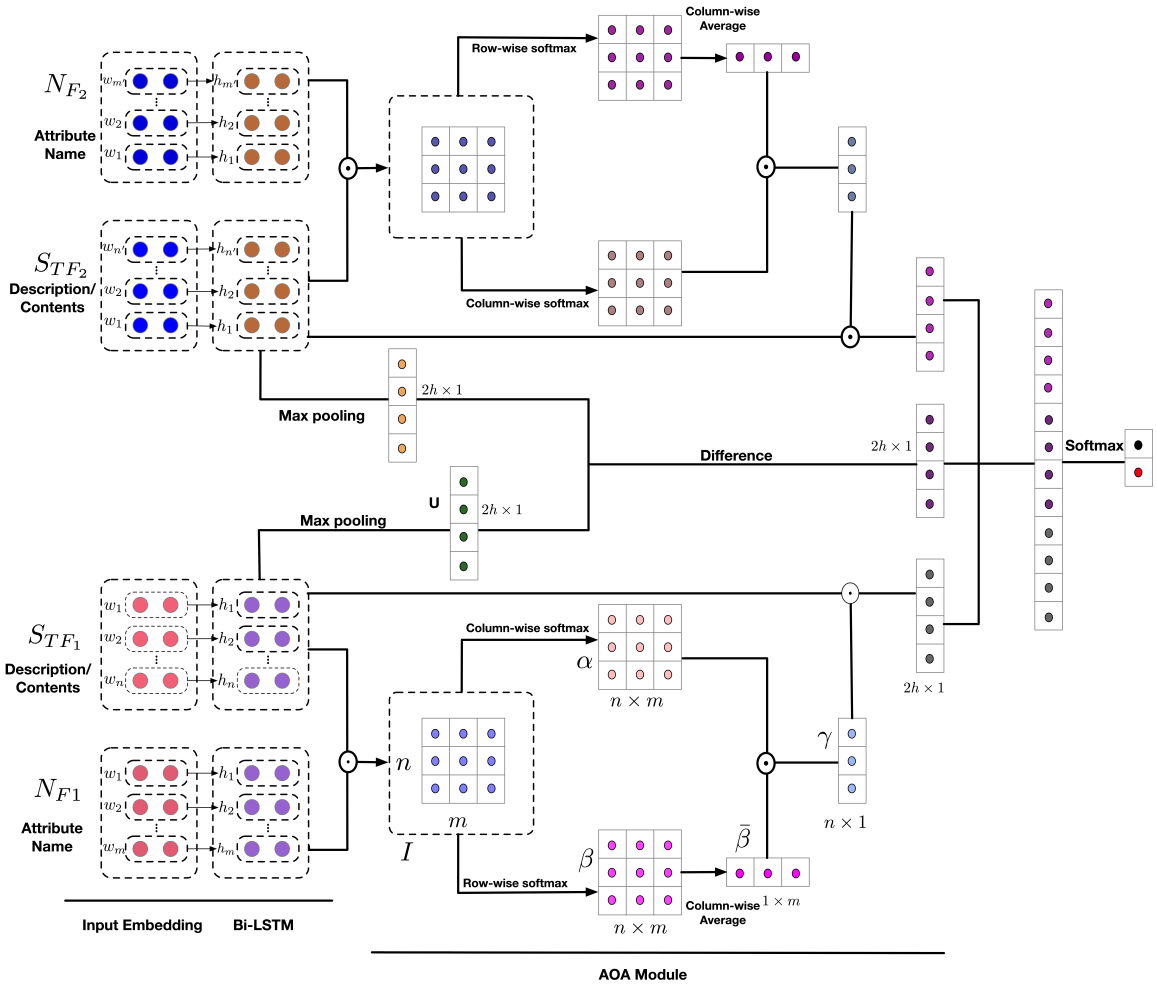


Figure 3.3: Illustration of SMAT's structure

Let $h_c \in R^{m \times 2h}$ denote the attribute name representation, where m is the attribute name length (i.e., number of words in the attribute name) and h is the hidden dimension. Let $h_s \in R^{n \times 2h}$ denote the element-attribute description representation, where n is the description length and h is the hidden dimension. The module first calculates the pair-wise interaction matrix $I = h_s \cdot h_c^T$, where the value of each entry represents the correlation of each word pair between the description and attribute name. A column-wise softmax and row-wise softmax are applied to the interaction matrix I, to obtain the attribute name to description attention, α, and description to attribute name attention, β, respectively. Thus for the t^{th} attribute word and k^{th} text description, the associated attentions are:

$$\alpha(t) = softmax(I(1,t), I(2,t), \cdots, I(m,t)) \tag{3.2}$$

$$\beta(k) = softmax(I(k,1), I(k,2), \cdots, I(k,n)) \tag{3.3}$$

Then, the attribute name-level attention $\bar{\beta}$ is calculated using a column-wise averaging of β. This attention indicates the important words in the attribute name. Finally, the sentence-level attention $\gamma \in R^n$ can be obtained by a weighted sum of each individual attribute name to description attention α. By considering the contribution of each aspect word explicitly, the AOA module learns the important weights for each word in the sentence.

$$\alpha_{ij} = \frac{\exp(I_{ij})}{\sum_i \exp(I_{ij})} \tag{3.4}$$

$$\beta_{ij} = \frac{\exp(I_{ij})}{\sum_j \exp(I_{ij})} \tag{3.5}$$

$$\bar{\beta} = \frac{1}{n} \sum_i \beta_{ij} \tag{3.6}$$

$$\gamma = \alpha \cdot \bar{\beta}^T \tag{3.7}$$

3.1.6 Data Augmentation & Controlled Batch Sample Ratio

As attribute-to-attribute mapping generally results in a skewed distribution, SMAT uses data augmentation and controlled batch sample ratio (CBSR) to achieve better predictive performance. Data augmentation occurs on two levels. The first is to generate new positive samples using synonyms for different words in the descriptors. For example, an augmented sample may replace the word "uniquely" with "unambiguously" and "identify" with "describe". However, since the number of synonyms is limited, we utilize a second technique to improve the attribute name description. We use the part-of-speech (POS) tags for the descriptions and concatenate the identified nouns to enlarge the dataset safely.

Since SMAT uses batch stochastic gradient descent (SGD) to learn the parameters, a batch can contain no positive samples and thus only properly learn the representation for negative samples. Thus, we controlled the ratio of positive samples in each batch size to ensure that our model learns from a few positive examples for each batch [23]. Note that since the positive samples are small, they are likely to be chosen repeatedly, while there is diversity in the negative samples.

3.2 OMAP: A New Benchmark Dataset

Since existing schema matching datasets only span purchase orders, web forms, and bibliographic references, we created OMAP, a new benchmark schema-level matching dataset that annotates several source-to-target mappings in the healthcare domain. Healthcare data is collected worldwide using a wide variety of coding systems. To draw conclusions with statistical power and avoid systematic biases, a large number of samples should be analyzed across disparate data sources and patient populations. Such broad analyses require data harmonization to a common data standard (e.g., the Observational Medical Outcomes Partnership (OMOP) Common Data Model (CDM)

Table 3.1: An example entry from the OMAP dataset.

CDM schema	Source schema	CDM description (Des 1)	Source description (Des 2)	Label
person-person_id	beneficiary summary-desynpuf_id	the person domain contains records that uniquely identify each patient in the source data who is time at-risk to have clinical observations recorded within the source systems.a unique identifier for each person.	beneficiarysummary pertains to a synthetic medicare beneficiary. beneficiary code	1

standard) to facilitate evidence gathering and informed decision-making [59]. Since patient data cannot be queried due to privacy concerns, schema-level matching is of great importance. OMAP maps between three different healthcare databases (source schema) and the OMOP CDM standard (target schema).

1. MIMIC-III [45]: A publicly available intensive care unit (ICU) relational database from the Beth Israel Deaconess Medical Center.

2. Synthea [80]: An open-source dataset that captures the medical history of over one million Massachusetts synthetic patients.

3. CMS DE-SynPUF [13]: A set of realistic claims data generated from 5% of Medicare beneficiaries in 2008.

For each dataset, the element table name with its descriptions and attribute column name with its descriptions are used to construct a sentence. The label is based on the final extract, transform and load design. If the table-column in the source schema was mapped to a table-column in the OMOP CDM the label is 1, otherwise it is 0. Table 3.1 provides one example from the OMAP dataset.

The summary statistics for each of the three conversions are captured in Table 3.2. Note that the dataset does not contain any patient information, only attributes and their descriptions.

Table 3.2: Summary statistics of each conversion captured in OMAP.

Data source	# elements	# attributes	# positive labels	# sentence pairs
MIMIC	25	240	129	64080
Synthea	12	111	105	29637
CMS	5	96	196	25632

Table 3.3: Summary statistics of the additional benchmark datasets used.

Data source	# elements	# related	# pairs	# Domains
Purchase Order[20]	50-400	659	63933	1
OAEI[1]	80-100	9494	825021	1
Web-forms[32]	10-30	5548	201769	18

3.3 Experiments

We designed the experiments to answer the question: How *accurate* is SMAT in automating the schema matching?

3.3.1 Datasets

We used the OMAP dataset and three popular schema matching benchmark datasets (summarized in Table 3.3). Reference matches in additional datasets were manually constructed by domain experts and considered as ground truth for our purposes. The experiments are performed for each dataset and the settings are consistent with existing schema matching papers [33, 57, 75]. For each dataset, 80% was used to train the initial prediction model, 10% was used to further tune the weights, and the remaining 10% was used to evaluate the experiments.

[1]The OAEI competitions can be found at http://oaei.ontologymatching.org/2011/benchmarks/

3.3.2 Baseline Models

SMAT is evaluated against five baseline models. For data sensitivity purposes, we focused only on schema-level matching. The entity matching solutions that involve semantic relatedness technique are chosen to represent the existing schema matching or entity matching work.

- **ADnEV** [70]. A schema matching model that utilizes DNN to post-process results from state-of-the-art matchers in an iterative manner.
- **InferSent** [15]. A state-of-the-art sentence embedding model that classifies the sentiment between two sentences. The last layer is modified to tackle a binary classification task. GloVe embeddings [63] are used for the input sentences.
- **DeepMatcher** [56]. An entity matching solution that customizes the recurrent neural network architecture to aggregate the attribute values and then compares the aggregated representations of attribute values.
- **DITTO** [52]. A state-of-the-art entity matching model that cast the problem as a sequence-pair classification and fine-tunes RoBERTa [54], a pre-trained Transformer-based language model.
- **BERT** [18]. Bidirectional Encoder Representations from Transformers (BERT) has achieved state-of-the-art results in many natural language understanding tasks. We fine-tuned the pre-trained BERT-base-uncased model on our datasets.

3.3.3 Experimental Setup

We implemented SMAT and the baseline models in Python 3.6 using PyTorch. Performances were measured on the Google Cloud Platform with Intel Xeon E5 v3 CPU @ 2.30Ghz, and a Nvidia Tesla K80 with 12 GB Video Memory.

For experiments in this paper, the embedding dimension is 300. The number of hidden units of BiLSTM is 1024 for InferSent and 300 for SMAT. For the classification model, we apply a fully connected layer with one hidden layer of 512 hidden units. SGD is chosen as the optimization algorithm with a batch size of 64. The learning rate and weight decay are 0.1 and 0.99 for InferSent and 0.001 and 0.99 for SMAT. For AdnEV, DeepMatcher, DITTO, and fine-tuning BERT model, Adam is chosen as the optimization algorithm with a learning rate of 0.001, 0.001, $3e-5$, $2e-5$, respectively, and the batch size as 64, 64, 64, and 32 respectively. These parameters were obtained from initial experiments on a subset of the training data as they provided the most robust performance across multiple runs.

3.4 Predictive Performance

Evaluation of SMAT with existing baseline models. Table 3.4 summarizes the results of the six models tested on the six datasets. We observe that the precision and recall varies depending on the dataset suggesting differences in the semantic content of their attribute names and descriptions. The results demonstrate that SMAT does not require additional hand-coding due to the overall strong performance. It achieves the best performance across all three metrics in 3 of the datasets (OAEI, MIMIC, CMS). It also yields the best F1 score for all but the Purchase Order dataset. Thus, our proposed model is fairly versatile.

ADnEV achieves a higher precision on Purchase Orders and Webforms and a better F1 score on Purchase Orders than others. Yet, SMAT outperforms the ADnEV model on OAEI and Web-forms in terms of F1 score by 12.4% and 16.1% respectively. Moreover, the results on the OMAP datasets illustrate the pitfall of ADnEV. Since ADnEV leverages other matchers, it is limited by the capability of the matchers. Thus, ADnEV may not be suitable for all domains. Furthermore, comparisons of the

Table 3.4: Comparison of precision (P), recall (R), and F1 (F) on the datasets.

Dataset	ADnEV			InferSent			DeepMatcher			DITTO			BERT			SMAT		
	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F
MIMIC	0.08	34	0.16	9.8	76.9	17.4	0.04	38.1	0.09	0.3	46.2	0.6	0.4	84.6	0.7	**11.5**	**84.6**	**20.2**
CMS	0.49	44	0.97	20.8	80.0	32.9	0.31	60.7	0.62	2.4	40	4.5	2.4	55.0	4.5	**33.9**	**95.0**	**50.0**
Synthea	0.14	21	0.28	19.2	90.9	31.7	0.06	48.8	0.13	0.7	63.6	1.3	0.9	**100**	1.8	**24.4**	90.9	**38.5**
Purchase Order	**80**	77	**78**	14.3	59.6	23.1	48.9	80.2	60.8	54.5	98.6	70.2	54.0	98.2	69.7	57.9	**99.5**	73.2
OAEI	78	76	76	84.5	99.9	91.5	56.1	62.9	59.3	80.5	99.9	89.2	78.3	99.8	87.8	**87.8**	**99.9**	**93.5**
Web-forms	**81**	69	72	68.4	**99.8**	81.2	48.2	74.5	58.5	68.8	95.5	80	63.5	96.3	76.5	79.1	99.3	**88.1**
Average	34.3	49.9	32.5	33.6	78.2	43.3	22.0	56.8	25.8	29.7	69.4	35.4	28.6	**88.8**	34.7	**45.7**	87.0	**56.3**

DNN-based models (InferSent, Fine-tuned BERT, and SMAT) and ADnEV in terms of F1 and recall also illustrate the power of end-to-end training without requiring additional feature engineering.

For the OMAP dataset, SMAT achieves a higher precision and recall score suggesting that the prediction capability of SMAT is better than the other models. However, the precision across these four datasets is noticeably lower than those of Purchase Order, OAEI and Web-forms. This may be a result of the more complex textual information in the healthcare domain. Moreover, there are many abbreviations that can prevent the general model from achieving a higher score. Simply finetuning the BERT could not help much. For example, the Purchase Order contains abbreviations much less than OMAP, after finetuning the BERT, it could improve the results comparing with other none transformer based model. This highlights the importance of benchmarking the models across various applications and supports the development of OMAP.

The results also capture the difference that arises from schema-level matching. Even though DITTO and DeepMatcher perform well in the entity matching task, they do not offer comparable performance across the different datasets. This may be due to the inconsistencies across the datasets present in the textual information. Moreover, InferSent seems to provide better F1 scores compared to the more complex transformer models outside of the Purchase Dataset. This suggests that the Bi-LSTM based sentence modeling approach shared by InferSent and SMAT may offer better predictive power compared to the more complex transformer-based models. In comparing InferSent and SMAT, the results suggest that SMAT's attention mechanism

and representation can help capture the elements and attributes in source schema and target schema differences better than the other models regardless of whether the textual information is rich (OMAP) or not (Purchase Order, OAEI and Web forms).

