Probing the Robustness of Pre-trained Language Models for Structured and Unstructured Entity Matching

by

Mehdi Akbarian Rastaghi

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science

Department of Computing Science

University of Alberta

Abstract

The paradigm of fine-tuning Pre-trained Language Models (PLMs) has been successful in Entity Matching (EM). Many contemporary works leverage PLM-based models to push the state of the results. However, using the power of transformer-based models has some downsides in this task. Despite their remarkable performance, PLMs exhibit a tendency to learn spurious correlations from training data. In this thesis, we aim to investigate whether PLM-based EM models can be trusted in real-world applications where data distribution is different from that of training. To this end, we design an evaluation benchmark to assess the robustness of structured EM models to facilitate their deployment in real-world settings. Then, we prescribe simple modifications that can improve the robustness of PLM-based EM models for structured and unstructured data. Also, to evaluate the model's performance on unstructured entity matching, we develop a new unstructured matching dataset. We extend our experiments further to study the effect of deep classifiers, data augmentation, and loss function on the model's robustness. Our experiments show that while yielding superior results for in-domain generalization, our proposed model significantly improves model robustness compared to state-of-the-art EM models.

Preface

The work on this thesis was a collaborative effort with Dr. Davood Rafiei as my graduate studies supervisor. Dr. Rafiei provided advice and feedback on improving the proposed approach, experimental design, presentation, and writing as I performed experiments, analyzed results, and derived conclusions. A part of this research is published at "The 31st ACM International Conference on Information and Knowledge Management" [3] in 2022.

To my family and friends

For their unconditional love and support

In practical life we are compelled to follow what is most probable ; in speculative thought we are compelled to follow truth.

– Baruch Spinoza, The Letters.

Acknowledgements

I would like to express my sincere appreciation for the assistance and guidance I have received from my supervisor, Dr. Davood Rafiei, as I carried out this research project. Clearly, this work would not have been possible without his continuous support and wisdom. I am pleased that I have had the chance to build and enhance my research skills with the generous help of his constructive feedback. I had the opportunity to think differently with the freedom he gave me under his supervision, and I was able to find novel solutions to problems.

By the same token, I would like to express my gratitude to the members of the Database Research Group, and in particular to my dear colleague, Ehsan Kamalloo, for his brilliant mind and ideas. I learned immensely from the knowledge-sharing sessions and their profound members with outstanding skills and senses. It helped me to stay on my path and focus on the research problem. As well as that, I would like to thank all my friends for being my ultimate source of moral and technical support, without which I would not have been able to enjoy my research to the fullest extent that I have.

My final gratitude goes out to both my parents and family, for they have been a source of inspiration and motivation for me all of my life and have motivated me to live to the fullest. Without their belief and encouragement, I would not have come so far.

Contents

List of Tables

List of Figures

List of Acronyms

DA	Data Augmentation
DK	Domain Knowledge Injection
DRP	Robustness to absence of non-key columns
EL	Entity Linking
EM	Entity Matching
EXT	Robustness to extraneous columns
LSTM	Long-Short Term Memory
MIS	Robustness to missing values
MLM	Masked-Language Modeling
NED	Named Entity Disambiguation
NER	Named Entity Recognition
NLP	Natural Language Processing
NSP	Next Sentence Prediction
PLM	Pre-trained Language Model
RNN	Recurrent Neural Networks
SFF	Robustness to column order
TYP	Robustness to different data types

Chapter 1

Introduction

1.1 Problem & Motivation

In most natural language tasks, named entities play an important role, but unfortunately, they can be traceless. In general, documents, records, and data sources contain mentions of entities that are ambiguous in most cases. The ambiguity sometimes can happen because of partial data representation. To shed more light, consider these two examples:

Example 1.1.1 Both sentences contains 'Bush' but the first one refers to George W. Bush, while the second one is about George H. W. Bush , the father.

1. Mr. **Bush** got a crucial helping hand in life because of his name and family connections. Otherwise, he would probably not have been admitted to Andover and then Yale. [44]

2. At Andover, **Bush** was captain of the baseball and soccer teams, and the senior class president. He graduated on his eighteenth birthday in 1942. That same day, he enlisted in the United States Navy. [39]

Although both sentences in this example contain the word '**Bush**', what is not explicitly clear is whether they reference the same real-world entity or, to be more accurate, the person who is being referenced in these sentences is the same. This ambiguity is primarily due to the fact that we have multiple people

with their names starting with **Bush** and these sentences do not contain the whole name, which is a typical pattern.

This problem can get even more complex in cases where the mention contains the full name, e.g. a product title or a person's full name, but still not linkable to a single real-world entity as there are multiple entities with the same identifier. For example, the name '**James Smith**' can refer to multiple persons, some of which are shown below, based on Wikipedia:

Example 1.1.2 4 examples out of more than 300 returned result for the name 'James Smith' from Wikipedia.

- **James Smith**, Australian rules footballer for Richmond Football Club.
- **James Smith**, American boxer and host of In This Corner
- **James Smith**, Medal of Honor recipient in the American Civil War
- **James Smith**, Canadian-born philosopher

In both examples, the surface text of the mention alone is not enough to identify the reference. However, by incorporating both the mention and its surrounding text into the analysis, we may be able to gain a more robust understanding of the context and, in most cases, be able to disambiguate the mentions. Considering the second sample in **Example 1.1.1**, the combination of other mentions in the surrounding text can lead us to link the mention to George H. W. Bush:

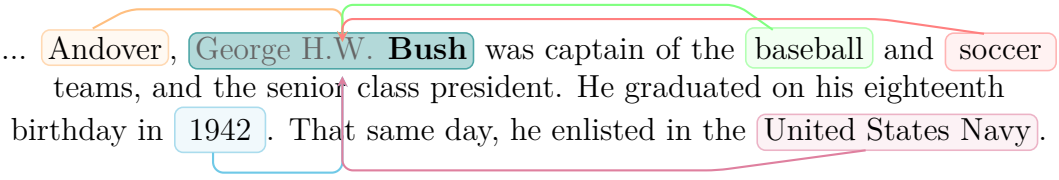


Figure 1.1: Cluster of mentions in the surrounding text can help with the mention disambiguation.

Linked mentions in a document can play a crucial role in Natural Language Processing (NLP) tasks. They may assist in extracting more information about named entities, discovering new entities, obtaining abstract representations of texts, and carrying out semantic searches. As a result, a few highly important subtasks in natural language understanding have been introduced for identifying these names:

- **Named Entity Recognition (NER)**, which recognizes and categorizes mentions based on predefined categories such as Person, Organization, Location, Time Expression, Quality, and other prevalent categories.
- **Named Entity Disambiguation (NED)**, which links the ambiguous mentions to a unique entity in a Knowledge Base (KB). These methods usually rely on a NER tool and a KB to extract the names from text and then match them to the candidate entities inside the KB.
- **Named Entity Matching (EM)**, Also known as Record Linkage, is a task that tends to match two different names in the same or across multiple data sources to determine whether these two mentions refer to the same real-world entity. Figure 1.2 shows different types of EM.

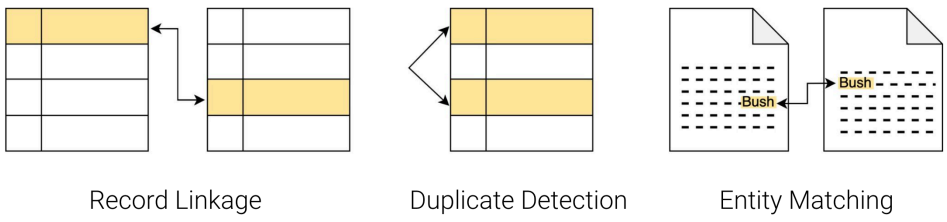


Figure 1.2: A few EM types. Duplicate Detection is a type of Record Linkage where the source and destination datasets are the same.

1.2 Entity matching

A **Named Entity** is a real-world object, such as a *Person*, a *Place*, an *Organization*, or a *Product*, which is often denoted by a proper noun in the context of the information extraction process. An entity mention can exist as a text span or a table record. An EM task can be defined as linking a proper name to a distinct real-world entity.

Problem Formulation Let D_1 and D_2 be two data sources containing entity mentions. The goal of EM is to find all pairs of entity mentions between D_1 and D_2 that refer to the same real-world entity based on the similarity of their attributes. These pairs are called matches. Commonly, the EM task has two stages: blocking phase and matching.

Blocking The goal of blocking is to reduce the search space for EM from cross product $D_1 \times D_2$ to a candidate set $C \subseteq D_1 \times D_2$ where $|C| << |D_1 \times D_2|$; C includes only pairs of entity mentions that are considered as potential matches. Without this step, EM has a time complexity of $O(n^2)$, as every mention must be compared and checked with all others [65]. Many researchers study blocking and propose methods that can be categorized into four classes: attribute equivalence, sorted neighbourhood, hash-based, and deep learning [65].

Attribute equivalence aims at filtering mention pairs based on their attribute values, keeping the pairs that share similar attributes. The simplest form of equivalence or similarity is substring matching, where a pair remains a candidate if their textual similarity score exceeds a threshold. Hash-based blocking is a generalization of attribute equivalence, which keeps a pair if they share the same hash value, using a pre-specified hash function.

In the sorted neighbourhood approach, records are sorted according to some

attribute(s). Two records are considered candidates if they are near each other in the sorted order. This may be implemented using a fixed window that is moved sequentially over the sorted list. For each window, a candidate set is created by pairing each record within the window with other records in the window. The set of all those candidates forms C. On the other hand, deep learning-based methods find candidate matches by capturing and considering the semantic properties of data [5].

In this work, we use datasets introduced in DeepMatcher [60], which has the candidate set C compiled and ready to use. These datasets are developed using the Magellan [40] framework by combining different blocking systems. As a result, our study does not focus on blocking.

As mentioned earlier, entity matching has to decide which two records refer to the same real-world entity. This is essential in data integration, which often arises when two data records describe to the same entity but are obtained from different sources. Although EM has a simple definition, it is generally difficult to suggest a sound solution. There are several factors that make it challenging.

Multiple Sources The main challenge is that EM models are required to match entities usually from different sources with different schema and poor data quality. Data pre-processing is often essential for a sound EM solution since stored data are hardly clean, well-structured, or homogeneous [6]. It is important to bring both data sources into a uniform or a similar format, by transforming the schemes as close as possible yet maintaining the semantics.

Voluminous Sources The number of potential or candidate matches can be huge even though the number of correct matches can be very small. Such characteristics can cause an extreme imbalance between positive and negative matches [6]. In order to avoid checking all possible combinations, the Block-

ing step is required to ignore pairs of entries that are far-fetched to refer to the same real-world entity. This process can help to reduce the number of candidate pairs and improve performance, particularly in massive datasets.

Deep Contextual Knowledge Finally, with mentions even structured ideally, deep semantic information and external sources may be required to predict a correct match.

All the aforementioned factors make EM complex. To match two candidate mentions correctly, it is necessary to have a substantial understanding of the language and sometimes domain-specific knowledge. Therefore, EM remains a challenge, despite all the advanced approaches that have been developed to date.

1.3 Characteristics of Data and EM Models

As illustrated, EM is neither an easy nor a straightforward task. Recent EM models utilize Deep Learning based techniques and significantly improve the state-of-the-art results. One of these techniques is the paradigm of fine-tuning Pre-trained Language Models (PLMs), which is shown to have a remarkable performance.

Despite their superior performance, in some cases, deep learning models may be susceptible to imbalanced data or to extracting wrong patterns by emphasizing learning the implicit data pattern rather than the task. In this case, the trained model is highly adapted to the distribution and pattern in the data, which can affect the validity of results and influence the robustness of the model for EM tasks.

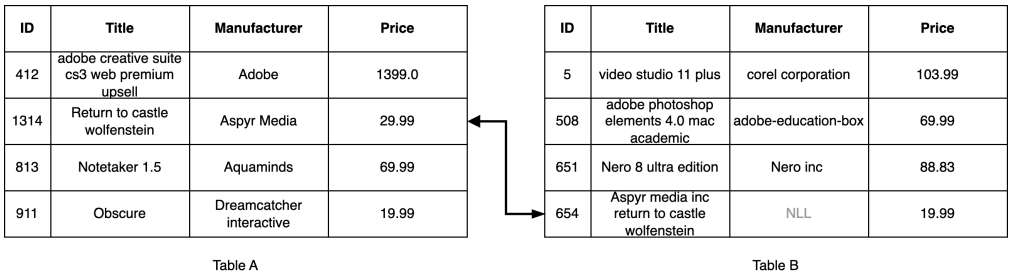
In this study, our main focus is on the PLM-based EM tasks. In order to provide a more detailed explanation of the task, the next section reviews different

data types on which EM may be performed.

1.3.1 Data types

Although EM arises in many domains and applications with different data formattings, these data types are usually categorized into three main classes: structured, unstructured, and semi-structured. Based on the type of data source that an EM model can work with, a model may also be categorized into structured EM for structured data, and unstructured EM for unstructured and semi-structured [60].

Structured Data Structured Data often refers to data in a tabular format where each row describes an entity using a set of columns and rows can have relations with other rows. This is also the standard format for many EM models. The main advantage of structured data type is that it makes the data serialization part much faster and easier as we have different attributes that define the entity as columns. Hence, a structured EM model can serialize these attributes by simply joining them together without much data pre-processing.



ID	Title	Manufacturer	Price
412	adobe creative suite cs3 web premium upsell	Adobe	1399.0
1314	Return to castle wolfenstein	Aspyr Media	29.99
813	Notetaker 1.5	Aquaminds	69.99
911	Obscure	Dreamcatcher interactive	19.99

Table A

ID	Title	Manufacturer	Price
5	video studio 11 plus	corel corporation	103.99
508	adobe photoshop elements 4.0 mac academic	adobe-education-box	69.99
651	Nero 8 ultra edition	Nero inc	88.83
654	Aspyr media inc return to castle wolfenstein	NLL	19.99

Table B

Figure 1.3: Structured EM models usually match records between two tables.

Unstructured Data For unstructured data, entity descriptions lack a pre-defined data model or a predefined structure. This type of data is usually heavy in text, and may contain dates, facts, and numbers in documents, which cannot be easily queried using SQL databases. Analyzing this type of data

is generally more complex than that of structured data as it may need more data preparation.

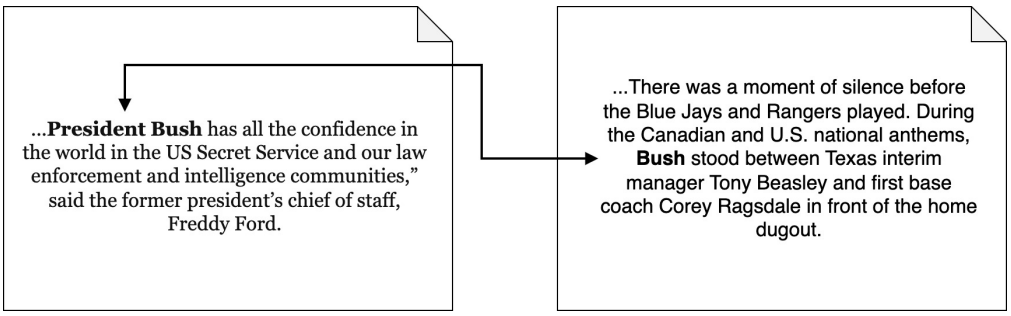


Figure 1.4: Unstructured EM models match text spans between two documents.

