Deep Learning Based Approach to Unstructured Record Linkage

Jurek-Loughrey, A. (2021). Deep Learning Based Approach to Unstructured Record Linkage. *International Journal of Web Information Systems*. https://doi.org/10.1108/IJWIS-05-2021-0058

Published in:
International Journal of Web Information Systems

Document Version:
Peer reviewed version

Deep Learning Based Approach to Unstructured Record Linkage

No Author Given

No Institute Given

Abstract

Purpose

In the world of big data, data integration technology is crucial for maximising the capability of data-driven decision making. Integrating data from multiple sources drastically expands the power of information and allows us to address questions that are impossible to answer using a single data source. Record Linkage (RL) is a task of identifying and linking records from multiple sources that describe the same real world object (e.g. person), and it plays a crucial role in the data integration process. RL is challenging as it is uncommon for different data sources to share a unique identifier. Hence the records must be matched based on the comparison of their corresponding values. Most of the existing RL techniques assume that records across different data sources are structured and represented by the same scheme (i.e. set of attributes). Given the increasing amount of heterogeneous data sources, those assumptions are rather unrealistic. The purpose of this paper is to propose a novel RL model for unstructured data.

Methodology

In our previous work [16] we proposed a novel approach to linking unstructured data based on the application of the Siamese Multilayer Perceptron model. It was demonstrated that our method performed on par with other approaches that make constraining assumptions regarding the data. This paper expands our previous work originally presented at iiWAS2020 [16] by exploring new architectures of the Siamese Neural Network, which improves the generalisation of the RL model and makes it less sensitive to parameter selection.

Findings

The experimental results confirm that the new Autoencoder based architecture of the Siamese Neural Network obtains better results in comparison to the Siamese Multilayer Perceptron model proposed in [16]. Better results have been achieved in three out of four datasets. Furthermore, it has been demonstrated that the second proposed (hybrid) architecture based on integrating the Siamese Autoencoder with a Multilayer Perceptron model, makes the model more stable in terms of the parameter selection.

Originality

To address the problem of unstructured RL, this paper presents a new deep learning based approach to improve the generalisation of the Siamese Multilayer Preceptron model and makes it less sensitive to parameter selection.

Keywords: Record Linkage, Unstructured Data, Siamese Neural Network

1 Introduction

Society worldwide is generating more and more data giving rise to the "data deluge" problem. Making sense of such data is necessary for making strategically important decisions by government bodies, security, healthcare, financial entities, to name a few. Often to enable decision-making, data from different sources have to be integrated [7]. For instance, integrating records from law enforcement watch lists with data coming from financial institutions, car rental companies, airlines, immigration agencies, and residency agencies could help to prevent terrorist attacks. As a part of the data integration process, records (from two or more data sources) that refer to the same real world entity (e.g. person) need to be linked. In many cases, datasets do not share a unique identifier (e.g. National Insurance Number), thus the process of linking records needs to be performed by matching their corresponding attributes. This becomes challenging due to issues such as different data formats, language ambiguity and abbreviations. Commonly applied RL methods require assistance from a domain expert to carefully hand-craft bespoke domain-specific linking rules that aid in determining the linkage likelihood of a candidate record pair [9]. This requires deep topical expertise in the domain and continuous maintenance to cope with any changes in the character of the data, which is a costly proposition in many realistic scenarios. Given its pivotal importance and challenges, there has been strong interest in RL in the last decade within the computer science domain [7] [11]. In particular, the application of Machine Learning (ML) offers a promising approach, which can be applied as an alternative to manual rule building [20]. However, the existing ML-based approaches to RL are based on the assumption that the data obtained from different sources is structured and represented by overlapping sets of attributes [25][32][9][30] [15] [18] [8] [28]. This is very restrictive in terms of real world applications, given the increasing number of unstructured data sources such as social media channels, for example. Consequently, using ML methods for RL tasks becomes more challenging and it has been limited to structured data only. In our previous work [16], we introduced a new approach to unstructured RL based on an application of the Siamese Neural Network and text embedding models. It was demonstrated that the proposed model performed on par with ML-based methods yet it does not make any assumptions with respect to the data. Following the experimental evaluation, we learnt that there was still room for improvement in terms of the generalisation of the model. The model also tended to be sensitive with respect to the parameter selection. In this work we tried to address the limitations of the model proposed in [16]. The main contributions of this work are as follows. We proposed a new architecture of the Siamese Neural Network, which improves the generalisation of he model in [16]. We further modified the architecture in order to obtain a model that is less sensitive to parameter selection.