3.5 Ablation Study

To gain further insights of the various components in SMAT, we conducted an ablation study. In particular, we examined the effectiveness and contributions of the AOA module, the BiLSTM module, and pooling strategy.

- *SMAT w/o AOA*: The AOA module in Figure 3.3 is dropped and instead the outputs of the attribute name BiLSTM and description BiLSTM are max-pooled together and concatenated with the difference of the two descriptions.
- *SMAT w/o BiLSTM*: The BiLSTM is substituted with the LSTM.
- *SMAT w/o Max-pooling*: The Max-pooling is changed to mean-pooling.
- *SMAT w/o attribute name*: The attribute name is dropped from the input, and the description itself is fed into the AOA module, so it calculates the mutual information with itself. The difference between the two descriptions and the two AOA outputs of descriptions are then concatenated together.
- *SMAT w/o DA*: The data augmentation with additional positive samples and concatenation of nouns to the column name is omitted during the training process.

Table 3.5 shows the results of ablation experiments on F1. It can be seen that the complete SMAT model outperforms the rest models on F1. In particular, comparing the result with *SMAT w/o AOA* illustrates the importance of the AOA module. The module captures the interaction between the description/content of attribute and

Table 3.5: Results for ablation experiments on F1. The best performance is bolded.

Datasets	SMAT	w/o AOA	w/o BiLSTM	w/o Max-pooling	w/o attribute name	w/o DA
MIMIC	**20.2**	18.3	18.8	6.7	18.2	18.6
CMS	**50.0**	36.3	39.9	13.8	38.6	39.0
Synthea	**38.5**	26.1	31.1	17.7	33.3	36.4
Purchase Order	**73.2**	26.2	59.6	64.7	28.5	58.9
OAEI	**93.5**	90.7	91.1	81.4	91.2	92.4
Web-form	**88.1**	84.9	86.4	81.1	84.1	82.3

the correlated attribute name better than max-pooling the outputs from BiLSTM. The same conclusion can also be drawn by comparing the result *SMAT w/o AOA* and *SMAT w/o attribute name*, the precision from *SMAT w/o AOA* is lower than that from *SMAT w/o attribute name*. It means even when there is no attribute name feature and data augmentation, the AOA module can still generate more useful features. Without BiLSTM module the performance drops significantly. Comparing the importance between BiLSTM and pooling strategies, we can see that the BiLSTM could be a more important module.

3.6 Case study

We compared the prediction from the different models to illustrate the potential and difficulty of automating the schema matching process. For illustrative purposes, we focus on the conversion of MIMIC-III database to the OMOP CDM as it has one of the most extensive column and table descriptions.

3.6.1 Correct prediction from all methods

An example where all the methods correctly assess the relatedness of two attributes is the match for the attribute **drug_exposure-start_date** in OMOP to **prescriptions-startdate** in MIMIC, which is shown in the first row of Table 3.1. As can be seen in the Table, there are common synonyms between the two field descriptions such as

"drug", "medication", and "prescription" as well as "start date", "date prescription was filled" and "date when the prescription started". The similar words as well as the simple context structure results in the automatic mapping between these two fields.

3.6.2 Correct prediction from only SMAT

An example where only our method correctly assess the relatedness of two attributes is the match for the attribute **drug_exposure-start_datetime** in OMOP to **inputevents_mv-starttime** in MIMIC. The description for the attribute in MIMIC is "*inputevents_mv is input drug data for patients from metavision icu databases. starttime records the start time of a drug input event.*" For SMAT, AOA module could extract the relation between attribute name, description tags, and the description, which means inputevents_mv related drug in MIMIC attribute name and its description. Similarly AOA also can capture the **drug_exposure-start_datetime** with its description about drug. However, InferSent, which utilizes max-pooling, potentially loses the concentration on the term "drug" because of the long sentence. Thus, InferSent and the other models often fail when the field description and table description are long. Also, DITTO predicts the relation with specific domain topic input, however, there is no healthcare-related topic word in their topic dictionary, so it also failed on the prediction. With the sub-words of the column name, the BERT also could not learn the useful information between the column name and its descriptions.

3.6.3 Incorrect prediction from all models

The final example is a match where all the methods fail to correctly identify the relatedness between the attributes **visit_detail-care_site_id** in OMOP and **transfers-curr_wardid** in MIMIC. The element and attribute description of **visit_detail-care_site_id** is *visit_detail table is an optional table used to represent details of each record in the parent visit_occurrence table. for every record in visit_occurrence table*

there may be 0 or more records in the visit_detail table with a $1:n$ *relationship where n may be 0. the visit_detail table is structurally very similar to visit_occurrence table and belongs to the similar domain as the visit; a foreign key to the care site in the care site table that was visited.* The description of **transfers-curr_wardid** is *transfers are physical locations for patients throughout their hospital stay; curr_wardid contains the current ward in which the patient stayed.* One of the possible explanations for the failure are that the two descriptions are ambiguous, and it is unclear that visit and wards are equivalent to one another. In addition, the operation object ID is also vague and prevalent in many other fields. Moreover, the second description only mentions ward without any additional attributes which hampers the contextual similarity. Thus, under such scenarios, fully automating the schema mapping may not be feasible without better field descriptions.

Chapter 4

Multi-Task Learning with Attention-over-Attention for Entity Matching

DNN methods have become the de-facto standard for tackling entity matching. By posing entity matching as semantic similarity matching, pre-trained NLP models can serve as token-centric solutions to achieve impressive performance [18, 52, 54, 56, 62, 85]. These algorithms leverage the popular transformer models such as BERT to automatically identify important entity description features using labeled examples without extensive engineering [73].

Unfortunately, the entity matching training samples may not provide sufficient information to learn the relatedness between entity pairs. As such, other sub-tasks can enrich the pragmatic knowledge encoded by BERT and improve performance. JointBERT [62] introduces the multi-task learning formulation by adding auxiliary tasks of identifying the individual entity classes to achieve state-of-the-art performance across some of the datasets [62]. However, one major drawback is that it fails to fully leverage the token representation power as only the representations of the

special [CLS] token are used for the downstream tasks. Although it can be used to represent the meaning of the entire sentence, it cannot be used 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 subword 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 [11, 43].

In this chapter, we demonstrate that individual token representations should be exploited for both the auxiliary and main tasks 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 dual-objective of binary matching and entity identifier prediction. We present a multi-class classification module for the entity identifier (such as GTIN, ISBN, or ORCID numbers) prediction task that learns the aggregation weights from the individual entity tokens using the AOA mechanism as shown in Figure 4.1. This provides flexibility for each entity classification task to identify the important aspects of the entity token 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 entity matching model, JointBERT [62], the joint matching model, JointMatcher [85], and several vanilla transformer-based entity matching models [18, 52, 54, 56] on four entity matching 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%.

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			

Figure 4.1: An example of the input to the BERT-based models and the prediction results from JointBERT and EMBA

4.1 Approach

4.1.1 Problem Definition

Given two 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 (i.e., title, description, and brand in Figure 4.1) of an entity description, the goal is to learn (1) whether the two entities refer to the same object (i.e., entity matching 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 task is known as a multi-class classification task where each entity ID is a class.

4.1.2 Overview

EMBA follows the common BERT input format used for entity matching. 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]. As shown in Figure 4.2, the output representations of the different entity tokens of the last encoder layer from BERT (i.e., $E_{e_1} = \{E_{D^1_{e1}}, \cdots, E_{D^m_{e1}}\}$ and $E_{e_2} = \{E_{D^1_{e2}}, \cdots, E_{D^n_{e2}}\}$) are passed into different modules, which are shown as follows,

- E_{e_1} will feed to a linear layer and softmax layer trained to predict the first entity identifier, ID_{e_1}, based on its description.

- E_{e_2} will feed to a linear layer and softmax layer trained to predict the second entity identifier, ID_{e_2}, based on its description.

- Both E_{e_1} and E_{e_2} will feed into the AOA module to capture the interaction across the pair of entities, whose output is then passed to a linear layer and a softmax layer trained on the entity matching task.

There are two major architecture modifications from JointBERT related to the use of the individual tokens (i.e., $\{E_{D^1_{e1}}, \cdots, E_{D^m_{e1}}\}$ and $\{E_{D^1_{e2}}, \cdots, E_{D^n_{e2}}\}$) and also the use of the AOA module for the entity matching task. In JointBERT, the training objective consists of two parts: (1) a binary cross-entropy loss for the entity matching problem and (2) one or more cross-entropy losses for the auxiliary task such as the entity identifier ID_{e_i} prediction based on their respective description D_{e_i}. The output representation of the [CLS] token is used to learn the task-specific modules for the two parts of the objective. While [CLS] is commonly used in NLP for many downstream classification tasks, this may not always yield the best performance. 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].

4.1.3 Entity Identifier 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 [11, 43]. Thus, for the entity identifier prediction task, we propose the use of the token embeddings themselves as the input representation for the cross-entropy loss. We note that one naïve approach is to use a different special token

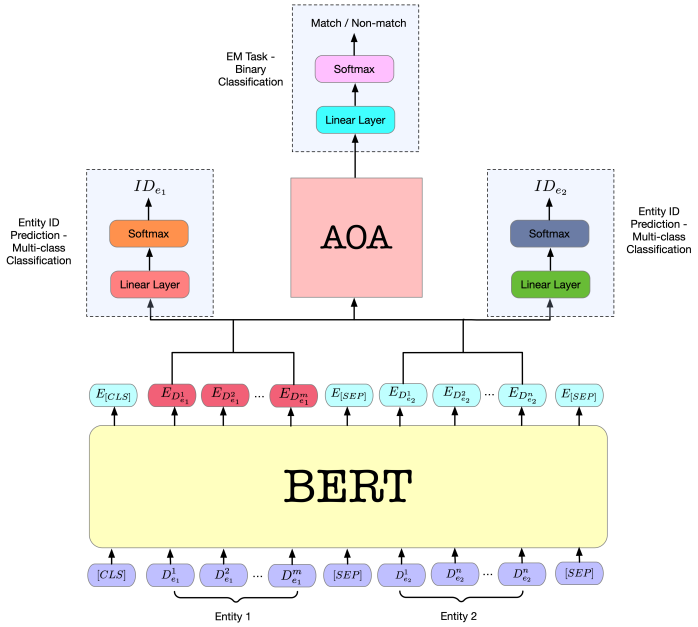


Figure 4.2: EMBA framework

(e.g., [SEP] token) for the second entity identifier 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. Therefore, EMBA uses the token embeddings from the entity description, $E_{D_{e_i}}$ 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. We also note that since each entity description has different lengths, m and n for entities 1 and 2, respectively, the weights are task-specific.

4.1.4 Attention-over-Attention for Entity Matching Prediction

For the entity matching problem, we again use the token representations for the two entities, E_{e_1} and E_{e_2}. These representations are fed to an AOA module to model the

token-level interactions between these two pairs. 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 captures the correlations between the entity description using two mechanisms. Notice that $E_{e1} \in R^{m \times h}$ denotes the first entity 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 second entity representation, where n is the second entity token length. The module first calculates the pair-wise interaction matrix $I = E_{e1} \cdot E_{e2}^T$, where the value of each entry represents the correlation of each token pair between the first and second entity. 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 the second entity when considering the first entity. 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 entity 1 and the t^{th} token embedding from entity 2, the associated attentions are:

$$\alpha(t) = softmax(I(1,t), I(2,t), \cdots, I(m,t)) \tag{4.1}$$

$$\beta(k) = softmax(I(k,1), I(k,2), \cdots, I(k,n)) \tag{4.2}$$

Then, the averaged second entity attention $\bar{\beta}$ is calculated using a column-wise averaging of β. Finally, the attention-over-attention $\gamma \in R^m$ 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 two entities are matching.

4.1.5 Dual Objective Training

`EMBA` uses the binary cross-entropy loss (BCEL) for the entity matching task and the cross-entropy loss (CEL) for the entity identifier prediction (multi-class prediction). Let $y_{em_i}, y_{e1_i}, y_{e2_i}$ denote the entity matching label, and the two entity identifiers, then we define the loss L as follows,

$$L_i = BCEL(y_{em_i}, \hat{y}_{em_i}) + CEL(y_{e1_i}, \hat{y}_{e1_i}) + CEL(y_{e2_i}, \hat{y}_{e2_i}) \tag{4.3}$$

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 entity matching classification, first entity ID prediction, and second entity ID prediction. During the training stage, both objectives are jointly optimized, so that the entity matching task will be improved by the other two multi-class classification tasks training simultaneously.

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. (4.3);
        Compute gradient: $\nabla(\theta)$;
        Update model: $\theta = \theta - \eta\nabla(\theta)$;
    end for
end while

4.2 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?

4.2.1 Datasets

We compare the performance of `EMBA` with several existing baseline methods on four entity matching benchmark datasets. The statistics pertaining to the training and testing sets are provided in Table 4.1.

WDC datasets. The WDC Product Data Corpus for Large-scale Product Matching [64], was built by extracting product offers from the Common Crawl. The WDC datasets serve as a popular entity benchmark dataset, 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 [89]. It 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 around 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. As such, we limit the number of words used for each attribute to meet the 512-token maximum length limit for BERT-based transformer models by only keeping at most twice the median length of the attribute value.

Other structured and textual datasets. We also compare the models using the abt-buy, dblp-scholar, and company entity matching benchmark datasets. The same preprocessed splits as the JointBERT and DeepMatcher evaluation settings are used. For these three datasets, each dataset represents a match between two mostly deduplicated datasets for different domains, namely products (abt-buy), scientific texts (dblp-scholar) and companies. Since the abt-buy and companies datasets do not contain multiple entity descriptions for many of the described entities, the results illustrate how `EMBA` performs in these settings.

4.2.2 Baseline Models

`EMBA` is evaluated against six baseline models. This section summarizes each of the models, along with their specific training settings. The models are trained three times and we report the average of the F1 score for the positive class. For all but JointBERT, we present the best result from either the JointMatcher or JointBERT paper [62, 85]. This means that for five of the six models, we will not report their standard deviation with their average.

- **DeepMatcher** [56]: An entity matching solution that customizes the recurrent

Table 4.1: Statistics about the datasets

Dataset	Size	# Pos. Pairs	# Neg. Pairs	Test Set	# Entities
WDC computers	xlarge	9690	58771	1100	745
	large	6146	27213		
	medium	1762	6332		
	small	722	2112		
WDC cameras	xlarge	7178	35099	1100	562
	large	3843	16193		
	medium	1108	4147		
	small	486	1400		
WDC watches	xlarge	9264	52305	1100	615
	large	5163	21864		
	medium	1418	4995		
	small	580	1675		
WDC shoes	xlarge	4141	38288	1100	562
	large	3482	19507		
	medium	1214	4591		
	small	530	1533		
abt-buy	default	822	6837	1916	819
dblp-scholar	default	4277	18688	5742	1635
company	default	22560	67569	22503	5640

neural network architecture to aggregate the attribute values and then compares the aggregated representations of attribute values. According to the paper, 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.