Semi-Structured Data Data in semi-structured format may not conform to a fixed structure such as a relational table; nevertheless, data may contain tags and markers to separate components or to enforce a hierarchy. Usually, semi-structured data contains a long body of unstructured data as a column or attribute.

1.4 Pre-Trained Language Models (PLMs)

Deep learning is a computational model, composed of multiple processing layers connected to learn a data representation with numerous levels of abstraction. Deep learning methods have improved the state-of-the-art in various tasks in NLP and other fields of study, including speech recognition, visual object recognition, and object detection. Deep learning uncovers complex features in large datasets by using the backpropagation algorithm to indicate how a machine should change its internal states that are used to compute the representation in each layer from the representation in the previous layer [73]. In other words, deep learning models consist of a large number of nodes distributed in different layers that captures features from the input of each layer as an abstraction to the next layer. Finally, the last layer compiles the internal

state of the previous layers, which are sometimes called hidden layers, as the output of the model. In order to capture the features from the input, we need to adjust the weight of each node in the network. A large amount of data may be needed for a model to master a complex task like Object Detection. This process is called the training step. After training a model, traditional deep learning models maintain their internal state to perform the task they are trained for, but they have a major problem with a sequence of inputs that are not independent.

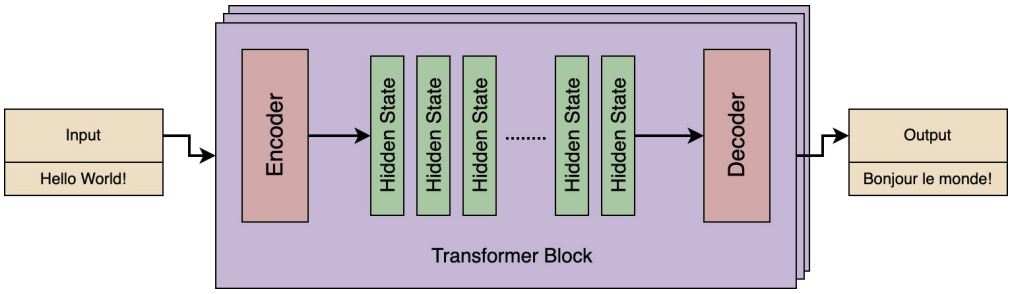


Figure 1.5: Unstructured EM models match text spans between two documents.

Tasks such as speech recognition [74], Machine Translation [79], and unsegmented connected handwriting recognition [29] are complex machine-learning problems for traditional feed-forward neural networks [83]. A deep learning model needs to be able to work on arbitrary sequences of inputs to solve these problems. Traditional models do not have any memory unit to maintain the association between the current input and previous ones. Recurrent Neural Networks (RNN) [35] and Long-Short Term Memory (LSTM) [33] networks are proposed to solve this problem. They have an internal memory unit that can maintain information from the previous iterations alongside the internal state of the model. Combining these two parameters helps them to understand the meaningful connection between inputs and solve more complex tasks, but even these models are not efficient enough for complex NLP tasks. The main issue with LSTM and RNN models is called Gradient vanishing. It simply means they cannot hold the history of older iterations and are not able to pro-

cess long sequences of data, including lengthy texts. The transformer model [85] is introduced to solve this problem. Transformer models use an Encoder-Decoder based architecture alongside an attention mechanism that helps them to extract context from a large sequence of data and keep their focus on the important part. Figure 1.5 shows the internal structure of a transformer block.

Pre-trained Language Models (PLMs) are general-purpose transformer-based models that are applicable to different Natural Language Processing (NLP) tasks. These models are trained on an extensive corpus such as Wikipedia. Due to the fact that transformers can process long input sequences, these models can understand the language context adequately. They can then get adapted for a specific NLP task [87] using the concept of fine-tuning a model, which is a transfer learning technique [2]. Document classification, text summarization, content generation, and sentence paraphrasing are some of the real-world applications of the PLMs [25], [80], [87], [90], [92], [93].

One of the most popular pre-trained language models is the Bidirectional Encoder Representations from transformers (BERT) [18] model, which to this day is still one of the state-of-the-art models. One of the key innovations of BERT is the application of bidirectional training to language modeling, contrary to previous efforts, which either analyzed text sequences left-to-right or combined left-to-right and right-to-left analysis. Here, the encoder part reads the entire sequence of input words at once, which helps the model to learn the context of a word based on all of its surroundings. Like other transformers, BERT is trained on a large dataset. The training step, as shown in Figure 1.6, consists of two objectives, Masked-Language Modeling (MLM) and Next Sentence Prediction (NSP):

- **MLM**: MLM stands for the training objective that 15% of the input tokens from the training set are masked randomly, and the model attempts to learn and predict the masked token based on the surrounding

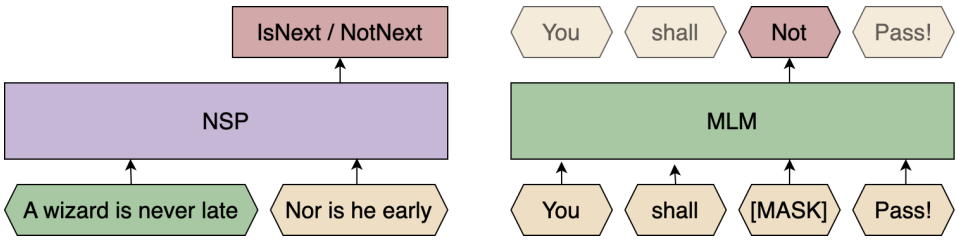


Figure 1.6: Masked-Language Modeling and Next Sentence Prediction objectives for training BERT

tokens.

- **NSP**: Unlike the MLM task, which focuses on a smaller context to predict the masked token, the NSP objective is to understand longer-term dependencies across sentences. In this task, the model receives two sentences and is expected to answer if the second sentence comes after the first sentence. Of the sentence pairs used for training, some are closely related sentences (with one sentence following the other in some text), but most of them are two random unrelated sentences. Learning to predict these connections can help the model to understand longer texts.

The performance of the models on different tasks shows that a bidirectional trained model would have a deeper understanding of the context compared to a single-directional model [18].

1.5 Contribution

Thesis Statement In this thesis, we aim to bring to light the unreliability problems of PLM-based EM models by studying their robustness capabilities under domain shift. We argue that under various out-of-domain scenarios that can arise in real-world applications, PLM-based models would fail as they are known to rely on spurious patterns in the data. To the best of our knowledge, no previous work has studied the robustness of PLM-based models for the EM task.

This study focuses on EM models using PLMs and aims at improving their robustness by applying simple modifications that are discovered based on intensive experiments. We introduce RobEM, an improved PLM-based EM model that focuses on robustness with the proposed modifications. PLMs, due to their exceptional language understanding performance, have been widely used in different tasks. While this characteristic makes them an excellent choice for handling the EM task, the same ability to discover and understand the implicit sense in data raises the question of whether the superior performance is related to the patterns in data or whether the model mastered the EM task.

The first step to understanding the question and its answer is to investigate the common strategies in EM models from a robustness perspective. In this step, we consider different structured EM task characteristics and how they affect a PLM-based EM model. To assess the robustness in a real-world setup, we evaluate the model with different distribution shifts on a few benchmark datasets. We also propose a new benchmarking schema for data perturbation and various distribution shifts.

Following an extensive evaluation of the model, we propose a few modifications to the current state-of-the-art model, Ditto, in order to improve the robustness of the model both within and outside the domain. Furthermore, we extend our proposed model, RobEM, to be applicable to unstructured data sources. As a final step, we provide a new distribution-shifted dataset, adapted from CoNLL En 2003, that can be used for the evaluation of RobEM in a real-world environment.

1.6 Outline

The rest of this study is organized as follows:

Chapter 2 discusses the background and the related work. In this chapter, we also review a few related tasks to entity matching.

Chapter 3 discusses our proposed model, RobEM, for structured EM. This chapter also examines the common strategies in EM models based on our proposed robustness approach. In the end, we compare the performance of RobEM to that of the current state-of-the-art models, to demonstrate the effectiveness of the proposed model.

Chapter 4 extends the RobEM model to unstructured data sources. Furthermore, we generate a dataset based on CoNLL En 2003 to evaluate our model on documents and compare its results with a state-of-the-art model. We also test the model on a new balanced dataset to assess its performance in out-of-distribution environment. The new dataset is generated by combining documents from CoNLL and scraping additional documents from the Internet using Google news and Wikidata.

Chapter 5 Finally, Chapter 5 recaps our findings, concludes the thesis, and provides some possible directions for future work.

Chapter 2

Related Works & Backgrounds

Information Extraction can be categorized into several sub-tasks, and one of these sub-tasks is Entity matching (EM) which attempts to determine whether two instance representations point to an identical real-world entity. The problem of entity matching has been considered under various names from different communities since its definition in 1959 by Newcombe et al. [61]. While numerous approaches have been suggested, most traditional EM approaches are based on features and attributes similarity, rule extraction, statistical methods, and learning techniques.

Despite the remarkable progress made using traditional approaches, most recent solutions use deep learning models such as recurrent neural networks and long short-term memory due to their superior representation of semantic relations and contextual information. However, some of these models are limited in their contextual representation, especially when they are applied to long sequences of text. PLMs are better positioned to address this problem, and many recent EM works employ PLMs.

In this chapter, we briefly review different classes of work on EM. We start with a review the most prominent and cited traditional EM methods to have a better overview of the task. Then we study the deep learning based approaches, their advantages, and limitations over traditional models. Furthermore, we explore

the PLM-based solutions to EM, their strengths, and some of the limitations this thesis aims to address.

Finally, we review a few closely related tasks to EM. While these related tasks aim for a different goal, some of the challenges are the same or similar. For example, entity alignment, as one of these tasks, tries to match entities from two knowledge graphs.

2.1 Structured Entity Matching

We start this section with a review of traditional methods before moving into discussing more contemporary deep learning approaches inclduing non-PLM and PLM-based models.

2.1.1 Traditional Approaches

EM has long been studied in the database community under different names such as data duplication, record linkage, etc. [21]. Jin et al. [37] use Levenshtein distance [48] to compare the textual descriptions of two enties. Levenshtein distance, commonly used for measuring textural similarity, is the minimum number of perturbations, in the character level, required to transform from string to another string. An optimal Levenshtein distance is determined using a dynamic programming algorithm. Using this method, the authors achieve an improvement in accuracy. However, the time complexity of this algorithm is an issue on large databases. Also feature-based methods are generally dependent on the choice of a similarity function and a similarity threshold. Finding a good similarity function and an optimal threshold is always challenging, making the approach not flexible for out-of-domain environments.

Rule-based EM models use a set of matching rules to measure the similarity

between a pair of records. As an example, $Jacard(title_a, title_b) \geq 0.9$ is a simple rule for matching the titles. In [22], the authors aim to find dependencies between attributes to optimize the rule set and match tuples from unreliable data sources by assessing a subset of key attributes. By identifying the dependencies between attributes, the authors optimize the rule set to determine what attributes to compare and how to compare them. For example, two tables containing credit card holder information may have two rows referencing the same person with the same value for the postal code but different values in the address field. Since there is a dependency between the address and the postal code, the similarity of the two addresses of the client may be inferred based on the postal code alone. They also propose a general algorithm to identify a subset of attributes as the key for EM. Their proposed method outperforms the baseline result by 20% in recall and precision and improves the performance up to 30% on a credit card dataset. Similarly, Lim et al. [52] use a rule-based method in their study, but they need to ensure that the rules yield the right results every time. It is, therefore, critical that the rules do not reflect heuristics but rather reflect absolute truths and serve as functional dependencies.

The authors of [15] suggest a learning-based solution to estimate the likelihood ratio and the correlation between entity pairs. First, they train their model on a labeled pair of entities. Afterward, the authors use the learned values and the trained weights as the statistical parameters of the model to present all possible matching pairs in the test dataset. Verykios et al. [86] suggest a method for matching records where the possibility of missing values and inconsistency exists. Their decision tree model is cost-optimal, where the problem of matching is considered as a classification task and a bayesian decision theory that is utilized with constant error costs for certain types of error. They use a probability distribution as the probable cost values for the errors when their model is uncertain.

Singh et al. [77] attempt to formalize matching rules as General Boolean Formulas and cast the problem of matching as Logic programming. Having all rules in General Boolean Formulas format allows them to generate new rules by a logical combination of basic rules. As a result, their method can generate a concise and interpretable rule set automatically, given only positive and negative matching samples.

Rule-based approaches formalize the selection of similarity metrics for different attributes; however, it cannot help with the fact that these types still need to be tuned by an expert to adapt to a new setup. Learning-based methods can minimize this effort. Tejada et al. [84] develop an EM framework, called Active Atlas, which compares the shared attributes of the instances being matched. In this system, specific attributes are essential for deciding if two entities match. Unlike previous methods that require manual construction of rules, Active Atlas learns to tailor mapping rules based on these essential attributes, through limited user input, to a specific application domain. The experimental results demonstrate that Active Atlas achieves higher accuracy across various application domains while requiring less expert involvement than previous methods.

Using trainable measures of textual similarity, Bilenko and Mooney [9] present MARLIN, a framework for improving duplicate detection. They propose learnable text distance functions for each database field. Additionally, the study shows that such measures can adapt to a notion of similarity that is appropriate to the domain values of a field. The authors present two learnable text similarity measures: an extended variant of learnable string Levenshtein distance and a new vector-space-based measure trained using Support Vector Machines. It has been shown that the proposed framework is more accurate than rule-based and attribute-based techniques at detecting duplicate data across six datasets, including Fodor and Zagat and Cora[1].

[1]Duplicate Detection, Record Linkage, and Identity Uncertainty: Datasets provided on: https://www.cs.utexas.edu/users/ml/riddle/data.html

2.1.2 Deep Learning Approaches

Prior to PLMs DeepMatcher [60], a prominent EM model, uses RNNs and the attention mechanism. In this paper, Mudgal et al. propose a two-step approach, where in the first step, an attribute embedding component takes the value of each column and generates an attribute embedding. Then in the second step, an attribute similarity representation summarizes and compares attributes to generate a final similarity score.

Auto-EM [98] applies transfer learning, similar to the popular trend in PLMs, to learn from a general-purpose model trained on massive knowledge bases. The authors show that a hierarchical neural network architecture can be used for training over a KB with rich synonyms of the same entity types in order to improve the ability of the model to cope with insufficient training data in an unexplored environment.

In another work [19], the authors present DeepER, an EM approach based on RNNs and LSTMs. Their fundamental contribution is specifying proper word embedding as their serialization layer for EM. This serialization layer is the primary part of an effective EM solution which serializes a tuple into a proper word embedding. In addition, authors develop an EM classifier and an efficient blocking strategy based on Locality Sensitive Hashing [27]. To this end, several word embeddings, including word2vec [57], GloVe [68], and fastText [10] are tested on structured data to investigate how the results are affected. DeepER achieves a higher F1-score and maintains its performance on multiple datasets introduced in Magellan [40] compared to the rule-based and ML-based state-of-the-art methods [17], [28], [40], [41].