1.1 Overview of the Record Linkage Task

An illustrative example of a RL task is presented in Table 1 containing records from two digital libraries, DBLP and ACM . In this case, RL can be applied to detect which of the record pairs represent the same publication, which in this case should be (ACM1, DB1) and (ACM2, DB2). Any other pair should be considered as non-match. Formally, for two sets of records S and T, RL is

Table 1: An Example of RL.

ID	Title	Authors	Venue
ACM1	A compact B-tree	Peter Bumbulis, Ivan T. Bowman	International Conference on Management of Data
ACM2	A theory of redo recovery	David Lomet, Mark Tuttle	International Conference on Management of Data
DB1	A compact B-tree	Ivan T. Bowman, Peter Bumbulis	SIGMOD Conference
DB2	A theory of redo recovery	Mark R. Tuttle, David B. Lomet	SIGMOD Conference
DB3	Enhanced Abstract Data Types in Object-Relational Databases	Praveen Seshadri	VLDB J.
DB4	Parametric Query Optimization	Raymond T. Ng, Timos K. Sellis, Yannis E. Ioannidis, Kyuseok Shim	VLDB J.

defined as a task of identifying pairs of records $(s, t) \in S \times T$ that correspond to the same real world entity. In the RL process, each pair of records from $S \times T$ is being classified as a match or non-match.

1.2 Machine Learning Based Approach to Record Linkage

The majority of existing work in the space of ML and RL has focused on structured data sources where records are represented by overlapping sets of attributes (e.g. Table 1). For two structured data sources with overlapping attributes, the similarity between any two records can be determined by comparing their corresponding attributes using a similarity measure. Formally, given a pair of records $\{s = (s.f_1, \ldots, s.f_N), t = (t.f_1, \ldots, t.f_N)\}$ (where $f_1, \ldots, f_N$ represent the overlapping attributes), a similarity measure m quantifies the similarity between two attributes values of s and t and can be formulated as:

$$m : F_i \times F_i \rightarrow [0, 1] \tag{1}$$

where F_i denotes the domain of attribute f_i. The similarity measure returns a numeric value ranging between 0 and 1, referred to as a similarity value. $m(s.f_i, t.f_i) = 1$ indicates that the pair of records have the exact same values on the attribute f_i, while $m(s.f_i, t.f_i) = 0$ indicates that there is no similarity between the values $s.f_i$ and $t.f_i$. Some of the commonly used similarity measures include Jaro [14], Jaro-Winkler [33], Jaccard [13], Q-Gram [29], and Levenshtein edit distance [22]. It has been demonstrated [5] that depending on the type of data, different similarity measures have different levels of accuracy. Moreover, there does not exist a single similarity measure that is optimal for all data sets [1]. Once the similarity values are determined for all candidate record pairs, each pair of records can be represented as a comparison vector of length N. Notationally, for two records $s = (s.f_1, \ldots, s.f_N)$ and $t = (t.f_1, \ldots, t.f_N)$ and a similarity measure m, a comparison vector for s and t is formulated as:

$$m(s,t) =< m(s.f_1, t.f_1), \ldots, m(s.f_N, t.f_N) > \tag{2}$$

Each element of the comparison vector represents a numeric similarity value calculated with a similarity measure on the corresponding pair of attributes of the records s and t. For a labelled dataset (i.e. each comparison vector is labelled as 1 - representing a matching record pair or 0 - representing non-matching records), the RL task can be then considered as a comparison vector classification problem [9]. Notationally:

$$[S, T, m] \Longrightarrow LR : \overrightarrow{V} \rightarrow 0, 1 \tag{3}$$

where $\overrightarrow{V} = \{m(s_i, t_j) : s_i \in S, t_j \in T\}$ is the set of comparison vectors generated for each record pair from $S \times T$. Using supervised learning for training a comparison vector classification model has been proved to be very effective in the past. However, ML methods can only be applied with structured data where records from different sources are represented by a common set of attributes.