- **DITTO** [52]: A state-of-the-art entity matching model that cast the problem as a sequence-pair classification and fine-tunes RoBERTa, a pre-trained Transformer-based language model [54]. We report the results from [62] 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

Table 4.2: Comparison of F1 on the test sets for the different datasets. The best performance is bolded and the second best performance underlined.

Dataset	Size	Deepmatcher	BERT	RoBERTa	DITTO	JointMatcher	JointBERT	EMBA
WDC computers	xlarge	88.95	94.57	94.73	96.53	95.73	96.37(±0.97)	**99.03**(±0.23)
	large	84.32	92.11	94.68	93.81	94.03	94.81(±1.69)	**97.96**(±0.18)
	medium	69.85	89.31	91.90	88.97	90.10	86.55(±0.91)	**93.06**(±0.35)
	small	61.22	80.46	86.37	81.52	**86.95**	76.15(±0.99)	83.15(±0.75)
WDC cameras	xlarge	84.88	91.42	94.39	94.74	93.57	96.34(±1.99)	**99.33**(±0.42)
	large	82.16	91.02	93.91	94.41	92.00	93.55(±0.78)	**97.84**(±0.02)
	medium	69.34	87.02	90.20	87.97	89.26	85.36(±2.01)	**91.88**(±0.79)
	small	59.65	77.47	**85.74**	78.67	84.15	77.33(±0.84)	80.98(±0.99)
WDC watches	xlarge	88.34	95.76	94.87	97.05	96.61	96.99(±1.29)	**99.18**(±0.17)
	large	86.03	95.23	93.93	97.17	95.89	96.66(±2.09)	**99.05**(±0.12)
	medium	67.92	89.00	92.28	89.16	93.18	85.66(±2.09)	**93.80**(±0.12)
	small	54.97	78.73	87.16	81.32	**91.31**	74.16(±2.78)	83.91(±0.16)
WDC shoes	xlarge	86.74	87.44	88.88	93.28	90.22	95.49(±3.60)	**98.72**(±0.25)
	large	83.17	87.37	86.60	90.07	89.01	92.40(±3.14)	**97.83**(±0.08)
	medium	74.40	79.82	81.12	83.20	85.63	78.73(±1.63)	**88.65**(±0.22)
	small	64.71	74.49	**80.29**	75.13	78.42	68.84(±1.96)	74.79(±2.66)
abt-buy	default	62.80	84.64	**91.05**	82.11	-	82.76(±0.28)	85.42(±0.82)
dblp-scholar	default	94.70	95.27	**95.29**	94.47	-	94.12(±0.21)	94.83(±0.09)
company	default	92.70	91.70	91.81	90.68	-	91.39(±0.48)	**92.73**(±0.54)

batch size to 8 due to memory constraints with warmup.

- **BERT-based Models**: Both uncased BERT and RoBERTa models are presented as in [62]. 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 models use the full input length of 512 tokens.

- **JointBERT** [62]: It is a dual-objective training method for BERT, which 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.

- **JointMatcher** [85]: It is a novel entity matching 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. As such, it does not need to 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.

We train EMBA and JointBERT 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. All models are allowed 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]. Also, we apply the early stopping strategy if a model performance on the validation set does not increase over 10 consecutive epochs. Both models are trained three times and we report the average performance with its standard deviation.

4.3 Predictive Performance

Table 4.2 summarizes the F1 results of the experiments across all models and datasets. With regards to the WDC datasets, EMBA achieves the best performance except for the small training size setting where JointMatcher and RoBERTa achieve a higher F1 score. It 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.

The results also illustrate that EMBA achieves the best performance on the company dataset and the second-highest performance on the abt-buy dataset. From the results, we observe that for smaller datasets (abt-buy, dblp-scholar, and the small training size for WDC datasets), RoBERTa can obtain a better result than EMBA. Since RoBERTa pre-trains on a larger corpus, the fine-tuning process is less likely to overfit on the small dataset. Yet once there are sufficient samples for fine-tuning, the effect of pre-

Table 4.3: The Entity ID prediction results on WDC Cameras datasets, where #1 is the first entity ID prediction task, and #2 is second entity ID prediction task.

	JointBERT			EMBA		
	#1 Accuracy	#2 Accuracy	Overall F1	#1 Accuracy	#2 Accuracy	Overall F1
xlarge	0.98	0.98	0.98	**1**	0.98	**0.99**
large	0.98	0.97	0.98	**0.99**	**0.98**	**0.99**
medium	0.93	0.93	0.93	**0.97**	**0.94**	**0.96**
small	0.11	0.19	0.33	**0.71**	**0.6**	**0.63**

training on a large corpus is mitigated, as can be seen from the shrinking performance gap between BERT and RoBERTa. In the future work, we will explore the reason that RoBERTa failed on these large datasets.

JointMatcher also exhibits a similar trend as RoBERTa, where it can yield better performance than EMBA for the smaller WDC training sizes. We posit that since JointMatcher utilizes the pre-trained RoBERTa embeddings, this gives it a performance boost over the BERT-based models (BERT, DITTO, JointBERT, and EMBA). We note that half the time, JointMatcher outperforms RoBERTa, which suggests that the relevance-aware encoder and numerically-aware encoder may have some potential impact but only for specific datasets. We also note that JointMatcher does not provide a consistent improvement over the single objective models, especially when compared to DITTO.

In comparison with JointBERT, EMBA can improve the performance up to 8%. Moreover, there is no setting where JointBERT offers better performance than our model across the four different datasets. 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.

4.3.1 Auxiliary Tasks Analysis

Since JointBERT and `EMBA` are dual-objective models, this paper will explore their performance on the auxiliary tasks. We retrieved the WDC cameras results from both models as shown in Table 4.3. `EMBA` outperforms JointBERT over all datasets. It shows that using different representations of the entities instead of [CLS] token for all tasks could improve the prediction performance. When we focus on small dataset, `EMBA` improves the results at most 60% comparing with JointBERT, which states the effectiveness of token utilization. However, it should be noticed that, even though the entity matching is the main task, the two auxiliary subtasks would weigh more than the entity matching task. In the future, we would develop other subtasks to serve the main task better.

4.3.2 Statistics Analysis

We conduct an analysis to determine whether `EMBA` provides a statistically significant improvement over JointBERT and assess the stability of the models. To do this, we perform two additional training sessions for both models and use the one-tailed t-test on the resulting five F1 scores from each model on the four different datasets. The null hypothesis (H_0) and alternative (H_a) hypotheses are as follows:

$$H_0 : \mu_{\texttt{EMBA}} \leq \mu_{JointBERT}$$

$$H_a : \mu_{\texttt{EMBA}} > \mu_{JointBERT}$$

Figure 4.3 shows the mean and standard deviation of the F1 scores for each model and the result of the t-tests. We notice that for all but *dblp-scholar*, we can reject the null hypothesis suggesting that `EMBA` provides statistically significant performance over JointBERT. For *dblp-scholar*, the null hypothesis cannot be rejected as the largest F1 score (94.36) from JointBERT is greater than the smallest of five F1 scores (94.31)

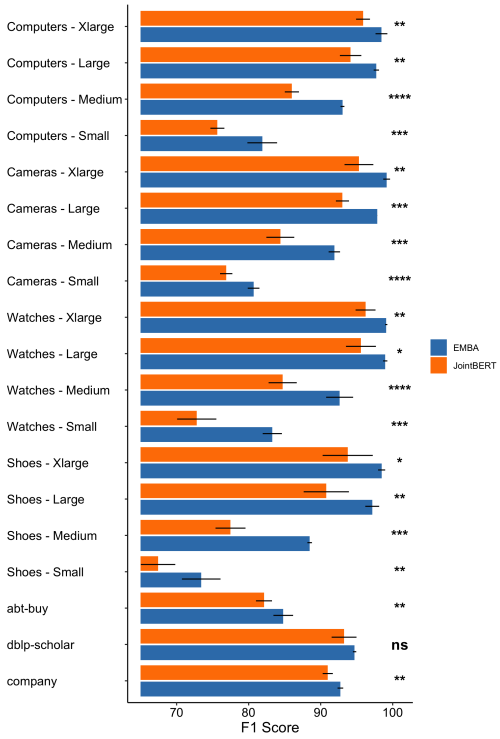


Figure 4.3: Statistical significance analysis of the F1 performance between EMBA and JointBERT. The mean and standard deviation (error bars) are shown, as well as the result of the t-test. * denotes if $p < 0.05$, ** if $p < 0.01$, *** if $p < 0.001$, **** if $p < 0.0001$, and ns if $p \geq 0.05$.

from EMBA.

The figure also illustrates the stability of EMBA. As the WDC training size increases, we can observe that there is less variation in the performance of EMBA. 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.

4.4 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

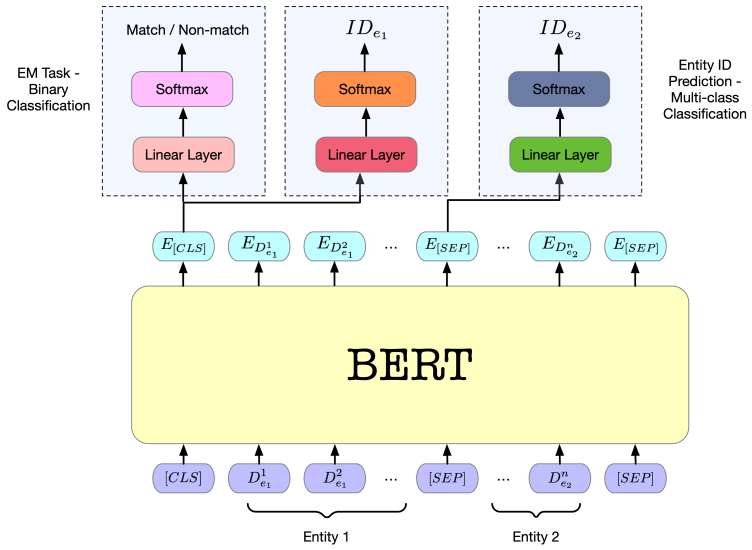


Figure 4.4: JointBERT-S where the [SEP] token is used for the second entity identifier prediction task and the [CLS] token is used for the binary classification and first entity identifier prediction.

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 4.4. 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

Table 4.4: Results for ablation experiments on F1. 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	97.49	<u>98.83</u>	97.49	97.65	97.48	96.86	**99.03**
	large	96.90	<u>97.83</u>	96.68	97.50	95.52	97.33	**97.96**
	medium	88.82	<u>92.33</u>	89.86	90.65	89.48	89.34	**93.06**
	small	77.55	<u>81.74</u>	76.47	80.18	77.31	67.52	**83.15**
WDC cameras	xlarge	98.02	98.32	98.00	<u>99.01</u>	98.19	98.60	**99.33**
	large	96.51	<u>97.66</u>	95.44	97.04	96.03	97.34	**97.84**
	medium	87.91	<u>91.13</u>	86.46	88.44	86.11	84.07	**91.88**
	small	78.30	<u>80.24</u>	74.66	75.80	78.12	57.92	**80.98**
WDC watches	xlarge	97.09	98.32	98.35	<u>98.84</u>	98.01	97.79	**99.18**
	large	98.46	<u>98.84</u>	97.87	98.33	98.02	97.84	**99.05**
	medium	87.46	<u>93.23</u>	89.03	91.22	87.44	84.42	**93.80**
	small	75.83	<u>83.77</u>	75.10	79.65	79.37	57.38	**83.91**
WDC shoes	xlarge	97.88	<u>98.67</u>	97.81	97.99	96.99	97.46	**98.72**
	large	95.16	97.50	<u>97.84</u>	96.88	96.11	93.07	**97.83**
	medium	82.61	85.67	80.65	<u>87.50</u>	81.63	71.74	**88.65**
	small	73.13	<u>73.73</u>	68.89	69.94	71.64	57.20	**74.79**
abt-buy	default	83.44	<u>85.17</u>	81.35	81.72	83.29	79.86	**85.42**
dblp-scholar	default	93.99	<u>94.58</u>	94.40	93.17	94.13	94.01	**94.83**
company	default	91.40	<u>91.94</u>	91.54	91.15	89.17	90.69	**92.73**

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 [81] (**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.

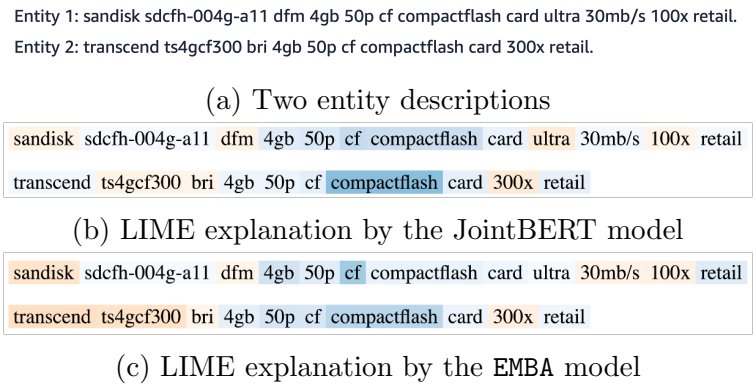
The results of the ablation study are summarized in Table 4.4. Unsurprisingly, `EMBA` model outperforms the other models, suggesting that all the components are needed for better matching performance. We can observe that simply swapping the representation to the [SEP] token for the second entity ID prediction 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 embedding.

The ablation study results also highlight the importance of using the individual token representations. Even using a simple average of the tokens (JointBERT-T) does not hinder the performance and even in some cases provides a slight benefit. 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 other multi-class classification tasks within these models (i.e., entity ID prediction). When using [SEP] token for second entity ID prediction, or averaging each entity tokens 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% comparing with JointBERT. We can also find details in the case study section.

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



(a) Two entity descriptions

(b) LIME explanation by the JointBERT model

(c) LIME explanation by the EMBA model

Figure 4.5: LIME explanations for a non-match classified incorrectly by the JointBERT and correctly by the EMBA.

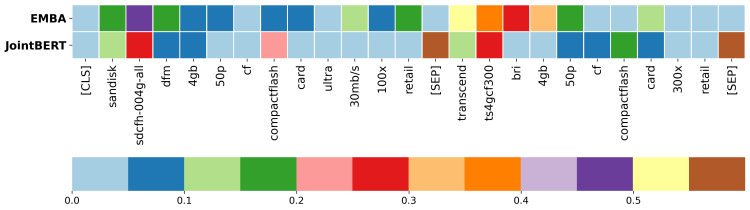


Figure 4.6: Attention visualization of an entity pair

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 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 4.5a. 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.

We first analyze the word importance using the same methodology used for JointBERT [62]. In particular, we utilize the Mojito framework [19] which is based on the LIME algorithm [66] and has been used to explain deep matching decisions [62]. 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 4.5 illustrates the LIME explanations generated with Mojito for a matching decision by JointBERT (see Figure 4.5b) and by `EMBA` (see Figure 4.5c). 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 highlights 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.

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 4.6 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 [82]. 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. However, `EMBA` feeds the token representations rather than special tokens to the tasks, and it can obtain the appropriate feedback from different tasks to optimize the attention weights. Therefore, the attention could focus on the crucial tokens such as the brand names and model numbers, so that it could 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.

Chapter 5

Cross-Attention Multi-task Learning for Schema and Entity Matching

Integrating data from multiple sources requires a common understanding of the underlying schema and the entities represented within each dataset. Both schema and entity matching are increasingly being performed using DL techniques. These approaches have shown promising results in improving the accuracy and efficiency of matching algorithms. Despite the fact that schema matching and entity matching are related, they involve different levels of abstraction and require different matching algorithms and techniques. Separating these tasks can allow specialized techniques and algorithms that are better suited to each task's requirements, leading to improved accuracy and performance. Unfortunately, serializing the two stages can result in error propagation and amplification.