PLM-based Approaches Recent EM models have shifted to the paradigm of fine-tuning PLMs to tackle the problem and achieve the highest performance using their excellent understanding and learning capability of languages. Ditto

[51], the state-of-the-art EM model, concatenates the attributes in a pair of records to form a sequence and fine-tunes a PLM using a sequence classification objective. As part of their work, the authors also study standard optimizations in EM, including summarization for long textual columns, enriching attributes with domain knowledge using Entity Resolution tools, and a wide range of Data Augmentation techniques. Ditto uses attribute headings or column names in its serialization step, which can be a challenge because the header is not always guaranteed to be available.

JointBERT [66] uses a dual-objective training method that combines binary matching and multi-class classification, forcing the model to predict entity identifiers and matching decisions. The multi-class classifier is responsible for learning and identifying descriptions of individual entities. Then the binary matching decides the matching of the pairs of identifying descriptions. For each pair of labeled entities, they train their model with two objectives, multi-class classification and binary matching, simultaneously. Through this dual-objective training, the model is forced to view the matching task not only as a comparison of two isolated entity descriptions but as an opportunity to incorporate into its decision-making information from other descriptions of the respective entities seen during training. This method improves the model performance for seen entities by 1% to 5%, in terms of the F1-score, compared to single-objective transformer-based solutions. However, for the same reason, the model underperforms on unseen entities.

A recent study [99] proposed REMS, a Relation-aware Entity Matching method, to investigate and employ the potential relationship between different columns to improve the task. In order to add these relationships, they first perform a preprocessing step to extract any possible relation. For example, for a record containing name, manufacturer, and release data, there is a *made by* relationship between the name and the manufacturer, which can be injected into the serialization. As another example, having a table for books with their title

and authors, we can join the title and author using a *written by* relationship. Using Sentence-BERT, they embed attribute and their relations into an entity description and cast the problem of EM as sentence similarity.

2.2 Unstructured Entity Matching

During the past few years, extensive efforts have been made to develop advanced matching techniques and benchmarks for entity matching (EM); however, most of these resources are developed under the assumption that both entities exist in a structured data format or table records. Nevertheless, the types of data collection in the real-world for EM could be unstructured, semi-structured, and structured, for which these approaches are not applicable. Unstructured EM is a task where the mentions are mostly text spans inside large documents, and representing these mentions is essential. Unlike the entity linking and entity disambiguation tasks that attempt to match mentions to a knowledge graph, unstructured EM aims to match different mentions in documents without any reference to a knowledge graph. Unstructured EM is crucial when documents contain mentions of emerging entities, where a reference to a knowledge graph is not available. This section reviews the literature on unstructured EM in addition to recent works introducing General Entity Matching as a new approach that applies to both structured and unstructured data types [88]. Similat to the structured EM, our review is not exhaustive and lists some of the most renowned works referenced in the literature.

2.2.1 Traditional Approaches

Ali and Cristianini [4] provide an end-to-end system for extracting and matching named entities from unstructured news. This work aims to match named entities from more than 9 million scraped news articles from the internet using Information fusion techniques. As the name suggests, information fusion is the

process of combining metrics and attributes to improve accuracy. They compute and aggregate the string similarity scores using Levenshtein and q-Gram [82], a graph neighborhood similarity [8], and a binary co-reference similarity of mentions to achieve a linear combined similarity metric which helps to improve the precision and recall. To compute the co-reference similarity, they generate a co-reference set for each mention and check for the intersection of the sets. The reported result of combined metrics outperforms the baseline results achieved by individual metrics.

Authors in [12] propose an end-to-end EM pipeline to minimize the human labeling effort for entity matching on unstructured data sets. Their model uses an Active Learning approach with several NLP techniques to preprocess the record pairs on unstructured data before a classification phase. Their model can be used as a plugin for other methods, such as support vector machines (SVM), random forests, and deep neural networks.

2.2.2 Deep Learning Approaches

An EM model is introduced by Brunner and Stockinger [13], based on four common PLMs, BERT [18], XLNet [95], RoBERTa [54] and DistilBERT [76]. The authors conduct comprehensive experiments to compare these PLMs. The authors propose a General Entity Matching model as it can work both with structured and unstructured data. They report that a PLM-based model outperforms traditional methodologies with a 27.5% margin on average. Paganelli et al. [64] conduct a deep study of a PLM-based model for EM on both structured and unstructured data. Their model is implemented using BERT and fine-tuned for the EM task. Their study aims to perform a multi-facet investigation of the elements to determine how fine-tuning the model on the EM task affects PLMs. According to their findings, fine-tuning mainly affects the last layers of the BERT architecture, and this layer primarily contributes to classifying matching / non-matching entities. Also, their study shows PLM-based

models can learn the structure of input data, demonstrating the susceptibility of the model to distribution shift and domain changes.

2.3 EM relation to other IE Tasks

EM is usually an important component in automated retrieval of information about a selected subject from textual data sources. There are multiple other components that are either related or address part of the same problem. Entity Alignment is another task in this category. While structured and unstructured EM concentrate on matching entities between two tables or documents, Entity Alignment seeks to find and match equivalent relations between entities in different knowledge graphs. The primary Entity Alignment task is detecting dangling entities where an entity cannot be matched or aligned with other entities across knowledge graphs. Sun et al. [81] design a multi-task learning framework based on the distribution of nearest-neighbor distances, nearest-neighbor classification, marginal ranking, and background ranking to solve the problem of dangling entities. Initially, they find the matching entities between two knowledge graphs in an Entity Alignment step. Then they create a candidate list using the list of remaining dangling entities in each knowledge graph. Finally, they use their detection model to find and match entities. Likewise, Wu et al. [91] estimate the similarities between entities with the help of a neighborhood matching network, which captures neighborhood differences and topological structures to handle the matching process. Jain et al. [36] develop a solution to speed up matching heterogeneous entity representations in a low resource setting based on PLMs.

Named Entity Recognition (NER), as another task in this category, identifies text spans in a textual document that mentions a real-world entity. Many NER tools have been proposed using rule-based approachs [1], [20], [23], [72] and Deep Learning [2], [26], [62], [96], [97].

Lastly, Named Entity Disambiguation or Entity Linking (EL) is another important task in IE. As mentioned in Chapter 1, the goal of EL is to link the mentions to their referent entities in a knowledge graph. Despite the large body of past works and contributions in this area, the large number of ongoing research shows that the problem is not solved and that entity linking is an important task [14]. Similar to the EM, most recent studies in EL develop their model using PLMs [49], [55], [71], [78], [94].

Chapter 3

Structured Entity Matching

The paradigm of fine-tuning Pre-trained Language Models (PLMs) has been successful in entity matching (EM). Despite their remarkable performance, PLMs exhibit a tendency to learn spurious correlations from training data. In this chapter, we aim at investigating whether PLM-based structured EM models can be trusted in real-world applications where data distribution is different from that of training. To this end, we design an evaluation benchmark to assess the robustness of EM models to facilitate their deployment in real-world settings. We also introduce an unsupervised baseline, based on an off-the-shelf PLM, to serve as a strong baseline for distribution shift evaluations.

Our assessments reveal that data imbalance in the training data is a key problem for robustness. We also find that data augmentation alone is not sufficient to make a model robust. As a remedy, we prescribe simple modifications that can improve the robustness of PLM-based EM models. Our experiments show that while yielding superior results for in-domain generalization, our proposed model significantly improves the model robustness compared to state-of-the-art EM models.

3.1 Introduction

In many real-world applications where data is integrated from multiple sources, matching mentions that refer to the same real-world entities is crucial. Entity matching (EM) aims at automatically detecting such mentions or records that are likely derived from different schemas. With the recent success of transfer-learning from large pre-trained language models (PLMs) [11], [18], [69], [70] in many NLP tasks, EM models such as Ditto [51] have followed suit to leverage PLMs for EM. The paradigm of fine-tuning PLMs has achieved remarkable performance on several well-known EM benchmarks.

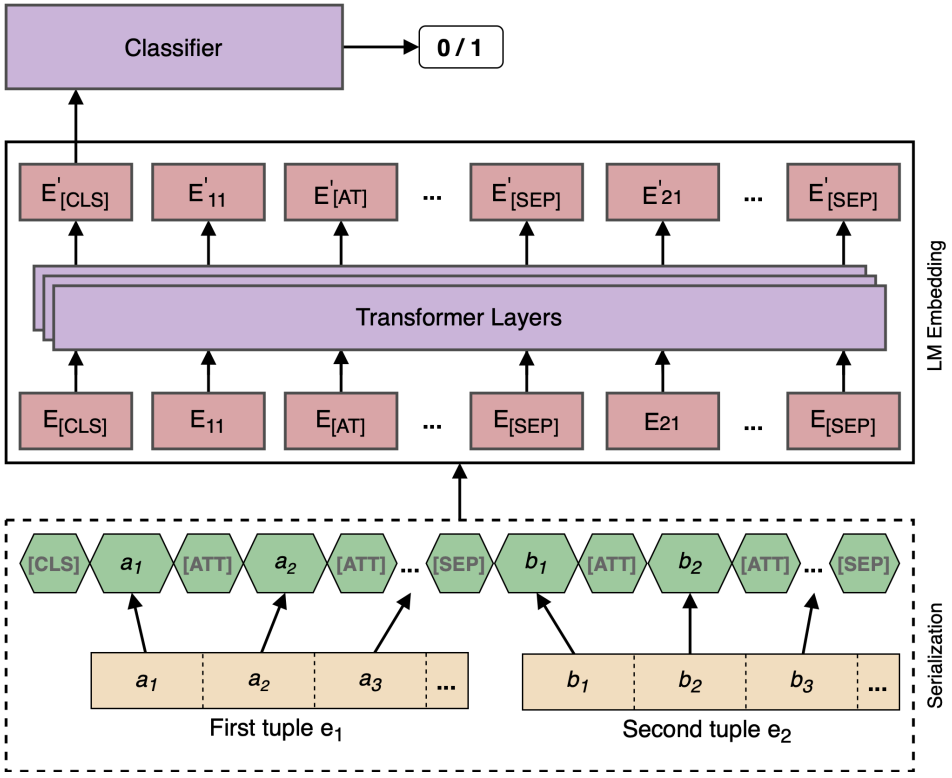


Figure 3.1: A schematic overview of our proposed model

Despite their success and popularity, PLMs are no panacea [7]. Numerous studies [56], [63] have found their tendency to learn the underlying spurious patterns in data. This essentially means that PLMs tend to acquire a superficial understanding of the task at hand and are more likely to fail under

different circumstances, such as distribution shift. Such shortcomings lead to inexplicable errors that inhibit their deployment in real-world applications.

EM can also be vulnerable to these problems. In this work, we investigate whether PLM-based models can be deployed for entity matching "in the wild," where the distribution of the test data often differs from that of the training data in a real-world setting. This is especially pivotal in EM as stored data from different sources are hardly homogeneous [6]. To this end, we study the robustness of these models to shed light on their pitfalls under various distribution shifts. Our focus in this paper is on "structured" data where the content of the records and the ordering of the fields vary from one domain to the next.

For this purpose, we first fine-tune a PLM-based model on a dataset, then evaluate it on several crafted test benchmarks in a zero-shot fashion. The benchmarks are created to evaluate EM models for two types of robustness that are prevalent in real-world settings: domain shift and structural shifts. For domain shift, we conduct out-of-domain evaluations, and for structural shifts, we devise perturbation strategies to modify the structure of tuples without altering the matching outcome.

In addition to the distribution shift and the change in structure between domains, the EM data is highly imbalanced. Table 3.1 shows this phenomenon for several well-known datasets. Data Augmentation (DA) is a common technique to circumvent this problem. In essence, DA makes a model invariant to changes that are less relevant to the task. Although shown effective, we find that DA alone is not sufficient for building a model that is robust to distribution shift. As a remedy, we propose a simple loss function to strengthen the models' ability to put more emphasis on the minority label. We also provide two other recommendations to make PLM-based EM models more robust. Our experiments corroborate that our proposed model is more robust than the

state-of-the-art EM models[1].

Our main contributions can be summarized as follows:

- We investigate the impact of common strategies in EM models from the robustness perspective.
- We design an evaluation framework to test the robustness of EM models under various distribution shifts.
- Based on our findings, we propose simple modifications to Ditto, the state-of-the-art EM model, to build a robust model that surpasses Ditto on in-domain tests as well as out-of-domain tests.

3.2 Preliminaries

The goal of EM is to successfully match mentions, derived from presumably different data sources, that refer to the same real-world entity. In this work, our focus is on the structured data where mentions are stored as tuples [60]. In particular, suppose $A = \{A_0, A_1, ..., A_n\}$ and $B = \{B_0, B_1, ..., B_m\}$ denote two data sources where A_i and B_j represent records in each data source. Each record consists of several attributes — i.e., $A_i = (a_1, a_2, a_3, ...a_k)$. $A_i = B_j$ if and only if both A_i and B_j depict the same real-world entity. EM is characterized as a binary classification task to predict whether two records are identical or not. In a PLM-based EM model, two records are concatenated together, separated by a special token. The model is fine-tuned using a sequence classification objective.

[1]Our code and models are released at `https://github.com/makbn/robem`

3.3 RobEM: A Robust Entity Matcher

We build our model, namely RobEM, atop Ditto, the state-of-the-art EM model that leverages PLMs for identifying identical pairs. Our primary focus is to make a PLM-based model for EM more robust. To this end, we modify Ditto as discussed next. Figure 3.1 shows a high-level overview of our model.

3.3.1 Data Imbalance

In EM datasets, there is an extreme imbalance between examples labeled as negative and positive [6], rendering the negative examples as the majority class. However, Ditto and other prominent EM models use the standard cross-entropy loss, which does not take into account the data imbalance during training. We circumvent this problem by a common method, known as weighted cross-entropy, which is basically the standard cross-entropy, albeit with weights for each class [100]. The weights are typically proportional to the frequency of each class in the training data [50].

3.3.2 Dispensing with Attribute Names

In practice, structured records are collected from anywhere on the web. These records may lack attribute names due to a variety of reasons — e.g., parsing difficulties, or missing information. However, using attribute names is a common practice in EM [51], [60]. As a result, EM models are likely to become impaired when faced with circumstances where attribute headers are not given, curbing their robustness capabilities. To overcome this issue, we dispense with the assumption that attribute names are present in the data to account for such cases. This essentially denotes that our model purely relies on the content of each record for matching.