1.3 Challenges

The key challenge of ML based approaches to RL is the fact that they can not handle unstructured and heterogeneous data sources. The majority of existing techniques assume that the data obtained from different sources is structured and represented by an overlapping set of attributes. This is a major limitation as nowadays the data is being obtained in many various formats including structured i.e. predefined data model (e.g. relational database), semi-structured (e.g. Extensive Mark-up Language, LATEX, web data, scientific data), and unstructured i.e. no predefined data model, usually text (e.g. text documents, email messages, research publications) records. Consequently, integration and linkage of unstructured and heterogeneous data types still remains a challenge.

2 Unstructured Record Linkage

2.1 Problem Definition

Consider two datasets of records $S = \{s_1, \ldots, s_n\}$ and $T = \{t_1, \ldots, t_m\}$, where each dataset may contain structured or unstructured records. We further assume that the records across S and T are not represented by the same scheme, i.e. are not represented by an overlapping set of attributes. For each record from S and T, we use L to denote a pre-trained language model (e.g. Word2Vec [23], GloVe [27] or BERT[6]), which can be used to embed the records into numerical vectors of the same dimension. For unstructured data such as emails, a pre-trained language model is applied to vectorise each of the records. For structured data (i.e. with pre-defined attributes), first an embedding vector can be obtained for each attribute value individually. The embedding of a record can then be calculated as a combined vector of all attribute embeddings (e.g. by averaging or concatenation). Here, we address the task of supervised RL, that of leveraging $\{S, T\}$ and a language model L to train a classification model for predicting a pair of records $\{s_i, t_j\}$ as a match or non-match. Notationally:

$$[S, T, L] \xrightarrow[\text{Learning}]{\text{Supervised}} RL : S \times T \rightarrow \{0, 1\} \tag{4}$$

The output of RL would be a 0/1 label, where 0 indicates that a pair of records does not refer to the same entity while 1 means that the two records are a match.

2.2 Existing Solution

In our previous work [16], we proposed a new approach to RL which can be applied to unstructured data. The proposed model was a Siamese Multilayer Perceptron whose architecture is presented in Figure 1. The model's architecture is composed of two twin Multilayer Perceptron (MLP) networks which share the same structure (i.e. number of layers and neurons) and parameters. The model takes as an input a pair of records represented by their embedded vectors obtained from a selected language model (x_1 and x_2). Each of the two networks produces a high level feature representation of the input vectors (o_1 and o_2). The training dataset is composed of pairs of records (i.e. vector embeddings) and the labels indicating whether a pair is a match or non-match. The training objective of the model is to learn a new representation of the records that minimise the Euclidean distance between matching records and maximise it for non-matching records. In the training process the contrastive loss function is used. This is a distance based function, as opposed to prediction based functions (e.g. Cross Entropy) commonly used in classification tasks. The contrastive loss tries to ensure that semantically similar records are embedded close together. Given a pair of embedding vectors (x_1, x_2), it is calculated as:

$$L = y \times \|\rho(x_1) - \rho(x_2)\|^2 + (1 - y) \times max(margin - \|\rho(x_1) - \rho(x_2)\|, 0)^2 \tag{5}$$

where y is the ground truth relation between the original records and ρ represents the function of the MLP model. In this case $y = 1$ if x_1 and x_2 represent embeddings of two matching records and $y = 0$ otherwise. The $margin$ parameter is used to tighten the constraint in the learning process. If two records do not represent the same entity (i.e. are not matching) then the distance between them should be at least the $margin$. In the classification process, a new pair of embedded vectors is passed as the input to the model and their new representations are provided as the output. Following this, the Euclidean distance between the two output vectors is calculated. Based on their distance and a pre-defined threshold, the pair of records is classified as a match or as a non-match. The empirical evaluation of the proposed Siamese Multilayer Perceptron (SMLP) demonstrated that the model performed on par with other approaches, which make constraining assumptions regarding the data.