There are three key underlying assumptions of existing entity matching studies: (1) the entity descriptions share a common schema, (2) the matching attributes in the datasets are positioned in the same relative order by the construction of the entity

Table 5.1: Preliminary results for shuffling the order of attribute values

Models	Attributes aligned	Computers	Cameras	Watches	Shoes	abt-buy	dblp-scholar
BERT	Yes	**80.46**	**77.47**	**78.73**	**74.49**	**84.64**	**95.27**
	No	73.69	70.37	65.86	61.45	78.58	81.79
JointBERT	Yes	**76.15**	**77.33**	**74.16**	**68.84**	**82.76**	**94.12**
	No	71.72	71.83	66.99	62.36	81.03	87.20

pairs [61], and (3) the values have been correctly entered (i.e., not misaligned where the value is accidentally placed with the previous attribute) nor missing. However, such methods are not applicable in practical scenarios where attribute values and names are not fully aligned. The first two assumptions assume that schema matching has been done properly without any mistakes, which is not practical given existing automated schema matching models as discussed in Chapter 3.

We illustrate the performance degradation under a simple permutation of the attributes. For example, the first entity is "*brand* + *title* + *description*" and the second entity is "*title* + *description* + *brand*". As shown in Table 5.1, the performance of BERT and JointBERT is reduced by at most 13% and 7%, respectively. This demonstrates the potential limitations of existing BERT-based models that rely on perfect attribute alignment to achieve reasonable performance.

Misalignment of the attribute values can also affect entity matching models that serialize the record pair with special tokens for splitting the column name and its content [52, 85]. For example, the restaurant address can be mistakenly placed with the business title. Thus, even preserving the relative order of the attribute across different tables can cause unexpected performance problems or require a significant amount of manual labor to properly align the values.

To address the above limitations and provide a dataset-invariant approach, we propose a cross attention-based model, `CaSE`, to integrate both the attribute name and values into the modeling process. Cross attention [91] is a type of attention mechanism that allows models to quickly and accurately switch their focus between different viewpoints or perspectives. This ability is motivated by the human ability

to effectively navigate and interact with our environment, such as driving a car or playing a sport. Existing research has shown that cross-view attention is associated with increased cognitive flexibility and adaptability, as well as improved social skills and decision making abilities. Furthermore, deficits in cross-view attention have been linked to a range of mental health disorders, such as attention deficit hyperactivity disorder (ADHD) and schizophrenia. Overall, cross-attention is an important cognitive function that plays a crucial role in our daily lives. We introduce it to the entity matching task to appropriately learn the interaction between the attribute name and its contents. We also introduce a new multi-task learning objective that combines entity matching and schema matching. By learning the corresponding attributes across schemas, our model does not require perfect alignment of the attributes to perform entity matching. Similarly, by leveraging the attribute values, the schema-matching task can utilize the data distribution to better identify attribute correspondences.

Since we are the first to propose to tackle both matching tasks joint, we curate a new benchmark dataset that combines both schema matching and entity matching tasks. Our benchmark extends 8 existing schema matching or entity matching datasets by manually labeling the instances for the other missing task (e.g., entity matches for a schema matching dataset). In addition, we also create 4 versions of the dataset to reflect real-world scenarios where the various key assumptions are violated. Figure 5.1 provides an example comparison between the curated and aligned existing benchmark dataset and the original, unaligned, and misaligned data. Our experimental results on the new benchmark illustrate that `CaSE` generally outperforms the single-task objective models with improvements ranging from 1-13%.

Current entity matching benchmark datasets

Dataset	NAME	RATING	PHONENUMBER	NO_OF_REVIEWS	ADDRESS
Zomato	2 Asian Brothers	3.1	(773) 681-0268	12	3222 W. Foster Avenue, Chicago, IL
Yelp	2 White Crew Cleaning	4	(773) 634-0830	26	3038 N Honore St, Chicago, IL 60657

(a) Current entity matching benchmark examples

Real-world datasets - Zomato

Dataset	Title	Average_rating	Phone Number	User_reviews	streetAddress
Zomato	2 Asian Brothers - North Park, Chicago	Rated 3.1/5	(773) 681-0268	Based on 12 votes	3222 W. Foster Avenue, Chicago

Real-world datasets - Yelp

Dataset	Business Title	Rating Value	telephone	reviewCount	Address
Yelp	2 White Crew Cleaning - Roscoe Village - Chicago, IL	4.0	(773) 681-0268	26	3038 N Honore St, Chicago, IL 60657

(b) Real-world dataset examples

Figure 5.1: Examples for current entity matching benchmark dataset and real-world datasets

5.1 Approach

5.1.1 Problem Definition

Given a source table and target table, we extract their respective column names as $D_{c_1} = \{D^1_{c_1}, D^2_{c_1}, \cdots, D^m_{c_1}\}$ and $D_{c_2} = \{D^1_{c_2}, D^2_{c_2}, \cdots, D^n_{c_2}\}$ (e.g., name, rating, and phonenumber in Figure 5.1a). We also extract the entity descriptions associated with the respective tables $D_{e_1} = \{D^1_{e_1}, D^2_{e_1}, \cdots, D^m_{e_1}\}$, 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 attribute values (e.g., 2 Asian Brothers, 3.1, (773) 681-0268). The goal is to learn (1) whether the two entities refer to the same object (i.e., entity matching task) based on the descriptions (i.e., D_{e_1} and D_{e_2}) and (2) which columns in the source table are related to those in the target table.

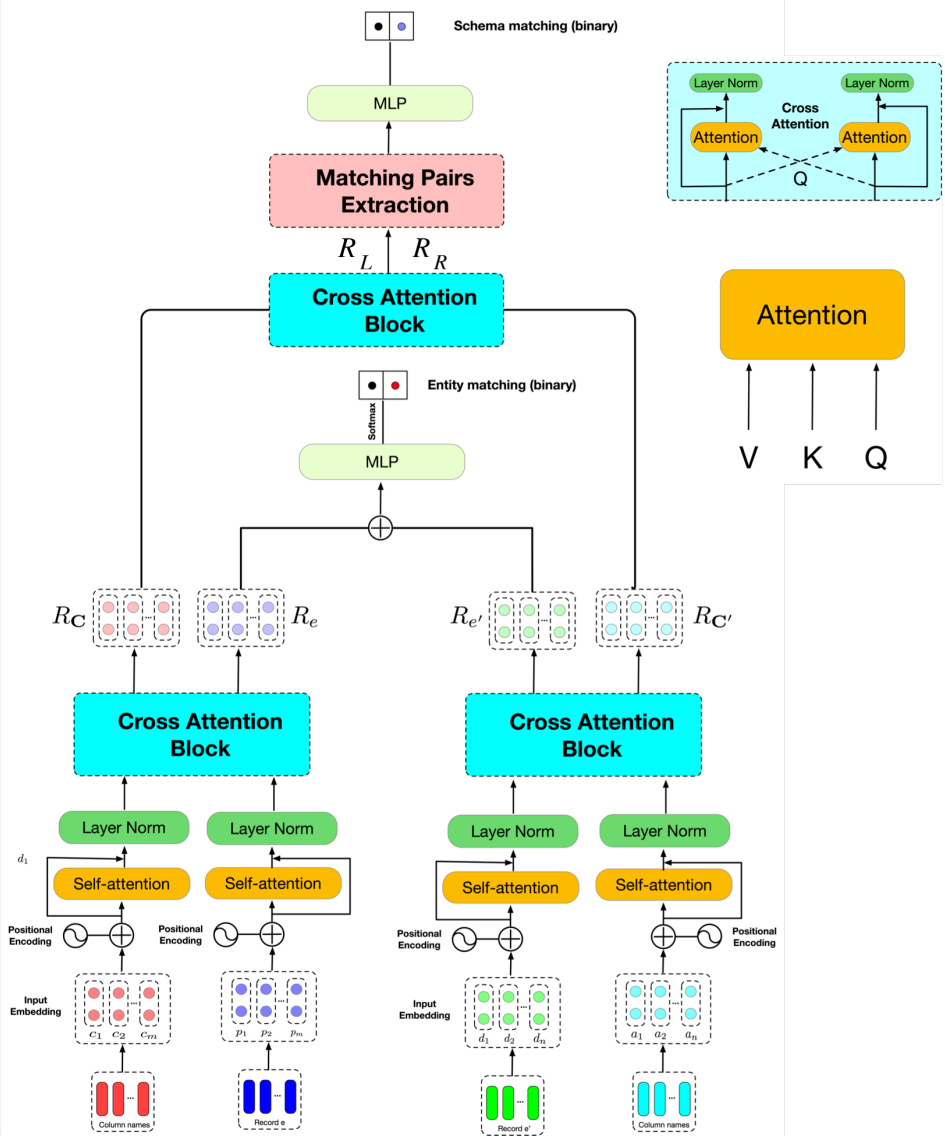


Figure 5.2: Illustrations for the multi-task learning with both schema matching and entity matching.

5.1.2 Overview

We propose a new cross attention-based framework `CaSE` that simultaneously solves both the entity matching and schema matching problem by incorporating a multi-task learning objective. The idea is for the model to learn both the interactions between the attribute name and value and between the two different schemas. In this manner, the model avoids the need to pre-train special tokens to separate the column name and value (e.g., DITTO) as well as the manual labor associated with aligning the attributes between two schemas.

As shown in Figure 5.2, we feed four kinds of inputs to `CaSE`.

1. The source table column names sequences I_{c_1} = [CLS] $D^1_{c_1}$ [SEP] $D^2_{c_1}$ [SEP] $\cdots$ [SEP] $D^m_{c_1}$ [SEP].
2. An entity sequence from the source table I_{e_1} = [CLS] $D^1_{e_1}$ [SEP] $D^2_{e_1}$ [SEP] $\cdots$ [SEP] $D^m_{c_1}$ [SEP].
3. The target table column names sequences I_{c_2} = [CLS] $D^1_{c_2}$ [SEP] $D^2_{c_2}$ [SEP] $\cdots$ [SEP] $D^n_{c_2}$ [SEP].
4. An entity sequence from the target table I_{e_2} = [CLS] $D^1_{e_2}$ [SEP] $D^2_{e_2}$ [SEP] $\cdots$ [SEP] $D^n_{c_2}$ [SEP].

The four sequences are then embedded using a (frozen) BERT layer to obtain $E_c = [c_1, c_2, \cdots, c_m]$, $E_e = [p_1, p_2, \cdots, p_m]$, $E_{c'} = [d_1, d_2, \cdots, d_n]$, and $E_{e'} = [a_1, a_2, \cdots, a_n]$. These embeddings are fed into the self-attention layer separately and then go to the cross attention blocks. The self-attention layer follows the function as follows,

$$f_{selfAtt}(H) = softmax(\frac{q(H)k(H)^T}{\sqrt{d_k}})v(H) \tag{5.1}$$

where H is the encoded features from previous layer. $q(\cdot)$, $k(\cdot)$, and $v(\cdot)$ are the query, key, and values respectively. d_k is the number of attention heads for normalization.

The cross attention block contains two attention modules. Both encoded features (H_c and H_e) will feed to this block, and follow the calculations below,

$$f_{crossAtt1} = softmax(\frac{q(H_c)k(H_e)^T}{\sqrt{d_k}})v(H_e) \tag{5.2}$$

$$f_{crossAtt2} = softmax(\frac{q(H_e)k(H_c)^T}{\sqrt{d_k}})v(H_c) \tag{5.3}$$

where $f_{crossAtt1}$ serves as the operation to discover inner-relationships from column names to the entity attributes, and $f_{crossAtt2}$ captures the alignment between entity attributes and column names. The numbers of columns and entity attributes are identically ordered in our inputs. Based on this, the contextual clues can be propagated between the columns and the entity attributes, for example, the column name can be enhanced by the entity attribute.

For entity matching, we concatenate the outputs from the two cross attention blocks associated with the entity (i.e., R_e and $R_{e'}$), and feed it to the multi-layer perceptron (MLP) for the classification task. For schema matching task, we want to further capture the interactions of columns between source and target tables, so we feed the source table column representation R_C and target table column representation $R_{C'}$ to a new cross attention block. Since there are m and n columns in the source and target tables respectively, there are $m \times n$ schema matching pairs. The matching pair extraction block will extract the corresponding column name representations from the outputs (i.e., R_L, R_R) of cross attention block. Specifically, if the column name contains more than one token, we will average them as the representation of the column name. Then we feed them into the MLP layer to do the schema matching task. Since each entity pair will have the same schema matching task, we will utilize majority voting at the inference stage.

Table 5.2: Overview of the datasets within our new benchmark dataset.

Datasets	# of Tables	# of Records		# of Matches	# of Non-matches	# of Attributes	# of schema matching pairs
		L	R				
Restaurants	2	533	331	130	270	5	25
Walmart-Amazon	2	2,554	22,074	1154	-	10	100
Baby Products	2	5,085	10,718	108	292	16	256
Bikes	2	4,785	9,002	130	320	8	64
Books	2	396	3,700	92	305	10	100
Phones	17	447	50	258	22,092	26	676
Headphones	6	444	51	226	22,418	27	729
TVs	8	428	60	182	25,499	61	3721

5.1.3 Dual Objective Training

`CaSE` uses the binary cross-entropy loss (BCEL) for the entity matching task and the binary hinge loss (BHL) for the schema matching task. Let $y_{em_i}, y_{s_{ij}}$ denote the entity matching label and the schema matching label, respectively. The loss is then defined as follows,

$$L_i = BCEL(y_{em_i}, \hat{y}_{em_i}) + \sum_{j}^{t} BHL(y_{s_{ij}}, \hat{y}_{s_{ij}}) \tag{5.4}$$

$$= -[y_{em_i} \log(\hat{y}_{em_i}) + (1 - y_{em_i}) \log(1 - \hat{y}_{em_i})] + \sum_{j}^{t} \max(0, 1 - y_{s_{ij}} \hat{y}_{s_{ij}}) \tag{5.5}$$

where i represents the entity pair, and for each i, t is the total number of the schema matching pairs.

5.2 Experiments

5.2.1 Datasets

The new benchmark dataset is obtained by extending 8 existing schema matching and entity matching datasets shown in Table 5.2. These datasets are for training and evaluating matching models for various domains including products, publications,

Table 5.3: Statistics of four types of datasets

Versions	Common schema	Same attribute order	Original values	Missingness Rate	Mis-alignment Rate
V1	Yes	Yes	No	2%	0%
V2	No	Yes	No	2%	0%
V3	No	Yes	No	35.18%	33.18%
V4	No	No	No	35.18%	33.18%

and businesses. Each dataset consists of candidate pairs from two or more structured tables of entity records of the same schema. For the existing schema matching datasets (i.e., Baby Products and Bikes), we manually annotate the entity matches. For existing entity matching datasets (the other 6 datasets), we manually add new schema matching labeled instances to indicate whether two corresponding attributes across the different tables are the same.