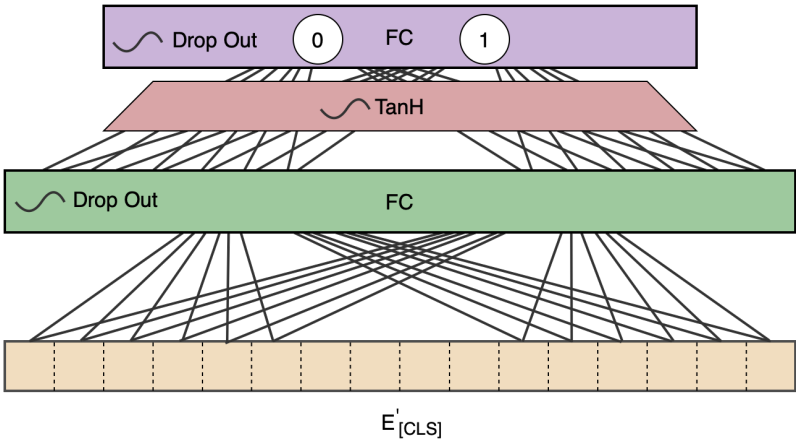


Figure 3.2: Deep classifier head we employ in our model

3.3.3 Classifier Head

In Ditto, the task-specific classification head that projects the output of a PLM to logits is a linear layer. However, due to the complexity of the task, adding a non-linearity in this layer can be helpful. Thus, as it is shown in Figure 3.2, we employ tanh with dropout, following the classifier head in RoBERTa [54], in the task-specific classifier head:

$$y = W_2 \cdot \tanh(W_1 \cdot E'_{\texttt{[CLS]}} + b_1) + b_2. \tag{3.1}$$

3.3.4 Unsupervised EM Baseline

Moreover, we introduce a simple baseline, dubbed UnsupEM, that takes an off-the-shelf PLM to determine whether two tuples are equivalent based on their similarity in the embedding space. In particular, we first feed each tuple into a PLM and take the output of the `[CLS]` token as the representation of the tuple. We then compute the cosine similarity between the two representations. A similarity that is above a certain threshold indicates equivalency.

3.3.5 Serialization

As we mentioned earlier, we use the attributes of each data entry to generate a sentence and solve the EM task as a sentence classification problem. Similar to previous state-of-the-art work [51], we join all attributes to create a sentence representation for each data entry. However, unlike previous work, we only use the value of attributes instead of utilizing both their names and values. While using the name of the columns can be considered an indicator for each attribute's start, it does not help to improve the results in our experiments.

Moreover, repeating the attributes' names for all data entry representations can cause a false positive bias problem, especially with shorter attributes, because adding the same repetitive tokens to each short attribute can make the sequences more similar from the model's point of view. One possible solution is considering names as language model special tokens, which needs a considerable amount of training data to fine-tune the existing special tokens embeddings and learn the new ones. Furthermore, including attributes' names can cause vulnerability when we do not have the same schema in both data sources A and B or the schema is missing.

In our case, we serialize each data entry just by concatenation of attributes using a separator token `ATTR`. To illustrate, for each data entry (A_i, B_j) where $A_i = (a_1, a_2, ..., a_k)$ and $B_j = (b_1, b_2, ..., b_k)$ we have the following serialization:

$S(A_i, B_j)$ ::= `[CLS]` a_1 `ATTR` a_2 `ATTR` ... `ATTR` a_k `[SEP]` b_1 `ATTR` b_2 `ATTR` ... `ATTR` b_k `[SEP]`

The `[SEP]` token is used to separate two sequences during serialization, while the `[CLS]` token is necessary for BERT [18] to encode the two sequence pairs. Then this encoded vector will be used by the classifier head to do the EM task. Our experiments show that excluding the names from the serialization

can contribute to the final robustness of the model.

3.4 EM Robustness Benchmark

To assess the robustness capabilities of EM models, we devise a series of probing tests, simulating the distribution shifts that may arise in real-world scenarios. The first test is domain shift — or out-of-domain — where the domain of test data differs from that of training data. Our main goal here is to understand whether PLM-based EM models have actually mastered the task rather than relying on spurious patterns in the data. To this end, a model, trained on one dataset, is tested against the other datasets.

The next series of tests attempt to replicate schema discrepancies between the tuples from two different data sources. For this purpose, we check the invariance against structural shifts via applying a set of five perturbation operations on the original test data to produce five new test sets. Those perturbation operations are discussed next.

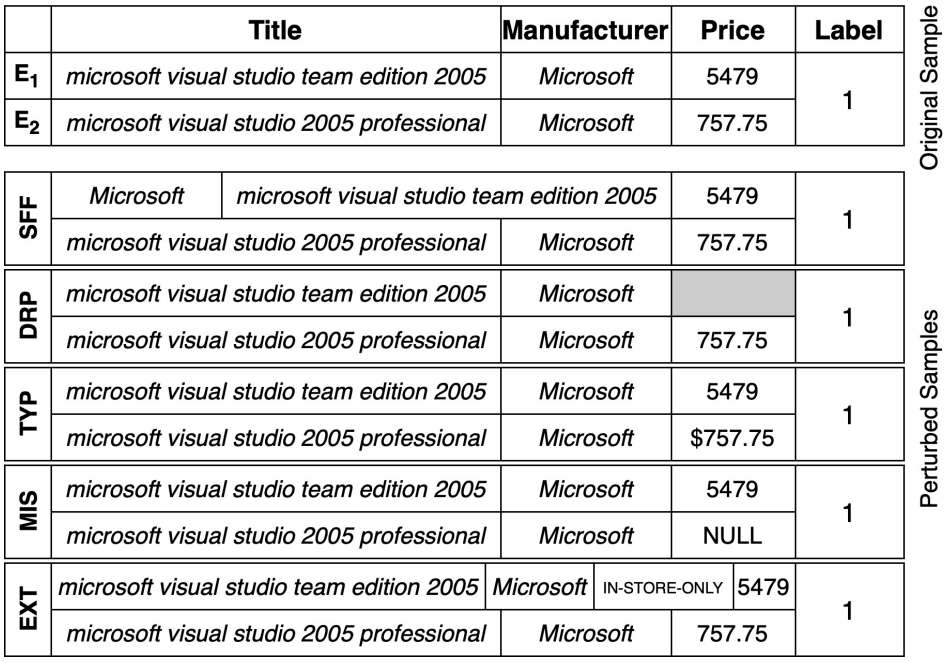


Figure 3.3: Perturbation operations for the schema discrepancy robustness benchmark. Key columns are shown in italic.

3.4.1 Robustness to column order (SFF)

The ordering of columns does not affect the matching result between a pair of tuples. To ensure this condition, we shuffle columns of a tuple for each example in the test data.

3.4.2 Robustness to absence of non-key columns (DRP)

Non-key columns are columns that do not contribute to the matching result between a pair of tuples — e.g., price in Figure 3.3. Matching should remain invariant to inclusion or exclusion of non-key columns. This condition is enforced by randomly dropping one or more non-key columns.

3.4.3 Robustness to missing values (MIS)

The existence of missing values is prevalent when dealing with noisy data sources such as web tables. To imitate this, we randomly replace one or more non-key columns with `NULL`.

3.4.4 Robustness to extraneous columns (EXT)

The presence of irrelevant non-key columns does not affect the matching result between a pair of tuples. We enforce this case by randomly adding columns from other datasets for each test example. EXT test can achieved by adding one or more irrelevant columns to the sample. The new columns can be of type integer, floating-point, or string. We suggest using columns from other datasets rather than random words for text columns.

3.4.5 Robustness to different data types (TYP)

Data entries can be expressed in a handful of ways without changing their semantics. This is especially the case for numerical values. For instance, "1k" is equivalent to "1,000" and "1e3". We curate several hand-crafted rules to randomly convert numbers to different formats to enforce this condition.

3.5 Experiments

3.5.1 Setup

We implemented our models using the Hugging Face transformers library [89]. We select $\text{RoBERTa}_{\text{base}}$ [54], a well-known PLM, that is shown to be effective in EM [51]. In addition, we apply the half-precision floating-point optimization to speed up training and prediction [31], [45]. We follow the hyperparameter

configuration of Ditto for training our models. In particular, we set the maximum sequence length to 256, the batch size to 64, and the number of epochs to 40. The learning rate is set to 3e-5 with a linear decay. All experiments were conducted on a single Nvidia V100 32GB GPU with the batch size set to 64 in both the training and evaluation cycles and the epoch number to 40.

3.5.2 Evaluation Metric

Like previous EM studies, we use the F1 score to evaluate our method and compare it with state-of-the-art models. By taking their harmonic mean, the F1 score combines the precision and recall of a classifier into a single metric, while accuracy only skims the correctly classified observations for both positive and negative samples. Typically, the F1 score is used to compare the performance of two classifiers and determine which is more effective [53]. We calculate F1 score:

$$Precision = \frac{TP}{TP + FP} \tag{3.2}$$

$$Recall = \frac{TP}{TP + FN} \tag{3.3}$$

$$F1score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{3.4}$$

where the TP, FP, and FN are based on the output of the model and datasets' ground truth.

3.5.3 Datasets

We use 8 datasets, introduced in DeepMatcher [60]. The datasets are derived from an entity resolution benchmark [42] as well as the Magellan data reposi-

tory [40]. The datasets are collected from a wide range of domains, including products, publications, and businesses. For all datasets, each example consists of candidate pairs from two structured tables within the same schema. The ratio of matched pairs varies from 9.4% for Walmart-Amazon to 25% for the Company dataset [51]. Generally, there are 1-8 attributes. Table 3.1 presents the size of each dataset[2].

Table 3.1: Datasets size. P. refers to the tuple pairs marked as a match. And N. is the number of non-matched instances. The Attributes column shows the number of attributes in each table being matched.

Dataset	**Train Set (N./P.)**	**Test Set (N./P.)**	**Attributes**
iTunes-Amazon	243 / 78	82 / 27	8
Amazon-Google	6175 / 699	2059 / 234	3
BeerAdvo-RateBeer	228 / 40	77/14	4
DBLP-ACM	6085 / 1332	2029 / 444	4
DBLP-Scholar	14016 / 3207	4672 / 1070	4
Fodors-Zagats	501 / 66	167 / 22	6
Walmart-Amazon	5568 / 576	1856 / 193	5
Abt-Buy	5127 / 616	1710 / 206	3

3.5.4 In-Domain Generalization

We first examine the generalization of EM models to unseen test data that are from the same domain as the training data. For in-domain experiments, we compare our results with two prominent neural entity matching models: Deepmatcher+ [38], an RNN-based model, and Ditto [51], a PLM-based model with the same number of parameters. Since all of these works use the same train, validation, and test splits for their experiments, we directly report the results from the corresponding papers [38], [51].

As presented in Table 3.2, RobEM consistently surpasses Ditto, on all datasets, except for two datasets, iTunes-Amazon (-1.65%), and DBLP-Scholar (-0.22%).

[2]A detailed description of datasets is available at: `https://github.com/anhaidgroup/deepmatcher/blob/master/Datasets.md`

Interestingly, the highest performance gain (+8.99%) is achieved on Abt-Buy. UnsupEM understandably trails all the baselines and RobEM on all datasets because it does not exploit any supervised signals from the data. We also repeat our evaluation with different combinations of our modifications. While the improvements achieved by each modification individually are comparable, the improvement acquired by dispensing with attribute names is more significant on all datasets with an average of 1.8% improvement. The largest improvement is seen for Abt-Buy with an improvement value of 3.95%.

Removing attribute names can primarily reduce false positive by removing unnecessary repetitive tokens in the representation sentences. The impact of having these repetitive tokes is noticeable when the data set has shorter text. In Abt-Buy, one data source has an average of 15 words over three attributes describing an entity. By adding column names, COL, and VAL tokens for each attribute, in the best scenario, nine tokens are added to the serialization of the data source, which matches the entity serialization from the other source. This means, on average, we have a 35.7% matching bias regardless of the label. Also, dispensing with attribute names can help if the same attribute has a different column name in each data source.

The improvement achieved by the weighted cross entropy loss function is 1.55% on average, and the largest improvement is seen for Fodors-Zagats with a higher degree of imbalance. Moreover, the mean advancement achieved by a deep classifier is 1.17%, with the highest impact on Abt-Buy. The results for the deep classifier modification show that datasets with a higher number of attributes are more challenging than datasets with fewer attributes.

3.5.5 Out-of-Domain Generalization

We compare RobEM with UnsupEM and Ditto in out-of-domain experiments, where the model is trained on one dataset (e.g. iTunes-Amazon) and tested

Table 3.2: F1 scores for in-domain experiments. DeepMatcher+ and Ditto results are taken verbatim from [51] (DK).

Dataset	DM+	UnsupEM	Ditto	RobEM	
iTunes-Amazon	91.2	54.54	97.80	96.15	-1.65
Abt-Buy	62.8	21.84	81.69	90.68	+8.99
Amazon-Google	70.7	31.04	74.67	76.64	+1.97
BeerAdvo-RateBeer	78.8	64.70	90.46	93.33	+2.87
DBLP-ACM	98.45	92.77	99.10	**99.21**	+0.11
DBLP-Scholar	94.7	69.77	95.80	95.62	-0.18
Fodors-Zagats	**100.0**	86.95	**100.00**	**100.00**	0.00
Walmart-Amazon	73.6	29.85	83.73	86.68	+2.95

Table 3.3: F1 scores for in-domain experiments using Data Augmentation (DA). DeepMatcher+ and Ditto results are taken verbatim from [51].

Dataset	Ditto (All) + DA	RobEM + DA	
iTunes-Amazon	97.06	**98.18**	+1.12
Abt-Buy	89.33	**90.9**	+1.57
Amazon-Google	75.58	**79.06**	+3.48
BeerAdvo-RateBeer	94.37	**96.55**	+2.18
DBLP-ACM	98.99	99.10	+0.11
DBLP-Scholar	95.60	**95.86**	+0.26
Fodors-Zagats	**100.00**	**100.00**	0.00
Walmart-Amazon	**86.76**	84.61	-2.15

on another dataset (e.g. Amazon-Google). UnsupEM offers a lower bound for supervised models. The vis-a-vis results — i.e., the difference between RobEM and the baselines — that consist of 56 runs are reported in Figure 3.4. Only on 15 cases in total, RobEM lags behind UnsupEM. Furthermore, when trained on BeerAdvo-RateBeer, RobEM struggles most with out-of-domain generalization. On the other hand, the models that are trained on iTunes-Amazon and BeerAdvo-RateBeer significantly outperform Ditto on all 7 datasets.

In total, RobEM trails Ditto on 17 cases. Overall, the improvements of RobEM over UnsupEM and Ditto are statistically significant in 34 and 31 cases, respectively. Finally, we find that UnsupEM is a strong baseline in out-of-domain

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

Table 3.4: Ditto Data augmentations

tests, leading both RobEM and Ditto on 13 tests.

3.5.6 Data Augmentation

Data augmentation (DA) is a long-known technique to counter data imbalance and to boost the generalization capabilities of models. In this section, we aim at evaluating RobEM and Ditto when trained on augmented data. We follow the DA method presented in Ditto. In particular, Ditto DA involves generating augmented data online during training. Each example is augmented via a series of consecutive operations that randomly perturb attributes and tokens. Following Ditto, we generate one augmented sample for each training example. Table 3.4 shows Ditto DA methods.

The in-domain results for DA are presented in the right columns of Table 3.2. DA does not improve the in-domain performance of Ditto on three datasets, iTunes-Amazon, DBLP-ACM, and DBLP-Scholar. However, DA brings in-domain improvements for RobEM on all datasets but two cases. RobEM+DA consistently outperforms Ditto+DA on all datasets, except on Walmart-Amazon.