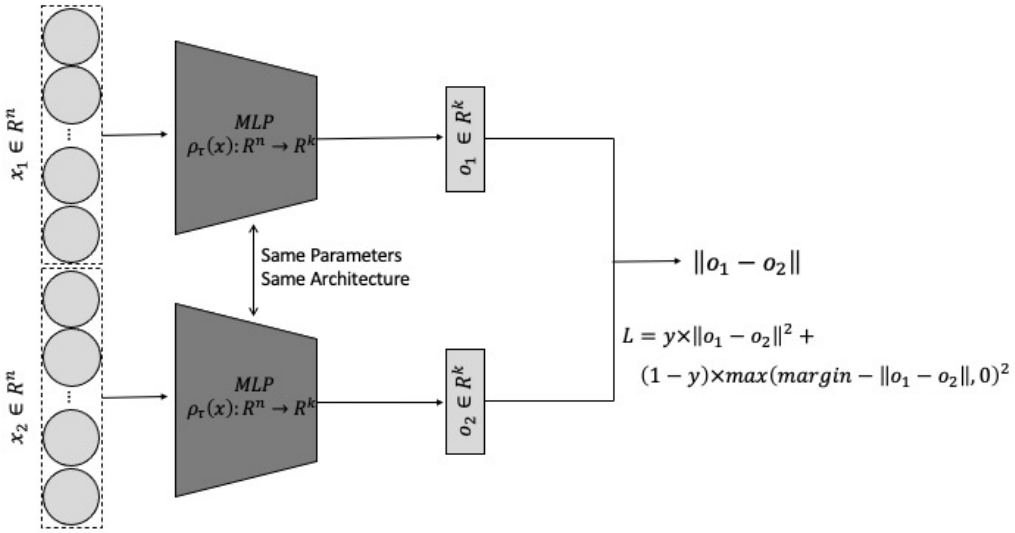

Fig. 1: Siamese Multilayer Perceptron with contrastive loss function applied for the RL task.

2.3 Proposed Approach

In this work we explore different architectures of Siamese Neural Networks (SNN) trying to improve the performance of the approached proposed in [16]. In particular, we explored two different directions. First, we proposed to use Autoencoder instead of the MLP architecture for the creation of the SNN. Autoencoder is an

Artificial Neural Network (ANN), which allows us to learn efficient data encodings in an unsupervised manner. The purpose of using Autoencoder is to learn a new representation of records by training the network to ignore any noise in the data. We hypothesise that combining the Autoencoder reconstruction loss with the contrastive loss function will lead to a better generalisation of the model and hence improve its performance on new (validation) data. As the second direction we proposed to use a hybrid approach based on combination of a Siamese Autoencoder and MLP in order to make the model less sensitive to the selection of its two parameters (i.e. $margin$ and $classification\ threshold$). Following the training of the Siamese Autoencoder we used the new data representation to further train a supervised MLP with binary cross entropy loss function. The two aforementioned SNN architectures are explained in detail in the following sections.

2.4 Siamese Autoencoder

The architecture of the proposed model is presented in Figure 2. The model is built with two identical Autoencoders (i.e. sharing the same architecture and parameters) and it requires a training dataset containing record pairs labelled as match/non-match. The input is composed of two vectors of the same dimension representing embeddings of a pair of records. Similarly as with the model proposed in [16], the goal of the training process is to learn new representations of the vectors, which minimise the distance between matching record pairs and maximise it for non-matching records. For this purpose, the contrastive loss (L_3) is being calculated on the bottlenecks (encoded inputs) of the two Autoencoders. In addition to this, the reconstruction losses of the two Autoencoders (L_1 and L_2) are being calculated and included in the final loss function of the Siamese model. Given a pair of embedding vectors (x_1, x_2), the final hybrid loss function of the proposed Siamese Autoencoder model is calculated as per Equation 6.