In general, the construction of entity matching benchmark datasets is time-consuming and labor-intensive [48]. Usually, these are curated by first building a global schema to unify the incoming data from different sources, which will return a clean and structural dataset. Figure 5.1 shows an entity matching example in the restaurant dataset. As can be observed, the attribute names and values are different between the benchmark dataset and the real-world dataset (i.e., what is found in the original XML files). Therefore, we propose four versions of the datasets as shown in Table 5.3. The first version consists of carefully curated entity matching benchmark datasets. For the second version (V2), the column names of source and target table reflect the names in the original files, which means the columns are no longer the same (highlighted in blue). The third version (V3) more closely mimics the real-world datasets where both the column names and the attribute values follow the original files. The only exception is that the relative order of the attributes is the same. The last version (V4) reflects the ordering of the original files, where schema matching was done beforehand to align the attribute order. The four versions of data examples can be seen in Figure 5.3.

V1 datasets

Dataset	NAME	RATING	PHONENUMBER	NO_OF_REVIEWS	ADDRESS
Zomato	Bonfyre American Grille	3.9	(608) 273-3973	573	2601 West Beltline Highway, Madison, WI

Dataset	NAME	RATING	PHONENUMBER	NO_OF_REVIEWS	ADDRESS	Label
Yelp	Bonfyre American Grille	3.5	(608) 273-3973	-	2601 W Beltline Hwy, Madison, WI 53713	1

V2 datasets

Dataset	Title	Average_rating	Phone Number	User_reviews	streetAddress
Zomato	Bonfyre American Grille	3.9	(608) 273-3973	573	2601 West Beltline Highway, Madison, WI

Dataset	Business title	Rating Value	telephone	reviewCount	Address	Label
Yelp	Bonfyre American Grille	3.5	(608) 273-3973	-	2601 W Beltline Hwy, Madison, WI 53713	1

V3 datasets

Dataset	Title	Average_rating	Phone Number	User_reviews	streetAddress
Zomato	Bonfyre American Grille	Rated 3.9/5	(608) 273-3973	Based on 573 votes	2601 West Beltline Highway, Madison, WI

Dataset	Business title	Rating Value	telephone	reviewCount	Address	Label
Yelp	Bonfyre American Grille (773) 681-0268 Chicago, IL 60657	3.5		-	3038 N Honore St	1

V4 datasets

Dataset	Title	Average_rating	Phone Number	User_reviews	streetAddress
Zomato	Bonfyre American Grille	Rated 3.9/5	(608) 273-3973	Based on 573 votes	2601 West Beltline Highway, Madison, WI

Dataset	telephone	Business title	reviewCount	Address	Rating Value	Label
Yelp		Bonfyre American Grille (773) 681-0268 Chicago, IL 60657	-	3038 N Honore St	3.5	1

Figure 5.3: Four versions of the benchmark datasets.

5.2.2 Baseline models

`CaSE` is evaluated against two baseline models. This section summarizes each of the models, along with their specific training settings. The models are trained three times and we report the average of the F1 score for the positive class.

- **DITTO** [52]: A state-of-the-art entity matching model that cast the problem as a sequence-pair classification and fine-tunes RoBERTa, a pre-trained Transformer-based language model [54].
- **SMAT** [87]: A state-of-the-art schema matching model that utilizes attention-over-attention to generate a semantic embedding and then feeds the embedding to a multi-layer perceptron to conduct the classification task.[1]

We train `CaSE`, DITTO, and SMAT on a single NVIDIA Tesla V100 GPU with 16GB VRAM. For DITTO, we follow the input format where the attributes of each entity description are concatenated into a single string with [COL] and [VAL] special

[1]Code available at `https://github.com/JZCS2018/CrossAttention`

Table 5.4: Comparison of F1 on the test sets for the different datasets.

Datasets	CaSE								DITTO								SMAT							
	Entity Matching				Schema Matching				Entity Matching				Schema Matching				Entity Matching				Schema Matching			
	v1	v2	v3	v4	v1	v2	v3	v4	v1	v2	v3	v4	v1	v2	v3	v4	v1	v2	v3	v4	v1	v2	v3	v4
Restaurants	**0.87**	**0.85**	**0.85**	**0.81**	0.56	0.52	0.52	0.52	0.81	0.76	0.72	0.71	0.51	0.50	0.50	0.50	0.45	0.44	0.44	0.41	0.50	0.50	0.50	0.50
Walmart-Amazon	**0.89**	**0.88**	**0.85**	**0.82**	0.53	0.51	0.51	0.51	0.86	0.87	0.77	0.69	0.51	0.50	0.50	0.50	0.49	0.46	0.42	0.40	0.55	0.50	0.50	0.50
Baby Products	**0.81**	**0.81**	**0.79**	**0.78**	0.66	0.62	0.62	0.62	0.72	0.65	0.60	0.54	0.58	0.50	0.50	0.50	0.52	0.47	0.43	0.42	0.62	0.58	0.58	0.58
Bikes	**0.89**	**0.89**	**0.87**	**0.85**	0.59	0.51	0.51	0.51	0.77	0.70	0.64	0.58	0.53	0.50	0.50	0.50	0.50	0.46	0.40	0.40	0.58	0.52	0.52	0.52
Books	**0.84**	**0.81**	**0.8**	**0.77**	0.51	0.50	0.50	0.50	0.76	0.72	0.71	0.62	0.52	0.50	0.50	0.50	0.54	0.49	0.42	0.41	0.54	0.52	0.52	0.52
Phones	**0.76**	**0.77**	**0.77**	**0.70**	0.71	0.71	0.71	0.71	0.75	0.72	0.70	0.63	0.50	0.50	0.50	0.50	0.50	0.46	0.42	0.40	0.53	0.53	0.53	0.53
Headphones	**0.83**	**0.83**	**0.8**	**0.78**	0.69	0.67	0.67	0.67	0.80	0.78	0.71	0.66	0.50	0.50	0.50	0.50	0.51	0.49	0.45	0.42	0.52	0.56	0.56	0.56
TVs	**0.84**	**0.84**	**0.8**	**0.8**	0.75	0.70	0.70	0.70	0.76	0.72	0.61	0.53	0.67	0.59	0.59	0.59	0.49	0.45	0.42	0.40	0.69	0.66	0.66	0.66

tokens. Any further preprocessing is omitted and left to the tokenizer of the respective models. All models are allowed 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, 1e-4]. Also, we apply the early stopping strategy if a model performance on the validation set does not increase over 10 consecutive epochs. All models are trained three times and we report the average F1 performance with its standard deviation.

5.3 Predictive Performance

Table 5.4 summarizes the F1 results of the experiments across all models and datasets. With regards to the entity matching tasks on all datasets, `CaSE` achieves the best performance compared with DITTO with around 1-19% F1 score improvement. We also note that the difficulty of the entity match increases from version 1 (V1) to version 4 (V4), as the F1 performance drops for both `CaSE` and DITTO. However, the degradation for `CaSE` is less than that of DITTO. It suggests that `CaSE` has the capability to more robustly deal with real-world data where there is misalignment both in attributes and the values themselves.

We observe that `CaSE` achieves the best performance of the three models on the schema matching task. However, the schema matching results for all models are

Table 5.5: Results for ablation experiments on F1 for the entity matching task.

Datasets	CaSE				CaSE w/ BERT				CaSE w/o CA				CaSE w/o BHL			
	V1	V2	V3	V4	V1	V2	V3	V4	V1	V2	V3	V4	V1	V2	V3	V4
Restaurants	**0.87**	**0.85**	**0.85**	**0.81**	0.86	0.78	0.76	0.69	0.84	0.81	0.77	0.72	0.76	0.7	0.67	0.57
Walmart-Amazon	**0.89**	**0.88**	**0.85**	**0.82**	0.85	0.8	0.79	0.74	0.86	0.84	0.8	0.76	0.8	0.76	0.74	0.71
Baby Products	**0.81**	**0.81**	**0.79**	**0.78**	0.79	0.75	0.74	0.68	0.79	0.76	0.76	0.72	0.71	0.67	0.67	0.62
Bikes	**0.89**	**0.89**	**0.87**	**0.85**	0.87	0.82	0.81	0.77	0.87	0.83	0.8	0.77	0.72	0.7	0.66	0.61
Books	**0.84**	**0.81**	**0.8**	**0.77**	0.82	0.76	0.73	0.64	0.83	0.8	0.8	0.72	0.76	0.73	0.72	0.67
Phones	**0.76**	**0.77**	**0.77**	**0.70**	0.74	0.69	0.66	0.6	0.72	0.7	0.68	0.62	0.7	0.68	0.68	0.63
Headphones	**0.83**	**0.83**	**0.8**	**0.78**	0.8	0.79	0.76	0.7	0.8	0.79	0.79	0.75	0.77	0.72	0.69	0.64
TVs	**0.84**	**0.84**	**0.8**	**0.8**	0.8	0.78	0.78	0.72	0.82	0.81	0.8	0.77	0.78	0.76	0.72	0.7

suboptimal, as the highest F1 is 0.75 for TVs. We posit the poor performance is because of the small size of the schema matching datasets. The number of schema matching pairs in TVs is larger than others (as summarized in Table 5.2). This suggests the exploration of more datasets that contain more schema matching pairs.

5.4 Ablation Study

To gain further insights into the various components in `CaSE`, we conduct an ablation study. In particular, we examine the effectiveness and contributions of the schema matching task. We compare the performance of `CaSE` with the following models:

- `CaSE` w/o BHL: We use the single task objective in Equation (5.5) without the binary hinge loss (BHL).
- `CaSE` w/o CA: We replace the cross-attention module that feeds into the schema matching MLP with a simple concatenation of the two column representations, R_C and $R_{C'}$.
- `CaSE` w/ BERT: Instead of using the cross-attention column representations, R_C and $R_{C'}$, we directly use the BERT embeddings, E_C and $E_{C'}$.

Table 5.5 shows the results from the ablation experiments on all datasets. As we can see, for version 1 (V1) where the source and target column names are the

same, there is no big difference when simply using the BERT embeddings for the schema matching task (i.e., `CaSE` w/ BERT) as well as `CaSE` w/o CA. However, when the data is more realistic and misaligned, especially the column names in source and target tables are different, without updating the column names representations, the performance drops by at least 5%. Comparing `CaSE` with `CaSE` w/o CA, we notice that there exists small performance drops among all datasets. The differences with/without CA module is not obvious, since the numbers of schema matching pairs are small, which can not provide more feedback in the dual objective training. However, to be noticed that, these three multi-task learning methods are better than the single task approaches, such as DITTO and `CaSE` w/o BHL.

It can be observed that `CaSE` w/o BHL performs better than DITTO on V4 datasets, and it infers the cross attention mechanism is better than the self attention mechanism used in the DITTO configuration, when the orders of the attribute values are different in the entity pairs.

5.5 Case Study

To better understand the potential benefit of `CaSE` in terms of explaining the matching decision, we analyzed several examples for the different versions of the Restaurants dataset. We present three scenarios as shown in Table 5.6. The first scenarios is a related entity pair (i.e., label = 1) where `CaSE` correctly predicts the match on three data versions while DITTO predicts the match only on V1 and V3. The second is also a related entity pair (i.e., label = 1) where `CaSE` correctly predicts the match while DITTO predicts a non-match. The difference between the first scenario and second scenario is that there exists different attribute value (e.g., the phone number) in scenario 2 even though the entity pair is related. The third is a non-related entity pair, which `CaSE` predicts correctly while DITTO predicts incorrectly.

Table 5.6: Analysis on different Restaurant dataset versions, where `CaSE` makes a correct prediction and DITTO does not.

(a) Scenario 1: Two matching entities (Label = 1).

V1		**Name**	**Rating**	**PhoneNumber**	**No_of_Reviews**	**Address**
	A	The Buena Vista	3.9	(415) 474-5044	422	2765 Hyde Street, San Francisco, CA
	A′	Buena Vista Cafe	4	(415) 474-5044	1560	2765 Hyde St, San Francisco, CA 94109
V3		**Name**	**Average Rating**	**Phone Number**	**User reviews**	**Street address**
	B	The Buena Vista	3.9	(415) 474-5044	422	2765 Hyde Street, San Francisco, CA
		Business Name	**rating value**	**telephone**	**review count**	**address**
	B′	Buena Vista Cafe	4	(415) 474-5044 2765 Hyde St, San Francisco, CA 94109	1560	
V4		**Name**	**Average Rating**	**Phone Number**	**User reviews**	**Street address**
	C	The Buena Vista	3.9	(415) 474-5044	422	2765 Hyde Street, San Francisco, CA
		telephone	**Business Name**	**review count**	**address**	**rating value**
	C′	(415) 474-5044 2765 Hyde St, San Francisco, CA 94109	Buena Vista Cafe	1560		4

(b) Scenario 2: Two matching entities with difference (Label = 1).

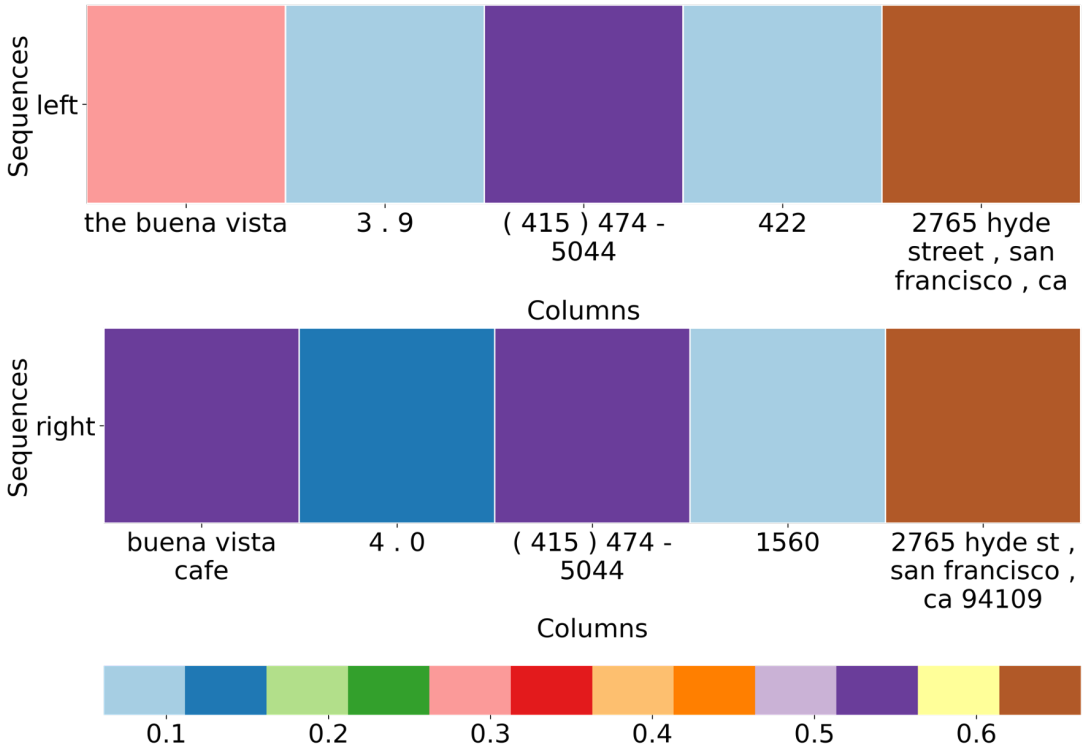
V1		**Name**	**Rating**	**PhoneNumber**	**No_of_Reviews**	**Address**
	D	The Taco Shop	3.3	(608)250-8226	37	604 University Ave, Madison, WI
	D′	The Taco Bros	4	(608)250-5075	25	604 University Ave, Madison, WI 53715
V4		**Name**	**Average Rating**	**Phone Number**	**User reviews**	**Street address**
	E	The Taco Shop	3.3	(608)250-8226	37	604 University Ave, Madison, WI
		telephone	Business Name	review count	address	rating value
	E′	(415) 474-5044 25	The Taco Bros		604 University Ave, Madison, WI 53715	4

(c) Scenario 3: Two non-matching entities (Label = 0).