In the out-of-domain experiments, the results, presented in Figure 3.5, are consistent with our findings in Figure 3.4(b). Specifically, RobEM+DA, trained on iTunes-Amazon, BeerAdvo-RateBeer, and Walmart-Amazon, outperforms Ditto on all 7 datasets by a significant margin, except for one case. DA helps RobEM, trained on Amazon-Google, to achieve better results than Ditto on 5 datasets. However, when comparing RobEM with DA and without, we find

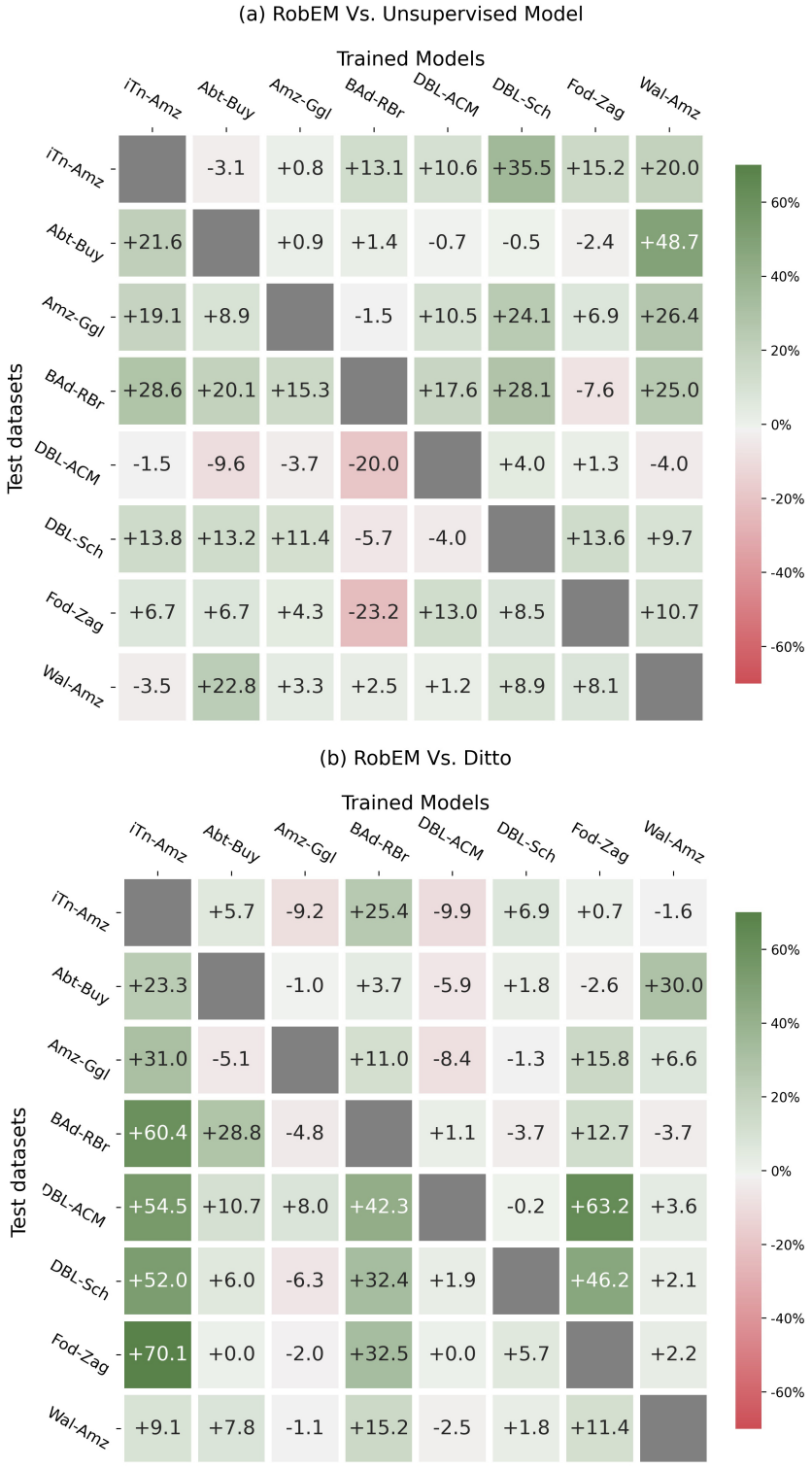


Figure 3.4: Difference between F1 scores of RobEM and two baselines, Ditto (b) and UnsupEM (a) in zero-shot out-of-domain experiments.

Figure 3.5: Difference between F1 scores of RobEM+DA and Ditto+DA in zero-shot out-of-domain experiments.

using DA improves 3 out of 56 tests. This essentially highlights that DA is not necessarily useful for robustness under domain shift.

3.5.7 Schema Discrepancy Generalization

We conduct our robustness test only on the best performing models from the out-of-domain experiment. More precisely, we adopt the models, trained on iTunes-Amazon using DA throughout this section. We surmise that our findings can be extended to other datasets as well. To understand the impact of DA, we employ DA techniques, akin to the ones we adopted to generate the robustness benchmark in Section 3.4. The idea is to imitate the cases of potential distribution shift to teach the model during training. For brevity, we only use the methods for SFF, as defined in Section 3.4, in this experiment to

generate augmented datasets offline:

- **Tuple Swap (SW)**, inspired by Ditto, refers to swapping the left tuple with the right one for each training example.
- **Attribute Shuffle (SF)**, inspired by Ditto, shuffles the order of attributes for each training example.

Table 3.5 shows the results, averaged over 20 runs, for RobEM and Ditto on our robustness benchmark. We report the average performance drop (Δ_{avg}), compared to in-domain results[3]. We observe that Ditto DA method is more robust, compared to SW and SF as it yields the lowest performance drop for both RobEM and Ditto. Also, RobEM is consistently more robust than Ditto across all three DA methods. Interestingly, SFF and DRP are the two most challenging perturbation tests for RobEM and Ditto, respectively. Thus, the STF generates different test data from a dataset for each of the above transformations in addition to the original test set. To generate data, for each sample that consists of a pair of entity rows, we keep one entity row unchanged and apply the above suggestions to the other. For comparison, the difference in means (MD) for each model can be calculated. Table 3.5 shows the STF results for RobEMand Ditto on iTunes-Amazon dataset.

We ran each transformation 20 times for each model and recorded the average. Then we subtract the original result from each to compute the related MD. In all 3 cases, our model achieves smaller negative MD, which shows its robustness to the data alteration.

[3]We attempted to replicate the Ditto results using the official codebase. Our results for Ditto+DA is 93.1%, whereas in the original paper, 97.1% is reported.

Table 3.5: F1 scores on iTunes-Amazon dataset on our EM robustness benchmark. The perturbation operations are defined in Section 3.4.

	DA	in-domain	SFF	DRP	MIS	EXT	TYP	Δ_{avg}
RobEM	SF	96.15	91.6	93.35	96.15	95.74	97.13	-1.35
	SW	98.11	93.83	96.29	97.69	99.25	97.95	-1.10
	Ditto	98.18	96.8	98.11	98.11	97.74	98.11	-0.40
Ditto	SF	96.42	95.47	91.94	96.42	96.42	93.69	-1.63
	SW	94.54	90.29	91.78	92.85	92.96	94.54	-2.05
	Ditto	93.10	92.81	89.47	90.56	93.76	93.10	-1.16

Table 3.6: The 9 datasets divided into 4 categories of domains summarized by Ditto [51].

Datasets	Domains
Amazon-Google, Walmart-Amazon	Software / Electronics
Abt-Buy, BeerAdvo-RateBeer	Product
DBLP-ACM, DBLP-Scholar, iTunes-Amazon	Citation / Music
Company, Fodors-Zagats	Company / Restaurant

3.6 Summary

In this chapter, we investigated the robustness capabilities of EM models under domain shifts and structural shifts. We prescribe simple guidelines to build robust models that are suitable for deployment in the wild. Our proposed model outperforms the state-of-the-art PLM-based EM model under distribution shift. We hope that our findings spurs development of more robust EM models. Also, our robustness benchmark can be a basis for a thorough assessment of future EM models.

We explore unstructured data, complex data augmentation techniques and other forms of distribution shift in the next chapter.

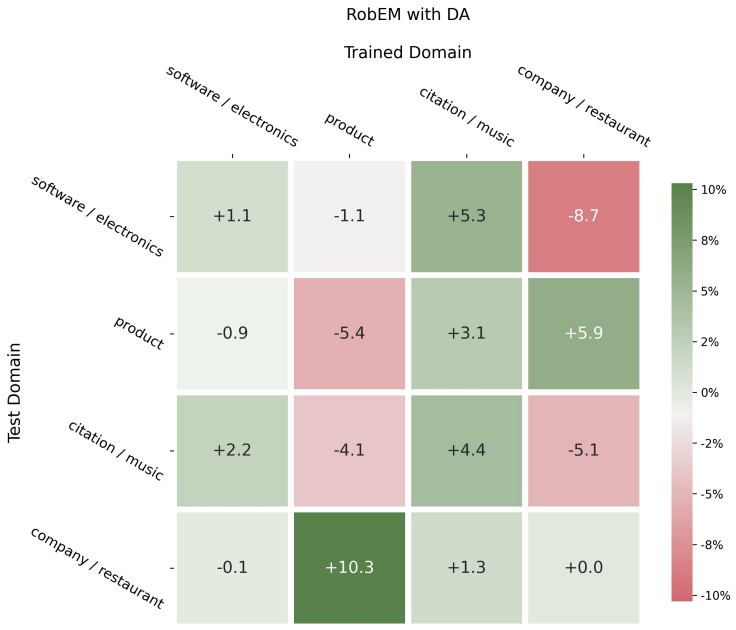


Figure 3.6: Out-of-domain F1 scores When DA is applied.

Chapter 4

Unstructured Entity Matching

4.1 Introduction

As a recap of our work in the previous chapter, our study shows the power of using PLMs such as RoBERTa [54] and BERT [18] for the EM task while investigating the impact of common strategies from the robustness perspective. We also demonstrate that the PLMs achieve a remarkable performance through a fine-tuning paradigm for structured EM. We suggest a few simple modifications to make Ditto more robust on in-domain and out-of-domain tests. We further propose a benchmark framework to evaluate the robustness of structured EM models in a real-world setting which is expected to be noisy. All our studies and research in the previous chapter are dedicated to a specific type of data known as structured data, where data are in some tabular formats. There has been a large body of work on EM for structured data, and our work focuses on using pre-trained language models.

Unstructured data, on the other hand, have not been explored widely through the lens of entity matching. A recent study reports that 95% of businesses cite the problem of managing unstructured data as a challenge for their business [46]. This emphasizes the importance of extracting and linking entities in unstructured sources to their referents or identifying emerging entities.

In this chapter, we challenge the power of the proposed EM model, RobEM, and evaluate it for the EM task on unstructured data. For unstructured data, to the best of our knoeledge, no large annotated data exists at the time of this writing; thus, we develop a dataset, called News-Pair dataset, based on CoNLL En 2003 [75], by pairing documents that refer to the same real-world entity. Documents can have multiple mentions and those mentioned can refer to different entities (as shown in **Example 4.1.1**). For our task, each document in our collection contains a primary mention, and each document pair has a label that shows if their primary mentions refer to the same real-world entity.

Example 4.1.1 Sample data from the generated unstructured EM dataset based on CoNLL documents

- **Left Document**:

  ...fire a shot into the top corner. It looked like turning into a rout as Hwang Sun Hong rapidly ... minutes later direct from a corner kick that **<MEN> Korean <MEN>** goalkeeper Kim Byung reached with one hand but failed to keep out. With 65 minutes gone...

- **Right Document**:

  ...(Netherlands) 41.58; 7.Jin Hua (China) 41.59; 8.Alena Koroleva (Russia) 41.64; 9.Chris Witty... Gerard Van Velde (Netherlands) 1:16.63 8.Kim Yoon-man **<MEN> South Korea <MEN>** 1:16.75 9.Jeremy Wotherspoon (Canada) 1:16.75 10.Miyabe Yasunori (Japan) 1:16.86 Women's 1,000 metres...

- **Label**: Match

An example document pair is shown in **Example 4.1.1**. We can see that the surface texts can be different, but the mentions can reference the same entity. More details of the dataset can be found in Section 4.4.1.

Unstructured EM is generally more challenging than structured EM because the attributes that contribute to the linking of the mentions are not explicitly listed. Since PLMs accept token sequences (i.e., text) as input, transforming a candidate pair into a sequence of tokens is a critical challenge in both struc-

tured and unstructured data. For structured data, as discussed in Chapter 3, RobEM appends the values of columns that describe an entity, forming a sequence with a separator token between column values. Each column value gives a feature of the entity and may possess information that helps with the entity matching. For unstructured data, textual documents are already in the form of a sequence of tokes. However, there are many different ways of extracting features from a textual document, and not all of them contribute to the task of disambiguation and matching. We propose a context-based approach to serialize textual documents to provide input to RobEM.

Finally, as an alternative model and a baseline for our comparison, we implement a serialization step for Ditto, based on their serialization for the structured data, and follow their guidelines to prepare the data for their model. We evaluate both Ditto and our model on the same data and under the same conditions. As noted n the previous chapter, despite their success and popularity, PLMs are no panacea and tend to learn the underlying spurious patterns in data and are more likely to fail under different circumstances, such as distribution shift [7], [56], [63]. For this reason, we extend our experiments further to challenge both unstructured models in an out-of-domain environment.

For the out-of-domain experiment, we change the natural label distribution of the dataset by adding some new samples. These samples are selected from the minority class, and for each sampled pair, more documents that mention those entities are collected from news sources on the web. Adding new pairs to the dataset gives us a balanced dataset, allowing us to evaluate the models under a distribution shift.

Our main contributions in this chapter can be summarized as follows:

- We extend our model by applying modifications to the serialization step to handle the task of EM for unstructured data. We do the same for Ditto based on their serialization for structured data.

- We develop a new dataset based on documents from CoNLL En 2003 for entity matching on unstructured data. To further generate a new balanced dataset, we collect online news documents by sampling for the minority class.
- Using the developed unstructured datasets, we evaluate both Ditto and RobEM extensively to assess their performances on the unstructured EM task.

4.2 Problem Definition

We define a mention to be a consecutive sequence of tokens in a textual document that refers to a real-world entity. For example, a mention may denote a noun, a noun phrase, or a pronoun. Let us assume we have two documents:

$$D_L = w_1, w_2, w_3, ..., w_{n-1}, m_l, w_{n+1}, ..., w_m, \text{ and}$$

$$D_R = w'_1, w'_2, w'_3, ..., w'_{n-1}, m_r, w'_{n+1}, ..., w'_k,$$

where D_L has m tokens and D_R has k tokens. Document D_L contains mention m_l and docuemnt D_R contains mention m_r. The two mentions can have the exact same surface text or may be different. The goal of an unstructured EM is to find out whether m_l and m_r refer to the same real-world entity, denoted as $Ref(m_l) = Ref(m_r)$. Since many entities are not present in a knowledge base, our goal is to do this entity matching without linking the mentions to a knowledge base.

Previous works define unstructured EM as an EM task where the features are long text. Company [40] and Abt-Buy [42] are two unstructured databases that have been widely used in the literature, and none of them are a good example of unstructured data for EM. The Company dataset consists of document pairs $p_i = (a, b)$ where document a is the text of the Wikipedia page of a company

and document b is the home page of the same company [60]. Both a and b may contain mentions of the entity, but those mentions are not annotated, and each document can have mentions of many other entities as well. For the Abt-Buy dataset, we have a table with three attributes for each left and right entity pair. The first attribute is a number and the third attribute is a short text. It is only the second attribute that has a long text.