$$L_{hybrid} = \alpha \times (\|x_1 - \varphi(\rho(x_1))\|^2 + \|x_2 - \varphi(\rho(x_2))\|^2) + \\ y \times \|\rho(x_1) - \rho(x_2)\|^2 + (1-y) \times max(margin - \|\rho(x_1) - \rho(x_2)\|, 0)^2 \quad (6)$$

The first component of the loss function from Equation 6 represents the reconstruction losses of the twin Autoencoder models, where ρ and φ represent the functions of the Encoder and the Decoder respectively. The second component refers to the contrastive loss calculated on the embedded representations of x_1 and x_2 calculated by the twin Autoencoders. α is a parameter indicating the weight of the reconstruction loss.

The motivation behind using the hybrid loss function is to downgrade the impact of the contrastive loss function on the training process by combining it with the reconstruction losses of the Autoencoder models. We hypothesise that training the model for the additional task will improve the generalisation of the RL model. As it was demonstrated in [21], adding the unsupervised task of reconstructing the input can improve the generalisation performance of a supervised task.

Following the training of the model with a set the record pairs labelled as match/non-match, only the Encoder model is used in the classification process. For a new record pair, their embeddings are first obtained from the language model. Following this, each embedded vector is passed through the Encoder model providing two vectors as output. In the final step, the Euclidean distance is calculated between the output vectors and depending on a pre-defined threshold the pair of records is classified as a match or non-match. Both parameters of the model (i.e. *margin* and *classification threshold*) need to be optimised with the application of a validation dataset.

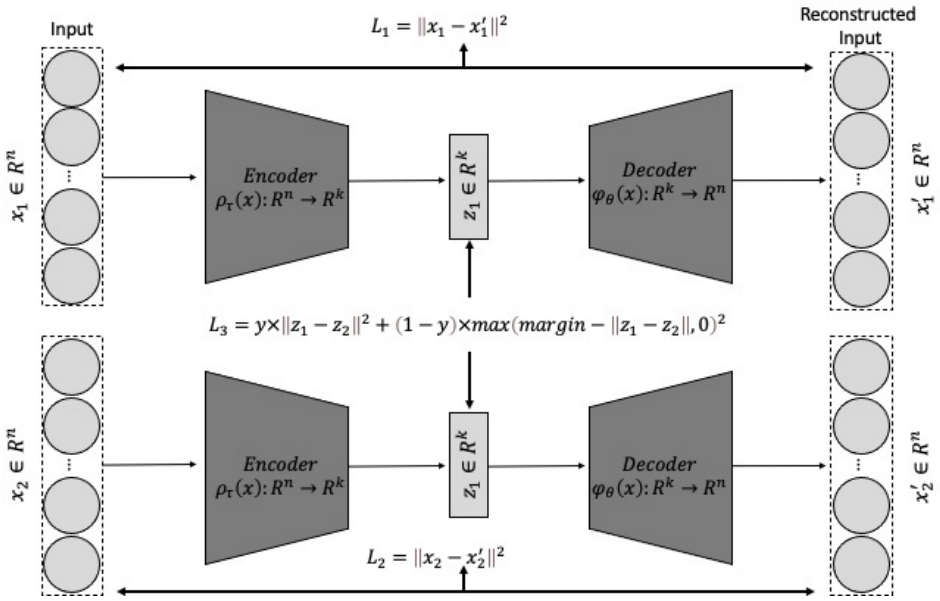

Fig. 2: Siamese Autoencoder applied for the RL task.