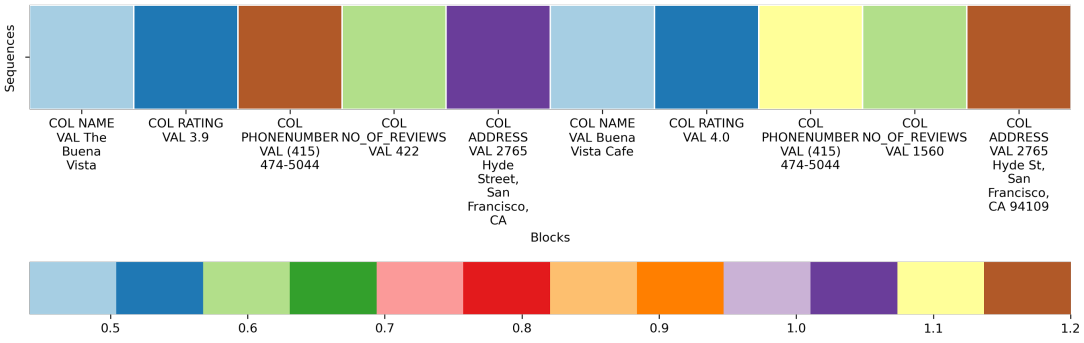
V1		**Name**	**Rating**	**PhoneNumber**	**No_of_Reviews**	**Address**
	F	Curd Girl	3	(608)555-5555	6	100 State St, Madison, WI
	F′	Naf Grill	4	(608) 256-0071	14	555 State St, Madison, WI 53703

Table 5.6a shows the matching pair examples in different data version. We know that the CaSE and DITTO all predict right for V1 and V3, even though V3 introduces the misalignment on the attribute value (e.g., the address for B′ shifts to the telephone). However, when the entity pair's attribute is different (V4), DITTO can not predict right. To understand the decision process of CaSE and DITTO, we visualize the attention weights of both models. For CaSE, we retrieve the attention weights for each entity from the cross-attention block. For DITTO, we retrieve the weights of last layer of the RoBERTa. Since DITTO utilizes the classification token $< s >$ for the entity matching downstream task, we aggregate the corresponding attention weights to the classification token according to the work [79]. Figure 5.4 and 5.5 show the attention weights of CaSE and DITTO on data V1 and V3 in scenario 1. As we can notice that, both CaSE and DITTO could capture the key information (e.g., Phone Number and Address) to make right predictions, but CaSE could also obtain the information on the store names. However, when the attribute order of both entity is different, as we can see in Figure 5.6, CaSE can capture the important information as data V1 and V3, while DITTO loses concentration on the related attributes in both entities. The DITTO takes more weights on review counts rather than the address, phone number, store name, which results in the wrong prediction.

When we focus on Scenario 2 in Table 5.6b, it shows the matching pair in two data version. And from analysis on Scenario 1, we know that the store name, phone number and address play an important role for prediction. Comparing scenrio 1, this matching pair has different attribute values in some of these important roles, for example, they are the same entity, but they have different phone numbers. CaSE predicts correctly on all data versions, while DITTO fails on these. To better understand the process of these prediction from both models, we also retrieve the attention weights in the same way. When looking at Figure 5.7, we notice that even though they have different phone numbers, CaSEcaptures other same information, such as the store name and

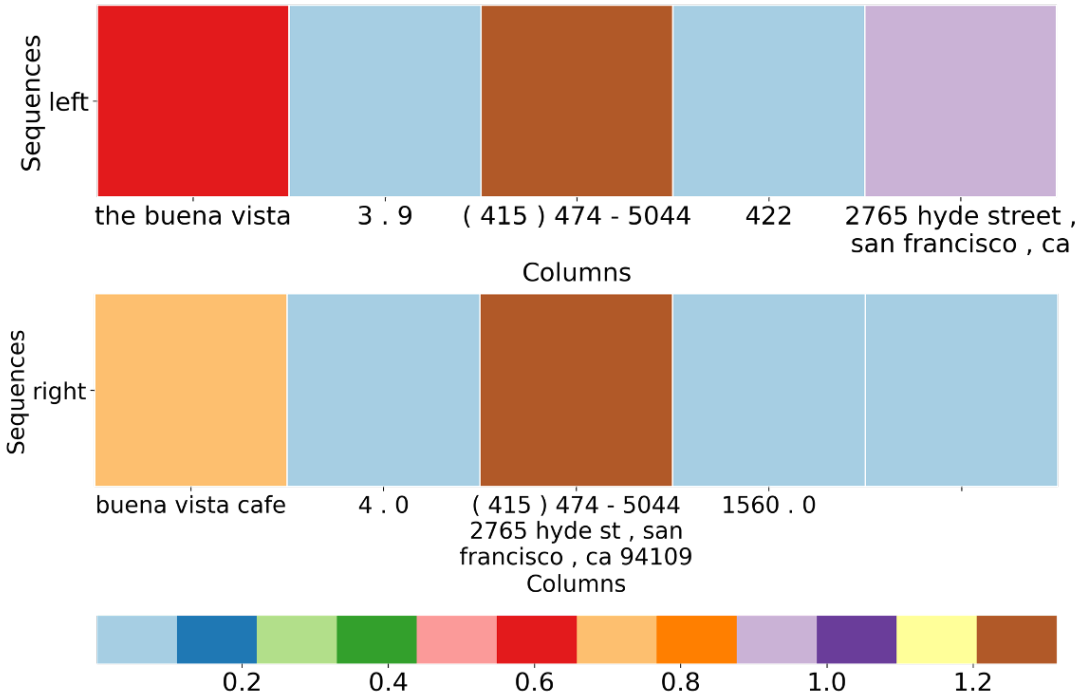


(a) Attention weights from `CaSE`

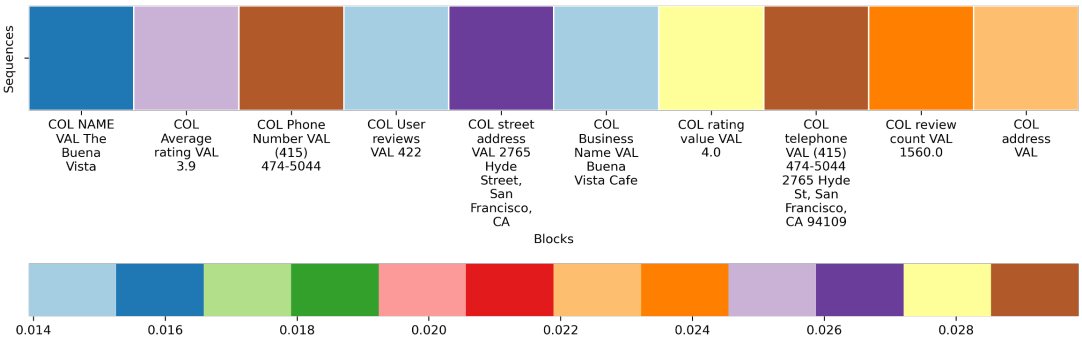


(b) Attention weights from DITTO

Figure 5.4: Attention weights visualization on data V1 in scenario 1.

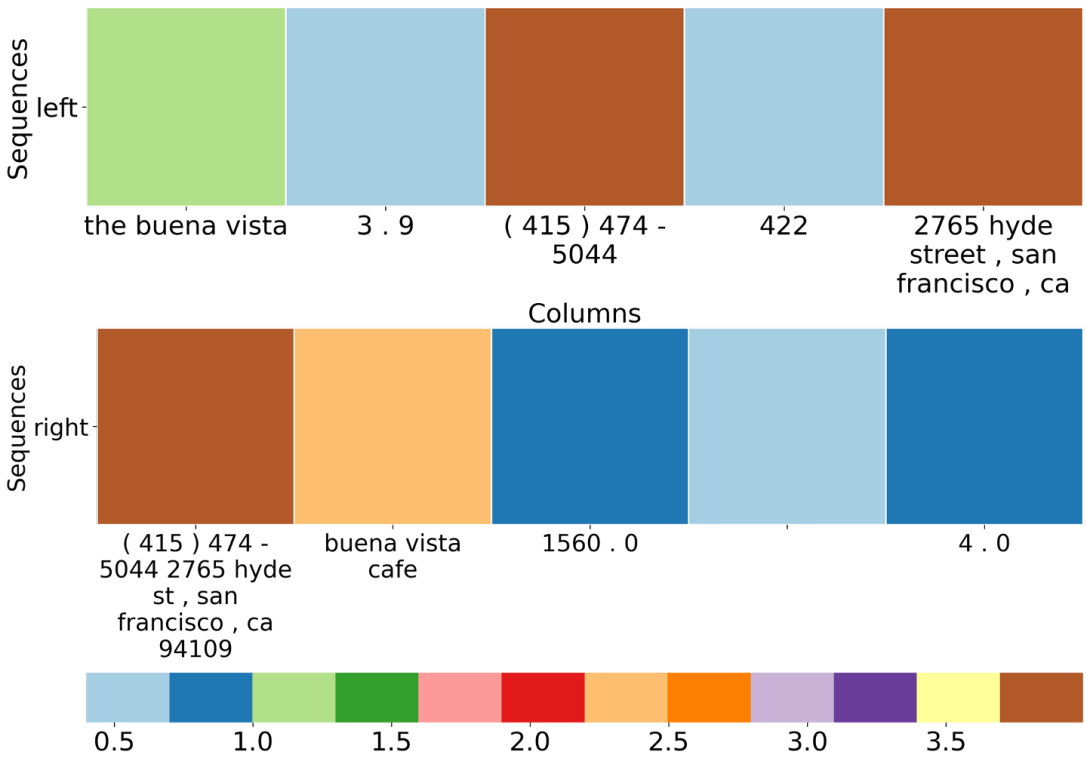


(a) Attention weights from CaSE

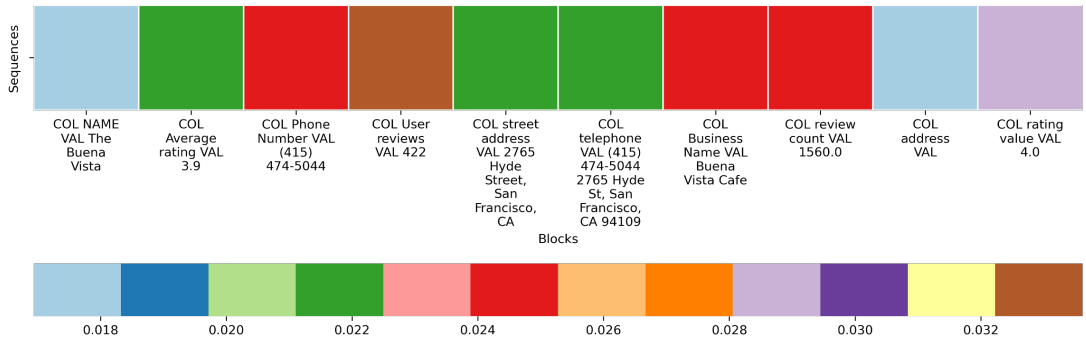


(b) Attention weights from DITTO

Figure 5.5: Attention weights visualization on data V3 in scenario 1.



(a) Attention weights from CaSE



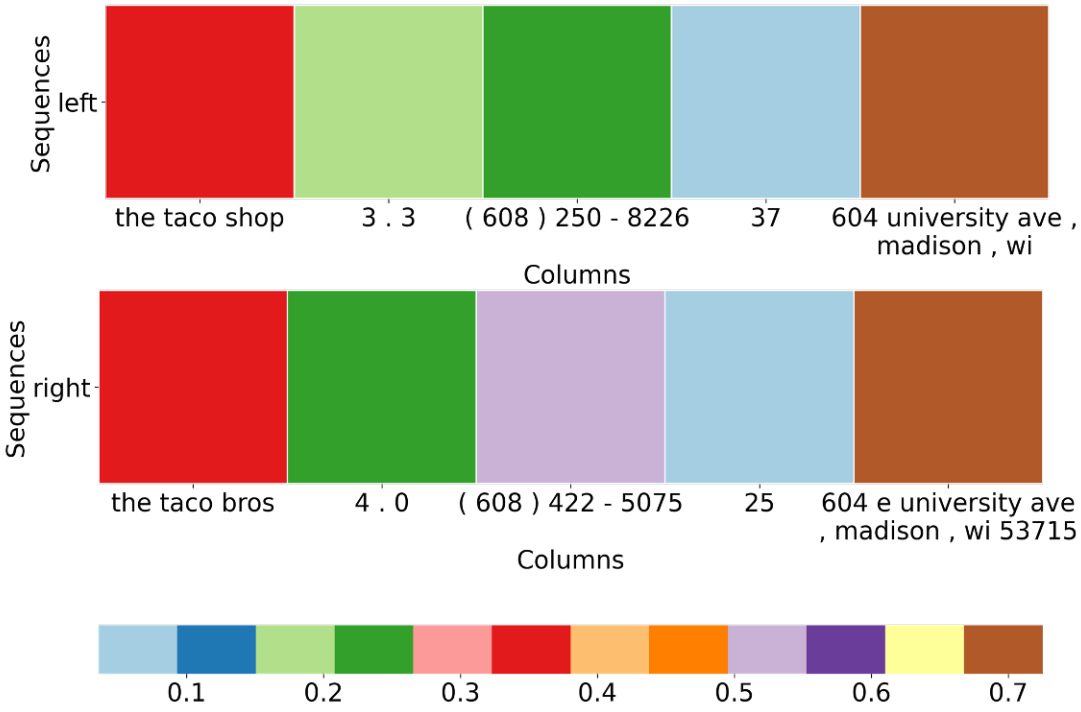
(b) Attention weights from DITTO

Figure 5.6: Attention weights visualization on data V4 in scenario 1.

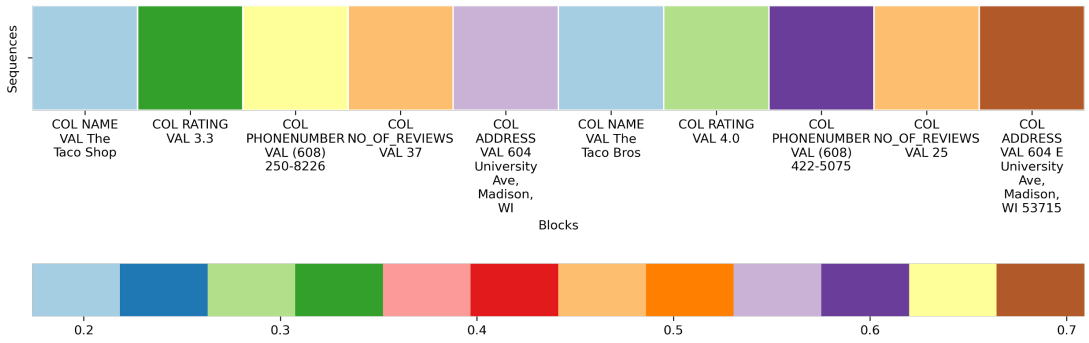
address. The analysis in Scenario 1 shows that the DITTO ignores the store name in different data visions, and it also ignores them, which results in the wrong prediction. The Figure 5.8 shows the DIITO loses focus on important attributes while `CaSE` keeps extracting the important information.