4.3 Unstructured Entity Matching

We want to study if our proposed model, RobEM, can be extended for an unstructured EM, maybe by adding a new serialization layer. Note that RobEM considers attribute values as entity features and appends them to create a sentence that represents the entity.

4.3.1 Serialization

For each mention, we need to extract features from the document that contains the mention. Various forms of features can be extracted from a textual context. Similar to previous studies, a bag of words model may be used to collect the features of an entity based on its mentions. Those features may be gathered from the entire document [16], [30] or a smaller context window of tokens surrounding the mention [32], [47], [58].

The co-occurrence of words in the contexts of or around two mentions may indicate that two mentions are related. In most cases, the whole document can be too long to be considered as a context and cannot be encoded with the limited input size of PLMs. We consider a window of text around a mention as its context and serialize the surface text of the mention and its left context and right context, as shown in Figure 4.1. For a document pair (D_L, D_R), we have the following serialization:

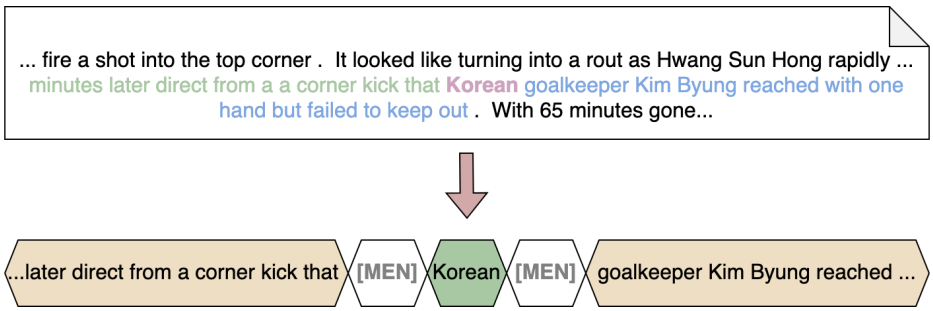


Figure 4.1: RobEM unstructured serialization. For each document D_L and D_R, we serialize the mention and its surrounding context window by appending them and annotating the mention with the special token `[MEN]`.

$$RobEM_S(D_L, D_R) ::= \texttt{[CLS]}\ D^L_{L\_CTX}\ \texttt{[MEN]}\ m_l\ \texttt{[MEN]}\ D^L_{R\_CTX}$$
$$\texttt{[SEP]}\ \ D^R_{L\_CTX}\ \texttt{[MEN]}\ m_r\ \texttt{[MEN]}\ D^R_{R\_CTX}\ \texttt{[SEP]}.$$

Likewise, we develop a similar serialization layer for Ditto following their guidelines:

$$Ditto_S(D_L, D_R) ::= \texttt{[CLS]}\ \texttt{[COL]}\ \texttt{L\_CTX}\ \texttt{[VAL]}\ D^L_{L\_CTX}\ \texttt{[COL]}\ \texttt{MEN}\ \texttt{[VAL]}\ \ m_l$$
$$\texttt{[COL]}\ \texttt{R\_CTX}\ \texttt{[VAL]}\ \ D^L_{R\_CTX}\ \ \texttt{[SEP]}$$
$$\texttt{[COL]}\ \texttt{L\_CTX}\ \texttt{[VAL]}\ \ D^R_{L\_CTX}\ \texttt{[COL]}\ \texttt{MEN}\ \texttt{[VAL]}\ \ m_r\ \texttt{[COL]}$$
$$\texttt{R\_CTX}\ \texttt{[VAL]}\ \ D^R_{R\_CTX}\ \texttt{[SEP]}.$$

4.3.2 Loss function

As mentioned in Chapter 3, the weighted cross-entropy loss function addresses the problem of overfitting deep learning models due to the data imbalance [100]. For unstructured EM, data imbalance is a bigger problem because the combination of non-matching mentions can be exponentially more than the matching ones. Hence, we use the weighted cross-entropy loss function here to amplify the impact of the minority class on the model.

4.4 News-Pair: Unstructured EM Dataset

We are not aware of a public dataset for the problem of unstructured EM. Hence, we develop News-Pair, an unstructured EM dataset, based on CoNLL En 2003 [75] documents and AIDA [34] annotations. Each document usually contains mentions of multiple named entities. Therefore, datasets like Company [40] that only contain a text body without any annotation of the mention are less useful. As an example, consider the following document pair from Company dataset:

Example 4.4.1 A document pair from the training set of Company dataset with the label 0 or non-match.

Document A:
Fremantle is a British multinational television production and distribution company based in London. Fremantle takes its name from Fremantle International, acquired by predecessor company All American Television in 1994. Pearson Television was renamed FremantleMedia on 20 August 2001, following the 2000 merger of Pearson ...
Document B:
... now Randstad in numbers 6 695 teachers and support staff in work 3 938 schools needed our help 165 654 jobs filled in 2013 14 130 specialist consultants delivering teachers the best in CPD want to enhance your skills and set your CPD targets for the year take advantage of our free CPD offering register now reversing the teacher exodus read our latest report on the changing face of education recruitment in the UK ...

Based on the label provided in the dataset, these two documents are not referring to the same entity. Although the first document is about Fremantle company and the second document is about Randstad, both documents make references to other entities as well. For example, consider the word **British** in Document A and **UK** in Document B. What can we say about these two mentions and if they are referring to the same entity? The problem is the dataset does not have any information about the exact mention that are matched.

4.4.1 Data format

Our News-Pair consists of 100,000 pairs of documents in the train set, 75,048 pairs in the validation set, and 59,130 pairs in the test set. Each document pair is generated by appending two documents extracted from CoNLL dataset.

We first cluster all mentions using their linked references. Then, for each mention, we extract a context window from its CoNLL document and save it as the new document. For generating 'positive' or 'match' samples, we select all combinations of mentions in different documents that reference the same entity. In more than half of the positive pairs, the surface text for the left and right mentions are identical, while many other pairs have non-identical mentions and still refer to the same entity. We divide the positive samples into mentions with identical surface text, referred to as **SM**, and mentions where the surface texts are not identical, referred to as **NM**. NM includes mentions that are non-identical but still reference the same entity.

Generating negative samples is more challenging. The combination of document pairs with non-matching references can be large, and this can result in a highly skewed dataset. To prevent this problem and to maintain the natural distribution of negative samples, we first generate samples where the left and right mentions have the same surface text but referencing different entities. This set is denoted as **SN**. For the final category, referred to as **NN**, we follow the previous steps and initially generate a massive set of mention pairs that have different surface texts and refer to different entities. We randomly select a subset of these samples with a size that matches the size of the SN set. This procedure is followed for the training, validation, and test sets. In each case, the documents are taken from the related sets in the oNLL dataset. Table 4.1 shows the size of each set in News-Pair dataset, divided into four categories, SM, SN, NM, and NN.

Table 4.1: Number of document-pairs in each mention type. **SM** includes pairs with the same surface text and the same reference, **SN** includes pairs with the same surface text but refer to different entities, **NM** includes pairs with different surface texts and the same reference, and **NN** includes pairs with different surface texts and refers to different references.

Set / Type	SM	SN	NM	NN	Total
Train	47,401	5,541	41,584	5,474	100,000
Validation	37,364	5,066	27,552	5,066	75,048
Test	27,656	5,784	19,906	5,784	59,130
Sum					234,178

4.4.2 Data Augmentation

Data augmentation (DA) has recently seen an increased interest in NLP due to more work in low-resource domains and new tasks, as well as the popularity of large-scale neural network models that require large amounts of training data [24].

In our case, we need to add new samples to our minority class to balance the dataset. To accomplish this, we need to add new negative samples to SN where the two mentions have the same surface text, but they refer to different entities. As mentioned earlier, many NN samples can be generated by combining mentions with different surface text that refer to different references. Increasing the number of these samples can cause the dataset to have an implicit relationship between the surface texts of the mentions and the non-match label, and the model can learn such relationships between non-matching pairs.

Data augmentation in Ditto, as described in Chapter 3, is not much of an option here. For a long text, changing or removing a token cannot augment a new high-quality sample and likely will generate a repetitive sample that can lead to overfitting on the train data.

In order to generate new high-quality samples, we use the available information about referent entities to search and scrap new data. We have 1065 pairs in

Table 4.2: Number of document-pairs in each set of balanced News-Pair dataset.

Set / Type	SM	SN	NM	NN	Total
Train	47,401	41,541	41,584	47,974	178,500
Validation	37,364	27,066	27,552	37,566	129,548
Test	27,656	19,784	19,906	28,784	96,130
Sum					404,178

our dataset where mentions have the same surface text, but those mentions reference different entities. For each of these pairs, we extract the surface text of the mention and both the left and the right references. Then we convert the references to Wikidata QIDs using the Wikipedia pages of the references. We create a query for each entity using its English description and entity type in Wikidata. Having two queries for each pair of mentions, we search for new online news documents in Google News and generate two sets of news for each pair. Figure 4.2 shows the steps for generating new documents.

Sorting through the news articles News articles returned by Google can be noisy. For instance, for the query 'Wall Street, a street in Manhattan,' the top 5 returned results are all about the 'Wall Street Stock Exchange.' We do not have any public information on how Google matches a query with the news, but this probably happens due to its high prior probability [43], [67].

To better understand the problem, we repeat our experiment with different queries and review the returned results for each query. While the returned set for each query is different, some results are common between different query formulations of the same entity, and those documents generally have a closer relationship with the entity. Therefore, we manually examine the returned result for each query and determine its correctness. To do that, we select a set of 20 random pairs from the SN category of our dataset. We retrieve the top three news for each pair, three news for the left entity, and three news for the right entity, using a description-based query (referred to as Q_1) and

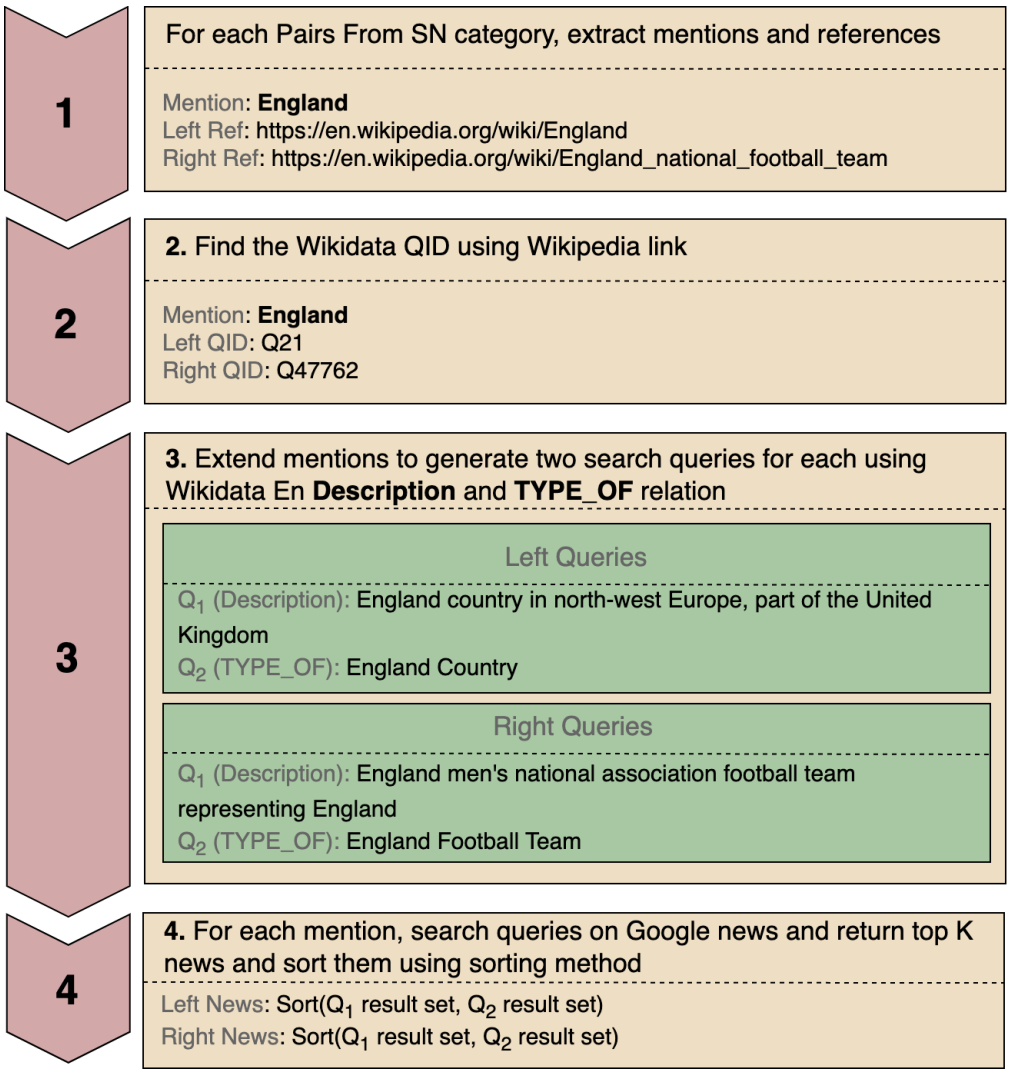


Figure 4.2: News-Pair DA for generating new negative samples using existing pairs and new online news documents. QIDs are extracted from the Wikipedia URL of references by parsing the HTML content. Each query returns a set of relevant news articles.

a TYPE_OF query (referred to as Q_2). We compute the accuracy under the following three scenarios:

- **Description Priority** In this experiment, we prioritize retrieved news by the description-based query (Q_1) for each entity in a pair. If the number of retrieved news by Q_1 is insufficient, we select and add news from the results of the TYPE_OF query (Q_2).
- **TYPE_OF Priority** In contrast to **Description Priority**, In this case, we prioritize retrieved news by the TYPE_OF query (Q_2) results. Likewise, we select and add news retrieved by Q_1, if needed.
- **Intersection** This method combines results from both sets giving priority to the results of the description-based query. We change the order of documents in the description-based results set if we have a news document that appeared in both sets. For the news in the intersection, the ordering is based on their rank on the description-based set. Figure 4.3 shows an example of sorting using the Intersection method.

We define accuracy as the percentage of selected results that matches the query. Our evaluation shows that the best accuracies for Description Priority and TYPE_OF Priority are %80 and %72, respectively. The accuracy for intersection is %82; therefore, we use this method to generate the new document pairs. Table 4.2 shows the size after adding our new pairs. We also generate the augmented data for the test and validation set for our out-of-domain test.