2.5 Siamese Autoencoder with Multilayer Perceptron

It has been demonstrated in our experimental evaluation (presented later in the paper) that the Siamese Autoencoder model is sensitive with the respect to the selection of its two parameters, namely *margin* and *classification threshold*. Trying to address this issue we proposed a modified version of the model which is trained in two phases. In the first phase the Siamese Autoencoder introduced in the previous section is trained with a set of record pairs labelled as match/non-match and the hybrid loss function. In the second phase, the new representations of the records (trained by the Siamese Autoencoder) is further fine-tuned with a MLP model for a supervised classification task (i.e. classifying a pair of records

as a match or a non-match). The architecture of the model is presented in Figure 3. A pair of vectors (representing a record pair) is first passed through the previously trained Encoder model providing two embedded vectors as an output. The two output vectors are combined into a single vector by computing the absolute values of the differences between their corresponding entrances. This representation of record pairs with their labels is further applied to train a MLP model for a supervised classification task (i.e. classifying a pair of records as a match or a non-match) with binary cross-entropy loss function. Combining representations of two records into a single vector and training a MLP model to classify such a record pair representation allows us to eliminate the *classification threshold* parameter, which is necessary while classifying a record pair based on their distance. Furthermore, by performing the second training phase we can reduce the impact of the *margin* selection in the training of the Siamese Autoencoder on the final performance of the model.

Following the training of the model, a new pair of vectors (i.e. records) is first passed through the Encoder model in order to obtain their embedded representation. The two outputs are then combined and passed as an input to the MLP model. Based on the output of the MLP, the record pair is classified and a match or a non-match.

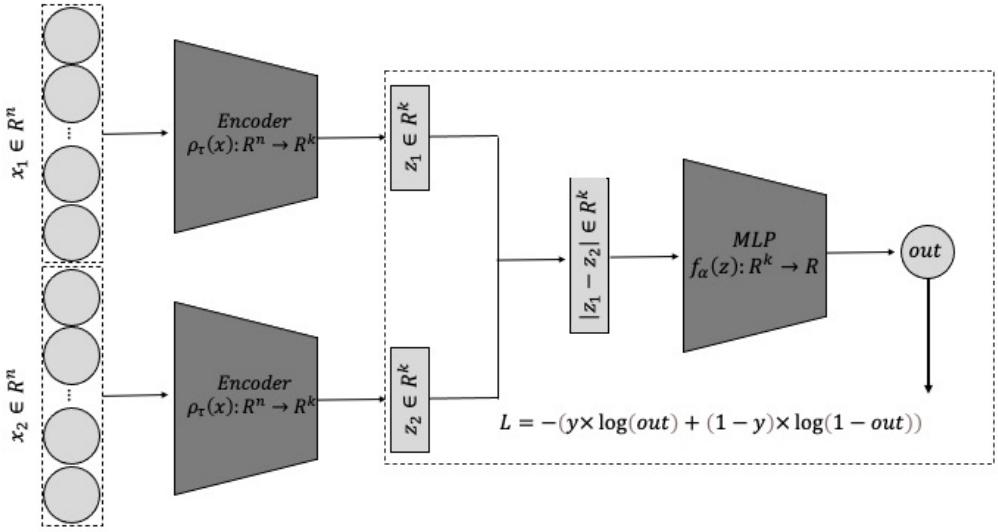

Fig. 3: Siamese Autoencoder with Multilayer Perceptron applied for the RL task.

3 Experimental Evaluation

In this section we present the experimental evaluation of the two RL models architectures presented in the previous section. We conduct a set of experiments in order to answer the following key questions:

- Is the Autoencoder based Siamese Neural Network with hybrid loss function more suitable for the RL task than the MLP based architecture proposed in [16]?
- How does combining the Siamese Autoencoder model with the MLP model impact its sensitivity to the parameter selection and the final classification performance?
- How does the performance of the proposed Siamese model architectures compare against the results obtained by the baseline approaches?

3.1 Experimental Setup

Datasets. The experiments are conducted with four datasets commonly used by the RL community. The properties of each dataset are listed in Table 2. The Restaurant[1] dataset contains records of 864 restaurants, each with five fields (name, address, city, phone, type). The Cora[1] dataset is a collection of 1,295 citations to computer science papers represented by 4 fields (author, title, venue, year). The DBLP-ACM and DBLP-Scholar are bibliographic datasets of computer science bibliography records represented by four attributes (title, authors, venue, year).