We examine the Scenario 3 in Table 5.6c, which contains the non-related entity pair. As we can see, their addresses have the same street name, city, and state name except the street number. `CaSE`could capture the same important information as other scenarios to make the right predictions. Therefore, we will explore where DITTO fails based on the attention visualization in Figure **??**. In Figure 5.10a, we can see DITTO consider the Phone number and address as the most important blocks ignoring the store names. Then we further explore the tokens DITTO chooses in the phone number and address blocks. Figure 5.10b shows the token attention weights in phone number blocks of both entities. As we can see, the [PHONENUMBER] tokens in both entities take highest weights comparing with other tokens. We know that the attribute token [PHONENUMBER] itself cannot decide if the entity pair is related or not. However, the attribute value tokens (e.g., [(608) 555-5555] and [(608) 256-0071]) can provide crucial information for prediction. We can see the same information for the address block in Figure 5.10c. The DITTO introduces the attribute names in its sequence which could hinder the key information exploration.

Finally, from the examples in Figure 5.4b, 5.5b, 5.7b, and 5.10, we notice that DITTO could extract the important blocks for the predictions. However, from Figure 5.10, we can see DITTO think these blocks are important, because of the same attribute names. As a result of this high attention weight, the entity pair could easily be considered related, but it would also introduce false positives.

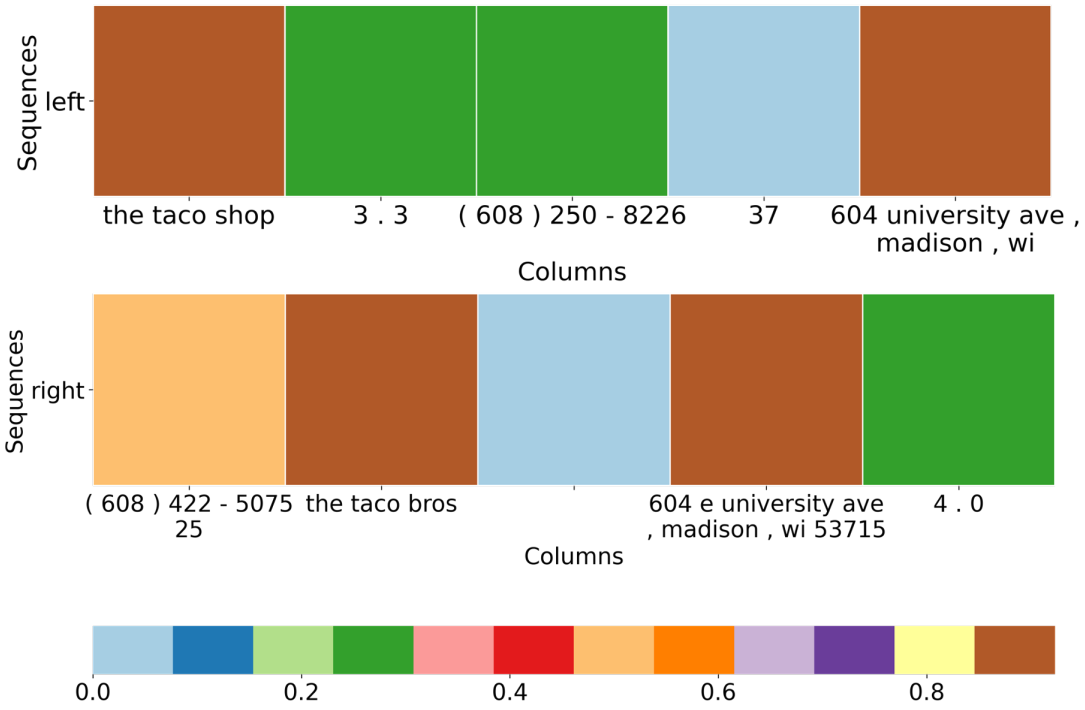


(a) Attention weights from CaSE

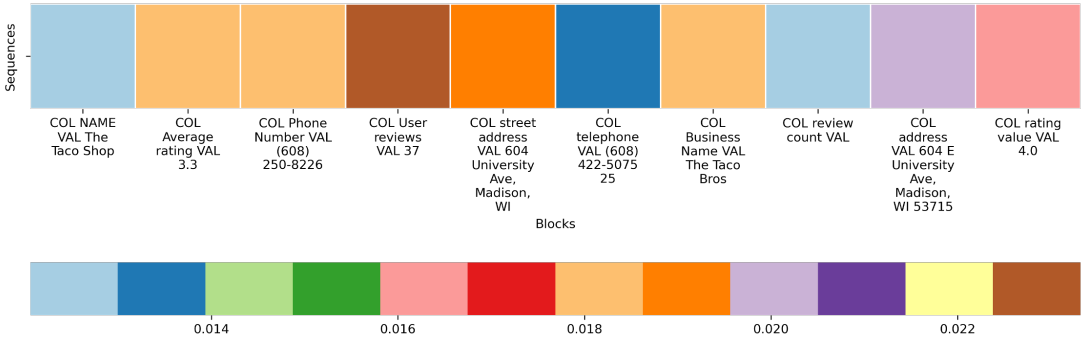


(b) Attention weights from DITTO

Figure 5.7: Attention weights visualization on data V1 in scenario 2.

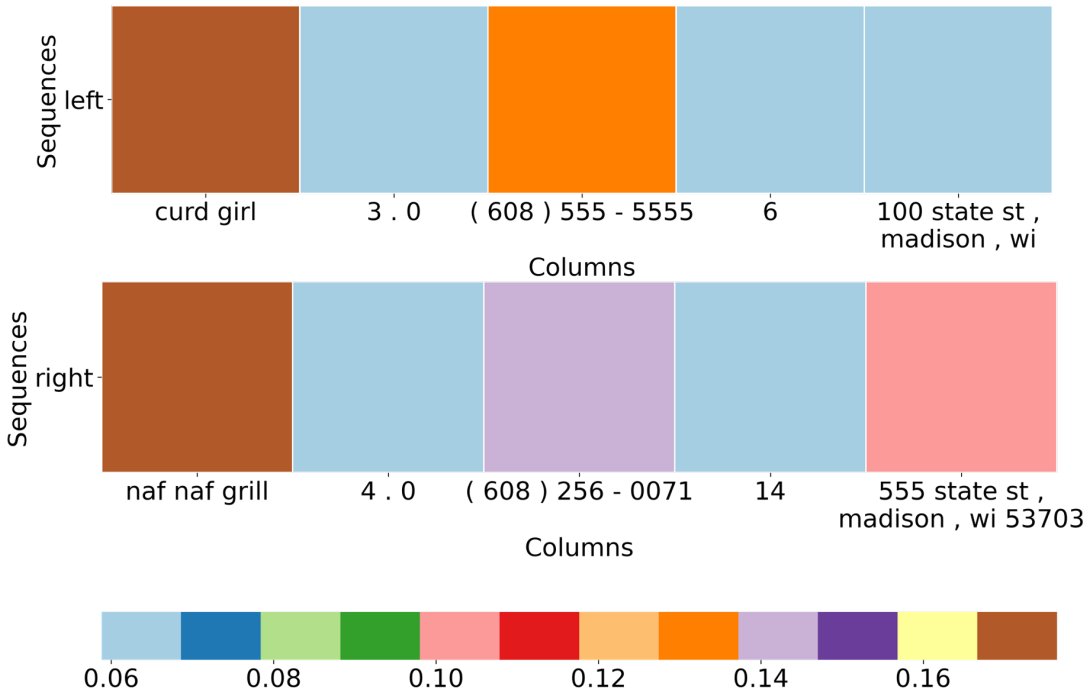


(a) Attention weights from CaSE

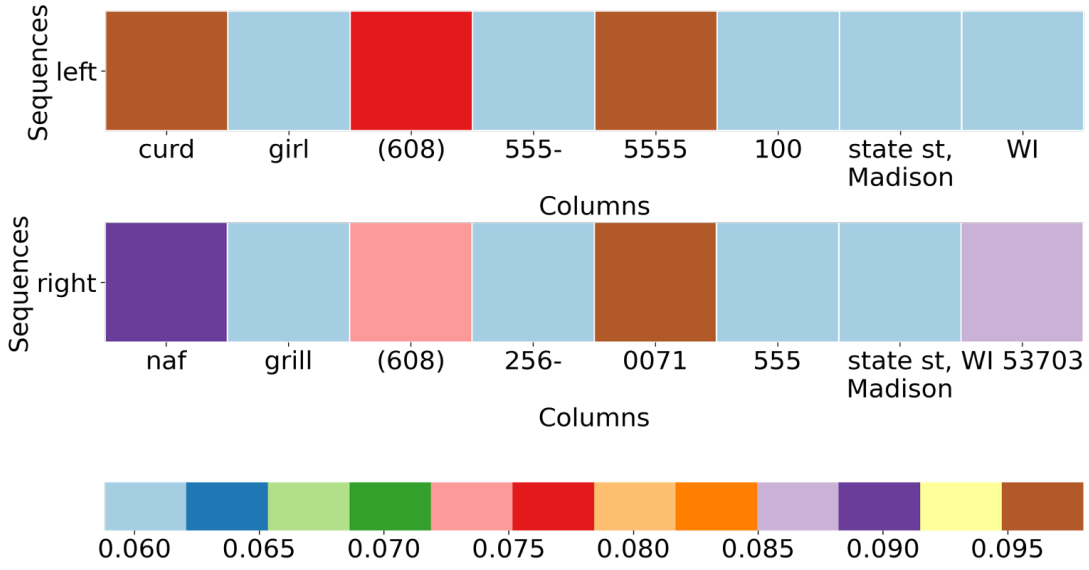


(b) Attention weights from DITTO

Figure 5.8: Attention weights visualization on data V4 in scenario 2.

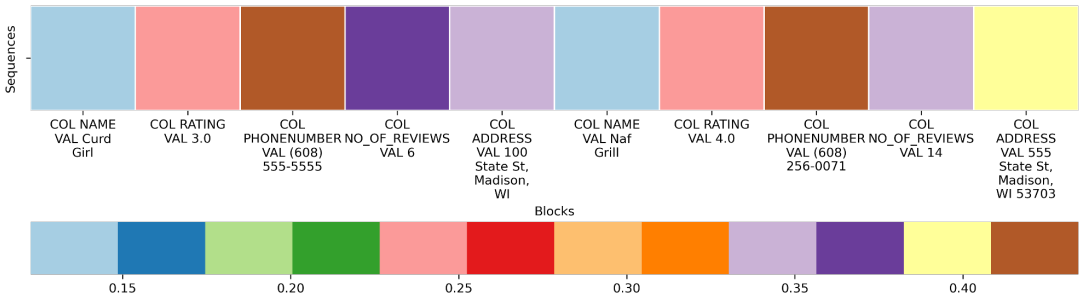


(a) Attention weights from CaSE

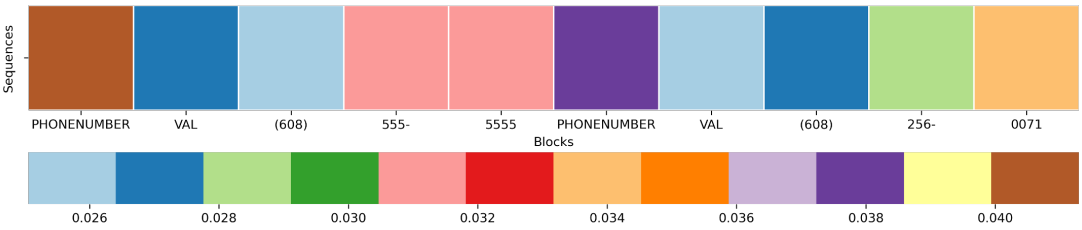


(b) Phone number attention weights from CaSE

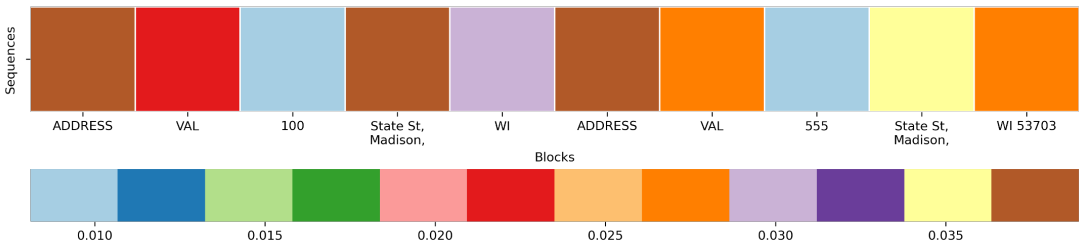
Figure 5.9: CaSE attention weights visualization on data V1 in scenario 3.



(a) Attention weights from DITTO



(b) Phone number attention weights from DITTO



(c) Address attention weights from DITTO

Figure 5.10: DITTO attention weights visualization on data V1 in scenario 3.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

In this thesis, we developed new DL models for schema matching and entity matching tasks by leveraging attention-based mechanisms.

We first demonstrate the effectiveness of AOA for the schema matching task by proposing `SMAT`. AOA can capture the relationships between the attribute name and columns without requiring prior domain knowledge. We also introduce a healthcare schema matching benchmark dataset `OMAP` as existing schema matching datasets only span purchase orders, web forms, and bibliographic references. The experimental results show that `SMAT` could improve the performance on `OMAP` and general schema matching benchmark datasets.

Next, we explore the AOA mechanism in combination with the multi-task learning paradigm on a BERT-based framework for the entity matching task. The proposed model, `EMBA`, extends JointBERT to utilize the individual BERT token representations with the AOA module to capture the relationships across the pair of entity token representations. The experimental results illustrate that `EMBA` achieves the best score on the larger datasets.

Finally, given the time-consuming and labor-intensive nature of building an entity matching benchmark dataset coupled with the key assumptions that do not reflect real-world applications, we curate a new benchmark dataset to contain different levels of entity matching complexity. We also propose to simultaneously perform the schema and entity matching task to overcome limitations related to the misalignment of attributes and their values. We introduce `CaSE`, a cross-attention framework with multi-task learning, to simultaneously conduct the schema and entity matching tasks. Our results on the new benchmark dataset demonstrate the potential of jointly matching both schema and entities to achieve comparable performance without requiring extensive data preprocessing.

6.2 Future Work

The proposed methods can be extended from the following aspects:

- Due to the limitation of existing datasets that support the entity ID prediction subtask in `EMBA`, we plan to annotate additional datasets to explore the generalizability of the model.

- When comparing the performance of `EMBA` with RoBERTa, we observe that RoBERTa provides better performance. We will try the RoBERTa as the backbone module for `EMBA` as well as other BERT distillation models to improve the performance for smaller datasets.

- Currently, `CaSE` considers the entity matching task as the main task, and does not obtain good performance on the schema matching task. We posit this is because the size (i.e., number of tables and attributes) of the schema matching dataset is small. Future work can explore the enrichment of other larger schema matching datasets to incorporate the entity matching datasets.

- Data privacy is one of the main concern in schema matching on sensitive datasets such as the healthcare datasets. We will introduce federated learning to `CaSE` through only sharing model updates e.g., the gradient information rather than the entity data.