4.5 Experiments

4.5.1 Setup

We use structured RobEM model from Chapter 3 and extend it by adding a layer for unstructured data. Similar to our experiments on structured data,

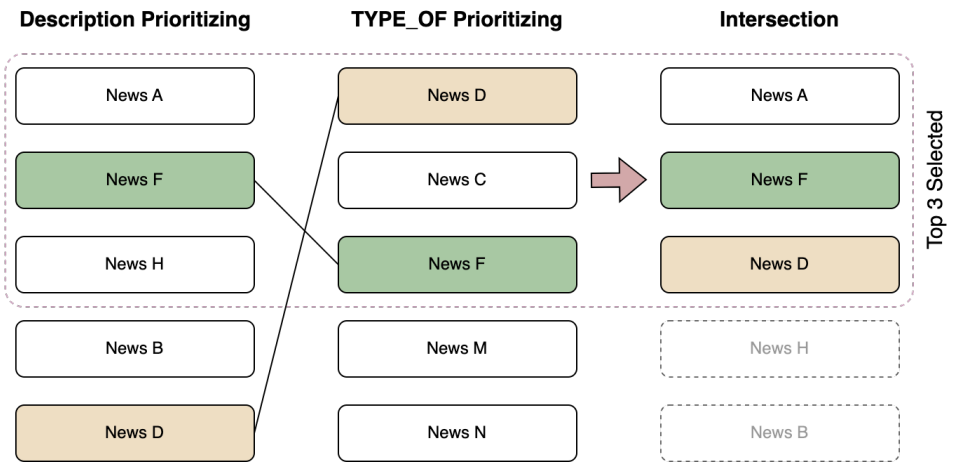


Figure 4.3: Sorting techniques for selecting new news documents for balanced News-Pair dataset.

we set the maximum sequence length to 256, the batch size to 64, and the number of epochs to 40. The learning rate is set to 3e-5 with a linear decay. All experiments were conducted on a single Nvidia V100 32GB GPU with the batch size set to 64 in both the training and evaluation cycles and the epoch number to 40.

4.5.2 Dataset

We evaluate our models on News-Pair and balanced News-Pair datasets and compare the model performance with that of Ditto. The details of these datasets are available in Tables 4.1 and 4.2.

4.5.3 In-Domain Generalization

In this experiment, the models are trained and tested on datasets that follow the same distributions. For example, a model trained on News-Pair is also tested on News-Paid. We compare our results with Ditto, a state-of-the-art model with the same number of parameters.

The authors of Ditto suggest a Domain Knowledge (DK) optimization step

Table 4.3: F1 scores for in-domain experiments. 'Pos. class' row reflects the F1 score based on SM and NM categories. 'Neg. class' represents the SN and NN negative samples.

	Ditto	RobEM		Ditto+DK	RobEM+DK	
Overall	95.32	97.37	+2.05	96.80	97.24	+0.44
Pos. Class	95.32	97.37	+2.05	96.8	97.24	+0.44
Neg. Class	78.69	88.96	+10.27	85.85	87.95	+2.1

where mentions in text are annotated with their types. The idea is to pre-process the serialized inputs by tagging informative spans (e.g., product ID, person name), inserting special tokens (e.g., ID, PERSON), and normalizing specific spans (e.g., numbers) [51]. They provide two versions for DK, referred to as General and Product, where General is applicable to all types of entities and Product is specifically for product entities. We conduct our experiments using their 'General DK' for both models and report our results. Table 4.3 presents the F1 score for our in-domain test using RobEM and Ditto. In all cases, with and without DK, our model consistently outperforms Ditto. DK improves the F-score of Ditto up to +8% for the non-matching class and by +1.92% overall, but it does not improve the performance of RobEM. We think the reason is that our model captures this extra information using a deep classifier without DK.

4.5.4 Out-Of-Domain Generalization

Numerous works employ the fine-tuning technique to improve their performance over a dataset. However, different factors may contribute to these improvements with the model learning the task, data trends, and/or the vocabulary. Recent research indicates that a high accuracy on a test set drawn from the same distribution as the training set is not a sign that the model has mastered the task. [59]. In our out-of-domain experiment, we aim at understanding the performance of the models when trained on one dataset and tested on another dataset with a different distribution. To this end, we use

Table 4.4: F1 scores for the out-of-domain experiment. We train both Ditto and RobEM models on each dataset and then test each model on another dataset.

		Test dataset			
		News-Pair		**Balanced News-Pair**	
		Ditto	**RobEM**	**Ditto**	**RobEM**
Train dataset	**News-Pair**			96.40	96.63
	Balanced News-Pair	94.29	94.88		

News-Pairs and balanced New-Pair for this experiment. First, we train both models on the training set of one dataset, then evaluate it with the test set of the other dataset. For example, we train RobEM on News-Pair and test it on balanced News-Pair.

Table 4.4 shows the result for the out-of-domain experiment. RobEM scores 96.63% in case the training set is News-Pair and the test set is balanced News-Pair, which is +0.23 higher than Ditto. Also, when we use our balanced News-Pair as the training dataset and News-Pair as the test dataset, our model achieves 94.88%, which is better than Ditto.

4.6 Summary

In this chapter, we extend our structured EM model for unstructured data and evaluate the performance of PLM-based models. Similar to structured EM, our experiments show that PLM-based models can be effective for unstructured data due to their ability to understand the language context. We also develop two new unstructured matching datasets, one using documents from CoNLL and another with additional new documents. In-Domain and Out-of-Domain experiments show that our proposed modification in the previous chapter can be generalized for the unstructured EM task.

Chapter 5

Conclusion and Future direction

In this study, we present RobEM, a robust EM solution, using transformer-based language models. RobEM uses a simple architecture to leverage pre-trained LMs and implements further optimizations, including loss function, data augmentation, and classification head, to improve the robustness of the model. Our results show that the proposed model improves the robustness of existing EM solutions on standard structured datasets while maintaining the performance. The in-domain and out-of-domain experiments show a marginal advantage over a state-of-the-art model. Also, we provide a robustness benchmark framework for structured EM to help researchers thoroughly test their proposed EM solution under distribution shifts.

We extend our model for unstructured EM. To evaluate our PLM-based model, we develop a new dataset based on the commonly used dataset, CoNLL En 2003. We then extend the proposed dataset using new documents that are scraped from online news to build a balanced version. Our experiments on both datasets show the effectiveness of the proposed modifications.

We plan to investigate our choices for additional optimizations in future works, including complex data augmentation techniques. Also, we plan to investigate the techniques to optimize the model speed to shorten the inference time in settings where the number of possible combinations can be extremely high.

References

[1] S. Abdallah, K. Shaalan, and M. Shoaib, "Integrating rule-based system with classification for arabic named entity recognition," vol. 7181 LNCS, 2012. DOI: 10.1007/978-3-642-28604-9_26.

[2] G. Aguilar, S. Maharjan, A. P. López-Monroy, and T. Solorio, "A multi-task approach for named entity recognition in social media data," in *Proceedings of the 3rd Workshop on Noisy User-generated Text*, Copenhagen, Denmark: Association for Computational Linguistics, 2017, pp. 148–153. DOI: 10.18653/v1/W17-4419. [Online]. Available: https://aclanthology.org/W17-4419.

[3] M. Akbarian Rastaghi, E. Kamalloo, and D. Rafiei, "Probing the robustness of pre-trained language models for entity matching," in *Proceedings of the 31st ACM International Conference on Information; Knowledge Management*, ser. CIKM '22, Atlanta, GA, USA: Association for Computing Machinery, 2022, pp. 3786–3790, ISBN: 9781450392365. DOI: 10.1145/3511808.3557673. [Online]. Available: https://doi.org/10.1145/3511808.3557673.

[4] O. Ali and N. Cristianini, "Information fusion for entity matching in unstructured data," vol. 339 AICT, 2010. DOI: 10.1007/978-3-642-16239-8_23.

[5] F. Azzalini, M. Renzi, and L. Tanca, "A deep-learning-based blocking technique for entity linkage," in *Database Systems for Advanced Applications: 25th International Conference, DASFAA 2020, Jeju, South Korea, September 24–27, 2020, Proceedings, Part I*, Jeju, Korea (Republic of): Springer-Verlag, 2020, pp. 553–569, ISBN: 978-3-030-59409-1. DOI: 10.1007/978-3-030-59410-7_37. [Online]. Available: https://doi.org/10.1007/978-3-030-59410-7_37.

[6] N. Barlaug and J. A. Gulla, "Neural networks for entity matching: A survey," *ACM Transactions on Knowledge Discovery from Data*, 2021. DOI: 10.1145/3442200. [Online]. Available: https://doi.org/10.1145/3442200.

[7] E. M. Bender, T. Gebru, A. McMillan-Major, and S. Shmitchell, "On the dangers of stochastic parrots: Can language models be too big?,"

2021. DOI: 10.1145/3442188.3445922. [Online]. Available: https://doi.org/10.1145/3442188.3445922.

[8] I. Bhattacharya and L. Getoor, "Collective entity resolution in relational data," *ACM Transactions on Knowledge Discovery from Data*, vol. 1, 1 2007, ISSN: 15564681. DOI: 10.1145/1217299.1217304.

[9] M. Bilenko and R. J. Mooney, "Adaptive duplicate detection using learnable string similarity measures," 2003. DOI: 10.1145/956750.956759. [Online]. Available: https://doi.org/10.1145/956750.956759.

[10] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *Transactions of the Association for Computational Linguistics*, vol. 5, 2017. DOI: 10.1162/tacl_a_00051.

[11] T. B. Brown, B. Mann, N. Ryder, *et al.*, "Language models are few-shot learners," *Advances in Neural Information Processing Systems*, vol. 2020-December, 2020, ISSN: 10495258. [Online]. Available: https://proceedings.neurips.cc/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

[12] U. Brunner and K. Stockinger, "Entity matching on unstructured data : An active learning approach," en, in *Proceedings of the 6th SDS*, Swiss Conference on Data Science, Berne, Switzerland, 14 June 2019, IEEE, 2019, ISBN: 978-1-7281-3105-4. DOI: 10.21256/zhaw-3197. [Online]. Available: https://digitalcollection.zhaw.ch/handle/11475/17388.

[13] U. Brunner and K. Stockinger, "Entity matching with transformer architectures - a step forward in data integration," en, in *Proceedings of EDBT 2020*, 23rd International Conference on Extending Database Technology, Copenhagen, 30 March - 2 April 2020, OpenProceedings, 2020, ISBN: 978-3-89318-083-7. DOI: 10.21256/zhaw-19637. [Online]. Available: https://digitalcollection.zhaw.ch/handle/11475/19637.

[14] J. P. Chiu and E. Nichols, "Named entity recognition with bidirectional lstm-cnns," *Transactions of the Association for Computational Linguistics*, vol. 4, 2016. DOI: 10.1162/tacl_a_00104.

[15] J. B. Copas and F. J. Hilton, "Record linkage: Statistical models for matching computer records," *Journal of the Royal Statistical Society. Series A (Statistics in Society)*, vol. 153, 3 1990, ISSN: 09641998. DOI: 10.2307/2982975.

[16] S. Cucerzan, "Large-scale named entity disambiguation based on wikipedia data," 2007.

[17] S. Das, P. G. Suganthan, A. Doan, *et al.*, "Falcon: Scaling up hands-off crowdsourced entity matching to build cloud services," vol. Part F127746, 2017. DOI: 10.1145/3035918.3035960.

[18] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, 2019. DOI: 10.18653/v1/N19-1423. [Online]. Available: https://aclanthology.org/N19-1423.

[19] M. Ebraheem, S. Thirumuruganathan, S. Joty, M. Ouzzani, and N. Tang, "Distributed representations of tuples for entity resolution," vol. 11, 2018. DOI: 10.14778/3236187.3236198.

[20] T. Eftimov, B. K. Seljak, and P. Korošec, "A rule-based named-entity recognition method for knowledge extraction of evidence-based dietary recommendations," *PLoS ONE*, vol. 12, 6 2017, ISSN: 19326203. DOI: 10.1371/journal.pone.0179488.

[21] A. K. Elmagarmid, P. G. Ipeirotis, and V. S. Verykios, "Duplicate record detection: A survey," *IEEE Trans. on Knowl. and Data Eng.*, 2007. DOI: 10.1109/TKDE.1990.10000. [Online]. Available: https://www.computer.org/csdl/journal/tk/1990/01/k0001/13rRUxBa56h.

[22] W. Fan, X. Jia, J. Li, and S. Ma, "Reasoning about record matching rules," *Proc. VLDB Endow.*, 2009. DOI: 10.14778/1687627.1687674. [Online]. Available: https://doi.org/10.14778/1687627.1687674.

[23] D. Farmakiotou, V. Karkaletsis, J. Koutsias, G. Sigletos, C. D. Spyropoulos, and P. Stamatopoulos, "Rule-based named entity recognition for greek financial texts," 2000.

[24] S. Y. Feng, V. Gangal, J. Wei, *et al.*, "A survey of data augmentation approaches for nlp," *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, 2021. DOI: 10.18653/v1/2021.findings-acl.84. [Online]. Available: https://aclanthology.org/2021.findings-acl.84.

[25] S. Garg and G. Ramakrishnan, "Bae: Bert-based adversarial examples for text classification," *EMNLP 2020 - 2020 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, 2020. DOI: 10.18653/v1/2020.emnlp-main.498.

[26] A. Ghaddar and P. Langlais, "Robust lexical features for improved neural network named-entity recognition," in *Proceedings of the 27th International Conference on Computational Linguistics*, Santa Fe, New Mexico, USA: Association for Computational Linguistics, 2018, pp. 1896–1907. [Online]. Available: https://aclanthology.org/C18-1161.

[27] A. Gionis, P. Indyk, and R. Motwani, "Similarity search in high dimensions via hashing," 1999.

[28] C. Gokhale, S. Das, A. H. Doan, *et al.*, "Corleone: Hands-off crowdsourcing for entity matching," 2014.

[29] A. Graves, M. Liwicki, S. Fernández, R. Bertolami, H. Bunke, and J. Schmidhuber, "A novel connectionist system for unconstrained handwriting recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, 5 2009, ISSN: 01628828. DOI: 10.1109/TPAMI.2008.137.

[30] S. Guo, M.-W. Chang, and E. Kiciman, "To link or not to link? a study on end-to-end tweet entity linking," in *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Atlanta, Georgia: Association for Computational Linguistics, 2013, pp. 1020–1030. [Online]. Available: https://aclanthology.org/N13-1122.

[31] A. Haidar, S. Tomov, J. Dongarra, and N. J. Higham, "Harnessing gpu tensor cores for fast fp16 arithmetic to speed up mixed-precision iterative refinement solvers," 2019. DOI: 10.1109/SC.2018.00050.

[32] X. Han, L. Sun, and J. Zhao, "Collective entity linking in web text: A graph-based method," in *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '11, Beijing, China: Association for Computing Machinery, 2011, pp. 765–774, ISBN: 9781450307574. DOI: 10.1145/2009916.2010019. [Online]. Available: https://doi.org/10.1145/2009916.2010019.

[33] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[34] J. Hoffart, M. A. Yosef, I. Bordino, *et al.*, "Robust disambiguation of named entities in text," 2011.

[35] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities.," *Proceedings of the National Academy of Sciences*, vol. 79, no. 8, pp. 2554–2558, 1982. DOI: 10.1073/pnas.79.8.2554. eprint: https://www.pnas.org/doi/pdf/10.1073/pnas.79.8.2554. [Online]. Available: https://www.pnas.org/doi/abs/10.1073/pnas.79.8.2554.

[36] A. Jain, S. Sarawagi, and P. Sen, "Deep indexed active learning for matching heterogeneous entity representations," *Proceedings of the VLDB Endowment*, 2021. DOI: 10.14778/3485450.3485455. [Online]. Available: https://dl.acm.org/doi/10.14778/3485450.3485455.

[37] L. Jin, C. Li, and S. Mehrotra, "Efficient record linkage in large data sets," 2003. DOI: 10.1109/DASFAA.2003.1192377.

[38] J. Kasai, K. Qian, S. Gurajada, Y. Li, and L. Popa, "Low-resource deep entity resolution with transfer and active learning," *ACL 2019 - 57th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference*, 2020. DOI: 10.18653/v1/p19-1586. [Online]. Available: https://aclanthology.org/P19-1586.

[39] S. Knott, *George h. w. bush: Life before the presidency*, 2018. [Online]. Available: https://millercenter.org/president/bush/life-before-the-presidency.

[40] P. V. Konda, *Magellan: Toward building entity matching management systems*. 2018. [Online]. Available: http://www.vldb.org/pvldb/vol9/p1197-pkonda.pdf.

[41] H. Köpcke and E. Rahm, "Training selection for tuning entity matching," *Proceedings of the Sixth International Workshop on Quality in Databases and Management of Uncertain Data (QDB/MUD'08)*, October 2016 2008, ISSN: 15740846.

[42] H. Köpcke, A. Thor, and E. Rahm, "Evaluation of entity resolution approaches on real-world match problems," *Proceedings of the VLDB Endowment*, 2010. DOI: 10.14778/1920841.1920904. [Online]. Available: https://doi.org/10.14778/1920841.1920904.

[43] W. Kraaij, T. Westerveld, and D. Hiemstra, "The importance of prior probabilities for entry page search," 2002. DOI: 10.1145/564379.564383.

[44] N. D. KRISTOF, *The cheerleader: Earning a's in people skills at andover*, 2000. [Online]. Available: https://archive.nytimes.com/www.nytimes.com/library/politics/camp/061000wh-bush.html.

[45] O. Kuchaiev, B. Ginsburg, I. Gitman, *et al.*, "Mixed-precision training for nlp and speech recognition with openseq2seq," *arXiv preprint arXiv:1805.10387*, 2018.

[46] R. Kulkarni, *Big data goes big*, 2019. [Online]. Available: https://www.forbes.com/sites/rkulkarni/2019/02/07/big-data-goes-big/?sh=20394e8920d7.

[47] S. Kulkarni, A. Singh, G. Ramakrishnan, and S. Chakrabarti, "Collective annotation of wikipedia entities in web text," 2009. DOI: 10.1145/1557019.1557073.

[48] V. Levenshtein, "Binary codes capable of correcting deletions, insertions, and reversals," *Soviet Physics Doklady*, vol. 10, 8 1966.

[49] B. Z. Li, S. Min, S. Iyer, Y. Mehdad, and W. T. Yih, "Efficient one-pass end-to-end entity linking for questions," *EMNLP 2020 - 2020 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, 2020. DOI: 10.18653/v1/2020.emnlp-main.522.

[50] X. Li, X. Sun, Y. Meng, J. Liang, F. Wu, and J. Li, "Dice loss for data-imbalanced NLP tasks," 2020. DOI: 10.18653/v1/2020.acl-main.45. [Online]. Available: https://aclanthology.org/2020.acl-main.45.

[51] Y. Li, J. Li, Y. Suhara, J. Wang, W. Hirota, and W. C. Tan, "Deep entity matching: Challenges and opportunities," *Journal of Data and Information Quality*, 2021. DOI: 10.1145/3431816.

[52] E.-P. Lim, J. Srivastava, S. Prabhakar, and J. Richardson, "Entity identification in database integration," *Information Sciences*, vol. 89, no. 1, pp. 1–38, 1996, ISSN: 0020-0255. DOI: https://doi.org/10.1016/0020-0255(95)00185-9. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0020025595001859.

[53] Z. C. Lipton, C. Elkan, and B. Naryanaswamy, "Optimal thresholding of classifiers to maximize f1 measure," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 8725 LNAI, PART 2 2014, ISSN: 16113349. DOI: 10.1007/978-3-662-44851-9_15.

[54] Y. Liu, M. Ott, N. Goyal, *et al.*, "Roberta: A robustly optimized bert pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019. DOI: 10.48550/arXiv.1907.11692. [Online]. Available: https://doi.org/10.48550/arXiv.1907.11692.

[55] L. Logeswaran, M. W. Chang, K. Lee, K. Toutanova, J. Devlin, and H. Lee, "Zero-shot entity linking by reading entity descriptions," *ACL 2019 - 57th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference*, 2020. DOI: 10.18653/v1/p19-1335.

[56] T. McCoy, E. Pavlick, and T. Linzen, "Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference," 2019. DOI: 10.18653/v1/P19-1334. [Online]. Available: https://aclanthology.org/P19-1334.

[57] T. Mikolov, K. Chen, G. Corrado, and J. rey Dean, "Distributed representations of words and phrases and their compositionality. nips (2013), 1–9. doi: H p," *Dx. Doi. Org/10.1162/Jmlr*, 2013.

[58] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *1st International Conference on Learning Representations, ICLR 2013 - Workshop Track Proceedings*, 2013.

[59] J. Min, R. T. McCoy, D. Das, E. Pitler, and T. Linzen, "Syntactic data augmentation increases robustness to inference heuristics," *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020. DOI: 10.18653/v1/2020.acl-main.212. [Online]. Available: https://aclanthology.org/2020.acl-main.212.

[60] S. Mudgal, H. Li, T. Rekatsinas, *et al.*, "Deep learning for entity matching: A design space exploration," 2018. DOI: 10.1145/3183713.3196926. [Online]. Available: https://doi.org/10.1145/3183713.3196926.

[61] H. B. Newcombe, J. M. Kennedy, S. J. Axford, and A. P. James, "Automatic linkage of vital records," *Science*, vol. 130, 3381 1959, ISSN: 00368075. DOI: 10.1126/science.130.3381.954.

[62] T. H. Nguyen, A. Sil, G. Dinu, and R. Florian, "Toward mention detection robustness with recurrent neural networks," *arXiv preprint arXiv:1602.07749*, 2016.

[63] T. Niven and H. Y. Kao, "Probing neural network comprehension of natural language arguments," 2020. DOI: 10.18653/v1/p19-1459. [Online]. Available: https://aclanthology.org/P19-1459.

[64] M. Paganelli, F. D. Buono, A. Baraldi, and F. Guerra, "Analyzing how bert performs entity matching," *Proc. VLDB Endow.*, vol. 15, no. 8, pp. 1726–1738, 2022, ISSN: 2150-8097. DOI: 10.14778/3529337.3529356. [Online]. Available: https://doi.org/10.14778/3529337.3529356.

[65] G. Papadakis, D. Skoutas, E. Thanos, and T. Palpanas, "Blocking and filtering techniques for entity resolution: A survey," *ACM Computing Surveys*, vol. 53, 2 2020, ISSN: 15577341. DOI: 10.1145/3377455.

[66] R. Peeters and C. Bizer, "Dual-objective fine-tuning of bert for entity matching," *Proc. VLDB Endow.*, 2021, ISSN: 2150-8097. DOI: 10.14778/3467861.3467878. [Online]. Available: https://doi.org/10.14778/3467861.3467878.

[67] J. Peng and I. Ounis, "Combination of document priors in web information retrieval," vol. 4425 LNCS, 2007. DOI: 10.1007/978-3-540-71496-5_80.

[68] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," 2014. DOI: 10.3115/v1/d14-1162.

[69] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," *OpenAI blog*, 2018. [Online]. Available: https://www.gwern.net/docs/www/s3-us-west-2.amazonaws.com/d73fdc5ffa8627bce44dcda2fc012da638ffb158.pdf.

[70] I. S. A. Radford, J. Wu, D. L. Rewon Child, and D. Amodei, "Language models are unsupervised multitask learners," *OpenAI Blog*, 2019. [Online]. Available: https://d4mucfpksywv.cloudfront.net/better-language-models/language-models.pdf.

[71] M. P. K. Ravi, K. Singh, I. O. Mulang, S. Shekarpour, J. Hoffart, and J. Lehmann, "Cholan: A modular approach for neural entity linking on wikipedia and wikidata," *EACL 2021 - 16th Conference of the European Chapter of the Association for Computational Linguistics, Proceedings of the Conference*, 2021. DOI: 10.18653/v1/2021.eacl-main.40.

[72] K. Riaz, "Rule-based named entity recognition in urdu," 2010.

[73] C. S. Punla, https://orcid.org/ 0000-0002-1094-0018, cspunla@bpsu.edu.ph, R. C. Farro, https://orcid.org/0000-0002-3571-2716, rcfarro@bpsu.edu.ph, and Bataan Peninsula State University Dinalupihan, Bataan, Philippines, "Are we there yet?: An analysis of the competencies of BEED graduates of BPSU-DC," *International Multidisciplinary Research Journal*, vol. 4, no. 3, pp. 50–59, 2022.

[74] H. Sak, A. Senior, and F. Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling," 2014. DOI: 10.21437/interspeech.2014-80.

[75] E. F. T. K. Sang and F. de Meulder, "Introduction to the conll-2003 shared task: Language-independent named entity recognition," *Proceedings of the 7th Conference on Natural Language Learning, CoNLL 2003 at HLT-NAACL 2003*, 2003.

[76] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "Distilbert, a distilled version of bert: Smaller, faster, cheaper and lighter," *ArXiv*, vol. abs/1910.01108, 2019.

[77] R. Singh, V. Meduri, A. Elmagarmid, *et al.*, "Generating concise entity matching rules," 2017. DOI: 10.1145/3035918.3058739. [Online]. Available: https://doi.org/10.1145/3035918.3058739.

[78] M. Srinivasan and D. Rafiei, "Location-aware named entity disambiguation," 2021. DOI: 10.1145/3459637.3482135.

[79] C. Su, H. Huang, S. Shi, P. Jian, and X. Shi, "Neural machine translation with gumbel tree-lstm based encoder," *Journal of Visual Communication and Image Representation*, vol. 71, 2020, ISSN: 10959076. DOI: 10.1016/j.jvcir.2020.102811.

[80] C. Sun, X. Qiu, Y. Xu, and X. Huang, "How to fine-tune bert for text classification?," vol. 11856 LNAI, 2019. DOI: 10.1007/978-3-030-32381-3_16.

[81] Z. Sun, M. Chen, and W. Hu, "Knowing the no-match: Entity alignment with dangling cases," *ACL-IJCNLP 2021 - 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, Proceedings of the Conference*, 2021. DOI: 10.18653/v1/2021.acl-long.278. [Online]. Available: https://aclanthology.org/2021.acl-long.278.

[82] E. Sutinen and J. Tarhio, "On using q-gram locations in approximate string matching," vol. 979, 1995. DOI: 10.1007/3-540-60313-1_153.

[83] K. M. Tarwani and S. Edem, "Survey on recurrent neural network in natural language processing," *Int. J. Eng. Trends Technol*, vol. 48, no. 6, pp. 301–304, 2017.

[84] S. Tejada, C. A. Knoblock, and S. Minton, "Learning object identification rules for information integration," *Information Systems*, vol. 26, no. 8, pp. 607–633, 2001, Data Extraction,Cleaning and Reconciliation, ISSN: 0306-4379. DOI: https://doi.org/10.1016/S0306-4379(01)00042-4. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0306437901000424.

[85] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, "Attention is all you need," 2017, ISBN: 9781510860964. [Online]. Available: http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf.

[86] V. S. Verykios, G. V. Moustakides, and M. G. Elfeky, "A bayesian decision model for cost optimal record matching," *VLDB Journal*, vol. 12, 1 2003, ISSN: 10668888. DOI: 10.1007/s00778-002-0072-y.

[87] H. Wang, J. Li, H. Wu, E. Hovy, and Y. Sun, "Pre-trained language models and their applications," *Engineering*, 2022, ISSN: 2095-8099. DOI: https://doi.org/10.1016/j.eng.2022.04.024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2095809922006324.

[88] J. Wang, Y. Li, and W. Hirota, "Machamp: A generalized entity matching benchmark," in *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, ser. CIKM '21, Virtual Event, Queensland, Australia: Association for Computing Machinery, 2021, pp. 4633–4642, ISBN: 9781450384469. DOI: 10.1145/3459637.3482008. [Online]. Available: https://doi.org/10.1145/3459637.3482008.

[89] T. Wolf, L. Debut, V. Sanh, *et al.*, "Transformers: State-of-the-art natural language processing," 2020. DOI: 10.18653/v1/2020.emnlp-demos.6. [Online]. Available: https://aclanthology.org/2020.emnlp-demos.6.

[90] C. Wu, F. Wu, Y. Yu, T. Qi, Y. Huang, and Q. Liu, "Newsbert: Distilling pre-trained language model for intelligent news application," *Findings of the Association for Computational Linguistics, Findings of ACL: EMNLP 2021*, 2021. DOI: 10.18653/v1/2021.findings-emnlp.280.

[91] Y. Wu, X. Liu, Y. Feng, Z. Wang, and D. Zhao, "Neighborhood matching network for entity alignment," *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020. DOI: 10.18653/v1/2020.acl-main.578. [Online]. Available: https://aclanthology.org/2020.acl-main.578.

[92] C. Xia, C. Zhang, H. Nguyen, J. Zhang, and P. Yu, "Cg-bert: Conditional text generation with bert for generalized few-shot intent detection," *arXiv preprint arXiv:2004.01881*, 2020.

[93] C. Xiao, X. Hu, Z. Liu, C. Tu, and M. Sun, "Lawformer: A pre-trained language model for chinese legal long documents," *AI Open*, vol. 2, 2021, ISSN: 26666510. DOI: 10.1016/j.aiopen.2021.06.003.

[94] X. Yang, X. Gu, S. Lin, *et al.*, "Learning dynamic context augmentation for global entity linking," *EMNLP-IJCNLP 2019 - 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Proceedings of the Conference*, 2019. DOI: 10.18653/v1/d19-1026.

[95] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, and Q. V. Le, "Xlnet: Generalized autoregressive pretraining for language understanding," 2019.

[96] L. Yao, H. Liu, Y. Liu, X. Li, and M. W. Anwar, "Biomedical named entity recognition based on deep neutral network," *International Journal of Hybrid Information Technology*, vol. 8, 8 2015, ISSN: 17389968. DOI: 10.14257/ijhit.2015.8.8.29.

[97] F. Zhai, S. Potdar, B. Xiang, and B. Zhou, "Neural models for sequence chunking," 2017. DOI: 10.1609/aaai.v31i1.10995.

[98] C. Zhao and Y. He, "Auto-em: End-to-end fuzzy entity-matching using pre-trained deep models and transfer learning," 2019. DOI: 10.1145/3308558.3313578. [Online]. Available: https://doi.org/10.1145/3308558.3313578.

[99] H. Zhou, W. Huang, M. Li, and Y. Lai, "Relation-aware entity matching using sentence-bert," *Computers, Materials and Continua*, vol. 71, 1 2022, ISSN: 15462226. DOI: 10.32604/cmc.2022.020695.

[100] Z. Zhou, H. Huang, and B. Fang, "Application of weighted cross-entropy loss function in intrusion detection," *Journal of Computer and Communications*, vol. 09, 11 2021, ISSN: 2327-5219. DOI: 10.4236/jcc.2021.911001.