- Since the real-world datasets involves data missingness and misalignment, we could extend our approaches to align the attribute values and impute the missingness based on the entity matching pairs. This could provide benefits for the recommender systems in our daily usage.

- It has been demonstrated that 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 arises numerous LLMs for question and answering, such as FLAN-T5 [12], LLaMA [76], and GPT-4 (backbone model of ChatGPT) [60]. We will implement these LLMs to fit our tasks, and facilitate the data integration domain.

Bibliography

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

[2] Bogdan Alexe, Mauricio Hernández, Lucian Popa, and Wang-Chiew Tan. Mapmerge: Correlating independent schema mappings. *Proceedings of the VLDB Endowment*, 3(1-2):81–92, 2010.

[3] Marcelo Arenas, Pablo Barceló, Leonid Libkin, and Filip Murlak. *Foundations of data exchange*. Cambridge University Press, 2014.

[4] Paolo Atzeni, Luigi Bellomarini, Paolo Papotti, and Riccardo Torlone. Meta-mappings for schema mapping reuse. *Proc. VLDB Endow.*, 12(5):557–569, January 2019. ISSN 2150-8097. doi: 10.14778/3303753.3303761. URL `https://doi-org.proxy.library.emory.edu/10.14778/3303753.3303761`.

[5] Zohra Bellahsene, Angela Bonifati, Fabien Duchateau, and Yannis Velegrakis. On evaluating schema matching and mapping. In *Schema matching and mapping*, pages 253–291. Springer, 2011.

[6] Philip A Bernstein, Jayant Madhavan, and Erhard Rahm. Generic schema matching, ten years later. *Proceedings of the VLDB Endowment*, 4(11):695–701, 2011.

[7] Ursin Brunner and Kurt Stockinger. 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, 2020.

[8] Sneha Chaudhari, Varun Mithal, Gungor Polatkan, and Rohan Ramanath. An attentive survey of attention models. *ACM Trans. Intell. Syst. Technol.*, 12(5), oct 2021. ISSN 2157-6904. doi: 10.1145/3465055. URL `https://doi.org/10.1145/3465055`.

[9] Chen Chen, Behzad Golshan, Alon Y Halevy, Wang-Chiew Tan, and AnHai Doan. Biggorilla: An open-source ecosystem for data preparation and integration. *IEEE Data Eng. Bull.*, 41(2):10–22, 2018.

[10] Chun-Fu Richard Chen, Quanfu Fan, and Rameswar Panda. Crossvit: Cross-attention multi-scale vision transformer for image classification. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 357–366, 2021.

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

[12] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*, 2022.

[13] CMS. Centers for medicare & medicaid services (cms). `https://www.cms.gov/OpenPayments/Explore-the-Data/Data-Overview.html`.

[14] Ronan Collobert and Jason Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167. ACM, 2008.

[15] Alexis Conneau, Douwe Kiela, Holger Schwenk, Loic Barrault, and Antoine Bordes. Supervised learning of universal sentence representations from natural language inference data. *arXiv preprint arXiv:1705.02364*, 2017.

[16] Yiming Cui, Zhipeng Chen, Si Wei, Shijin Wang, Ting Liu, and Guoping Hu. Attention-over-attention neural networks for reading comprehension. *arXiv preprint arXiv:1607.04423*, 2016.

[17] Dong Deng, Raul Castro Fernandez, Ziawasch Abedjan, Sibo Wang, Michael Stonebraker, Ahmed Elmagarmid, Ihab F Ilyas, Samuel Madden, Mourad Ouzzani, and Nan Tang. The data civilizer system. In *8th Biennial Conference on Innovative Data Systems Research, CIDR 2017*, 2017.

[18] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[19] Vincenzo Di Cicco, Donatella Firmani, Nick Koudas, Paolo Merialdo, and Divesh Srivastava. 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*, pages 1–4, 2019.

[20] Hong-Hai Do and Erhard Rahm. Coma—a system for flexible combination of schema matching approaches. In *VLDB'02: Proceedings of the 28th International Conference on Very Large Databases*, pages 610–621. Elsevier, 2002.

[21] Hong-Hai Do and Erhard Rahm. Matching large schemas: Approaches and evaluation. *Information Systems*, 32(6):857–885, 2007.

[22] AnHai Doan, Pedro Domingos, and Alon Y Halevy. Reconciling schemas of disparate data sources: A machine-learning approach. In *Proceedings of the 2001 ACM SIGMOD international conference on Management of data*, pages 509–520, 2001.

[23] Q. Dong, S. Gong, and X. Zhu. Imbalanced deep learning by minority class incremental rectification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(6):1367–1381, June 2019. doi: 10.1109/TPAMI.2018.2832629.

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

[25] Muhammad Ebraheem, Saravanan Thirumuruganathan, Shafiq Joty, Mourad Ouzzani, and Nan Tang. Distributed representations of tuples for entity resolution. *Proceedings of the VLDB Endowment*, 11(11):1454–1467, 2018.

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

[27] Entity Resolution. Data integration: Entity resolution. `https://thodrek.github.io/CS839_spring18/lectures/Lecture_9_ER1.pptx`.

[28] Ronald Fagin, Laura M Haas, Mauricio Hernández, Renée J Miller, Lucian Popa, and Yannis Velegrakis. Clio: Schema mapping creation and data exchange. In *Conceptual modeling: foundations and applications*, pages 198–236. Springer, 2009.

[29] Ronald Fagin, Phokion G Kolaitis, Lucian Popa, and Wang-Chiew Tan. Schema mapping evolution through composition and inversion. In *Schema matching and mapping*, pages 191–222. Springer, 2011.

[30] Raul Castro Fernandez, Essam Mansour, Abdulhakim A Qahtan, Ahmed Elmagarmid, Ihab Ilyas, Samuel Madden, Mourad Ouzzani, Michael Stonebraker, and Nan Tang. Seeping semantics: Linking datasets using word embeddings for data discovery. In *2018 IEEE 34th International Conference on Data Engineering (ICDE)*, pages 989–1000. IEEE, 2018.

[31] Cheng Fu, Xianpei Han, Jiaming He, and Le Sun. Hierarchical matching network for heterogeneous entity resolution. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, pages 3665–3671, 2021.

[32] Avigdor Gal. Uncertain schema matching. *Synthesis Lectures on Data Management*, 3(1):1–97, 2011.

[33] Avigdor Gal, Haggai Roitman, and Roee Shraga. Learning to rerank schema matches. *IEEE Transactions on Knowledge and Data Engineering*, 2019.

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

[35] Mozhdeh Gheini, Xiang Ren, and Jonathan May. Cross-attention is all you need: Adapting pretrained transformers for machine translation. *arXiv preprint arXiv:2104.08771*, 2021.

[36] Alon Halevy, Ema Nemes, Xin Dong, Jayant Madhavan, and Jun Zhang. Similarity search for web services. In *Proceedings of the 30th VLDB Conference*, pages 372–383, 2004.

[37] Lushan Han, Abhay L Kashyap, Tim Finin, James Mayfield, and Jonathan Weese. Umbc_ebiquity-core: Semantic textual similarity systems. In *Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity*, pages 44–52, 2013.

[38] Stephen Hayne and Sudha Ram. Multi-user view integration system (muvis): An expert system for view integration. In *[1990] Proceedings. Sixth International Conference on Data Engineering*, pages 402–409. IEEE, 1990.

[39] Bin He and Kevin Chen-Chuan Chang. Statistical schema matching across web query interfaces. In *Proceedings of the 2003 ACM SIGMOD international conference on Management of data*, pages 217–228, 2003.

[40] Mauricio Hernandez, Howard Ho, Felix Naumann, and Lucian Popa. Clio: A schema mapping tool for information integration. In *8th International Symposium on Parallel Architectures, Algorithms and Networks (ISPAN'05)*, pages 1–pp. IEEE, 2005.

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

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

[43] Junjie Huang, Duyu Tang, Wanjun Zhong, Shuai Lu, Linjun Shou, Ming Gong, Daxin Jiang, and Nan Duan. Whiteningbert: An easy unsupervised sentence embedding approach. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 238–244, 2021.

[44] Andrew Jaegle, Sebastian Borgeaud, Jean-Baptiste Alayrac, Carl Doersch, Catalin Ionescu, David Ding, Skanda Koppula, Daniel Zoran, Andrew Brock,

Evan Shelhamer, et al. Perceiver io: A general architecture for structured inputs & outputs. *arXiv preprint arXiv:2107.14795*, 2021.

[45] Alistair EW Johnson, Tom J Pollard, Lu Shen, H Lehman Li-wei, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. Mimic-iii, a freely accessible critical care database. *Scientific data*, 3:160035, 2016.

[46] Mohamed Salah Kettouch, Cristina Luca, Mike Hobbs, and Sergiu Dascalu. Using semantic similarity for schema matching of semi-structured and linked data. In *2017 Internet technologies and applications (ITA)*, pages 128–133. IEEE, 2017.

[47] Phokion G Kolaitis. Schema mappings, data exchange, and metadata management. In *Proceedings of the twenty-fourth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 61–75, 2005.

[48] Pradap Venkatramanan Konda. *Magellan: Toward building entity matching management systems*. The University of Wisconsin-Madison, 2018.

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

[50] Bing Li, Wei Wang, Yifang Sun, Linhan Zhang, Muhammad Asif Ali, and Yi Wang. Grapher: token-centric entity resolution with graph convolutional neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8172–8179, 2020.

[51] Peizhao Li, Jiuxiang Gu, Jason Kuen, Vlad I Morariu, Handong Zhao, Rajiv Jain, Varun Manjunatha, and Hongfu Liu. Selfdoc: Self-supervised document representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5652–5660, 2021.

[52] Yuliang Li, Jinfeng Li, Yoshihiko Suhara, AnHai Doan, and Wang-Chiew Tan. Deep entity matching with pre-trained language models. *arXiv preprint arXiv:2004.00584*, 2020.

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

[54] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[55] Giansalvatore Mecca, Paolo Papotti, and Donatello Santoro. Schema mappings: From data translation to data cleaning. In *A Comprehensive Guide Through the Italian Database Research Over the Last 25 Years*, pages 203–217. Springer, 2018.

[56] Sidharth Mudgal Sunil Kumar. Deep learning for entity matching: A design space exploration. Technical report, 2018.

[57] Quoc Viet Hung Nguyen, Matthias Weidlich, Thanh Tam Nguyen, Zoltán Miklós, Karl Aberer, and Avigdor Gal. Reconciling matching networks of conceptual models. Technical report, 2019.

[58] Zhaoyang Niu, Guoqiang Zhong, and Hui Yu. A review on the attention mechanism of deep learning. *Neurocomputing*, 452:48–62, 2021.

[59] Observational Health Data Sciences and Informatics. *The book of OHDSI*. Independently published, 2019.

[60] OpenAI. Gpt-4 technical report, 2023.

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

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

[63] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

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

[65] Erhard Rahm and Philip A Bernstein. A survey of approaches to automatic schema matching. *the VLDB Journal*, 10(4):334–350, 2001.

[66] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. " 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*, pages 1135–1144, 2016.

[67] Haggai Roitman and Avigdor Gal. Ontobuilder: Fully automatic extraction and consolidation of ontologies from web sources using sequence semantics. In *International Conference on Extending Database Technology*, pages 573–576. Springer, 2006.

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

[69] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*, 2015.

[70] Roee Shraga, Avigdor Gal, and Haggai Roitman. Adnev: cross-domain schema matching using deep similarity matrix adjustment and evaluation. *Proceedings of the VLDB Endowment*, 13(9):1401–1415, 2020.

[71] Duyu Tang, Bing Qin, and Ting Liu. Aspect level sentiment classification with deep memory network. *arXiv preprint arXiv:1605.08900*, 2016.

[72] Balder ten Cate, Phokion G Kolaitis, Kun Qian, and Wang-Chiew Tan. Active learning of gav schema mappings. In *Proceedings of the 37th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*, pages 355–368, 2018.

[73] Kai-Sheng Teong, Lay-Ki Soon, and Tin Tin Su. Schema-agnostic entity matching using pre-trained language models. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 2241–2244, 2020.

[74] Saravanan Thirumuruganathan, Nan Tang, Mourad Ouzzani, and AnHai Doan. Data curation with deep learning. In *EDBT*, pages 277–286, 2020.

[75] Nguyen Thanh Toan, Phan Thanh Cong, Duong Chi Thang, Nguyen Quoc Viet Hung, and Bela Stantic. Bootstrapping uncertainty in schema covering. In *Australasian Database Conference*, pages 336–342. Springer, 2018.

[76] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[77] Jianhong Tu, Xiaoyue Han, Ju Fan, Nan Tang, Chengliang Chai, Guoliang Li, and Xiaoyong Du. Dader: Hands-off entity resolution with domain adaptation. *Proc. VLDB Endow.*, 15(12):3666–3669, sep 2022. ISSN 2150-8097. doi: 10.14778/3554821.3554870. URL https://doi.org/10.14778/3554821.3554870.

[78] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[79] Jesse Vig. A multiscale visualization of attention in the transformer model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 37–42, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-3007. URL https://www.aclweb.org/anthology/P19-3007.

[80] Jason Walonoski, Mark Kramer, Joseph Nichols, Andre Quina, Chris Moesel, Dylan Hall, Carlton Duffett, Kudakwashe Dube, Thomas Gallagher, and Scott McLachlan. Synthea: An approach, method, and software mechanism for generating synthetic patients and the synthetic electronic health care record. *Journal of the American Medical Informatics Association*, 25(3):230–238, 08 2017. ISSN 1527-974X. doi: 10.1093/jamia/ocx079. URL https://doi.org/10.1093/jamia/ocx079.

[81] Zhen Wang, Xiang Yue, Soheil Moosavinasab, Yungui Huang, Simon Lin, and Huan Sun. Surfcon: Synonym discovery on privacy-aware clinical data. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 2019.

[82] 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. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, October 2020.

[83] Wensheng Wu, Clement Yu, AnHai Doan, and Weiyi Meng. An interactive clustering-based approach to integrating source query interfaces on the deep web. In *Proceedings of the 2004 ACM SIGMOD international conference on Management of data*, pages 95–106, 2004.

[84] Shen Yan, Xuehan Xiong, Anurag Arnab, Zhichao Lu, Mi Zhang, Chen Sun, and Cordelia Schmid. Multiview transformers for video recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3333–3343, 2022.

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

[86] Clement Yu, Wei Sun, Son Dao, and David Keirsey. Determining relationships among attributes for interoperability of multi-database systems. In *[1991] Proceedings. First International Workshop on Interoperability in Multidatabase Systems*, pages 251–257. IEEE, 1991.

[87] Jing Zhang, Bonggun Shin, Jinho D Choi, and Joyce C Ho. Smat: An attention-based deep learning solution to the automation of schema matching. In *European Conference on Advances in Databases and Information Systems*, pages 260–274. Springer, 2021.

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

[89] Ziqi Zhang, Christian Bizer, Ralph Peeters, and Anna Primpeli. Mwpd2020: Semantic web challenge on mining the web of html-embedded product data. In *MWPD@ ISWC*, 2020.

[90] Chen Zhao and Yeye He. Auto-em: End-to-end fuzzy entity-matching using pre-trained deep models and transfer learning. In *The World Wide Web Conference*, pages 2413–2424, 2019.

[91] Brady Zhou and Philipp Krähenbühl. Cross-view transformers for real-time map-view semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13760–13769, 2022.