For the evaluation purposes of the proposed approaches, all datasets have been unstructured so that each record is represented by unstructured text composed of all attributes values merged together.

Blocking. In order to simplify the linkage process, we first perform blocking to get rid of the obvious non-matching record pairs. This is a standard preprocessing step with any RL method. In this work we use a schema-agnostic blocking method proposed in [26], which is one of the state-of-the-art unsupervised blocking techniques applicable to unstructured datasets. The records are first divided into overlapping blocks according to their common tokens. Following this, record pairs are either retained or removed based on their number of shared common blocks. In the linkage process, only record pairs from the same block are compared.

[1] `https://www.cs.utexas.edu/users/ml/riddle/data.html`

[2] `http://dbs.uni-leipzig.de/en/research/projects/object_matching/fever/benchmark_datasets_for_entity_resolution`

Table 2: Properties table for evaluated datasets.

Name	Number of Attributes	Number of Records	Number of Records Pairs	Number of Matches	Task
Restaurant	5	864	372,816	112	Deduplication
Cora	4	1,295	837,865	17,184	Deduplication
DBLP-ACM	4	2,616+2,294	6,001,104	2,224	Linkage
DBLP-Scholar	4	2,616+64,263	168,181,505	5,347	Linkage

3.2 Siamese Autoencoder

In this section we evaluate the proposed Siamese Autoencoder RL (referred to as SA) model in comparison to the Siamese MLP (referred to as SMLP) proposed in [16]. The Encoder and Decoder of the SA model have the same architectures including one hidden layer. The dimension of the bottleneck is set to 50. The SMLP is composed of two twin MLP networks containing one hidden layer. For both architectures, we use Leaky Rectified Linear Units (leaky ReLu) as the activation function in the hidden layer. SMLP uses the contrastive loss function, while for SA we apply the hybrid loss combining the contrastive and reconstruction losses. The parameters of the contrastive loss function are optimised with the application of a validation dataset.

The loss function is combined with a standard back propagation algorithm where the gradient is added across the twin networks due to the tied weights. We use Adam optimiser with a learning rate of $\eta = 0.001$ and mini batch size of 256. For the weights we apply He initialisation with uniform distribution. Biases are initialised to be zero. We apply l_1 weights regularisation and dropout with the keep probability set to 0.5. We train each network for 100 epochs. For the generation of the record embeddings (i.e. vectors provided as the inputs to both of the models) we applied BERT [6], one of the state-of-the-art language models.

The results in a form of F-measure obtained by the two models with each of the four datasets are demonstrated in Table 3. For each dataset, the experiments were conducted with 10-cross validation. It can be observed that the Autoencoder based Siamese Neural Network outperformed the MLP based architecture on three out of four datasets. Even though the difference is rather small, SA performs consistently better than SMLP. This indicates that as we expected, adding the unsupervised task of input reconstruction to the supervised RL task improves the generalisation of the model.

Table 3: F-measure obtained by the SA and the SMLP models.

Dataset:	**Restaurant**	**Cora**	**DBLP-ACM**	**DBLP-Scholar**
SMLP	0.97	0.999	0.955	0.849
SA	0.975	0.999	0.965	0.857

3.3 Siamese Autoencoder with Multilayer Perceptron

In this section we evaluate the performance of the second proposed architecture that combines the SA with the supervised MLP model (AS+MLP). The AS+MLP model does not require the *classification threshold* parameter as it is trained to classify each pair of records based on the combination of their embedded representations rather than based on the distance between them (as in case of the SA). It still however relies on the *margin* parameters, which is used during training the SA model. In order to investigate how sensitive the SA and the SA+MLP are to the selection of their parameters values, we evaluate both of the models using a range of different parameters values. For the *margin* parameter we use values ranging from 2.5 to 4.5 with the increment of 0.1. For the *classification threshold* we applied values form 1 to 4 incrementing the values by 0.1. The summary of the results (F-measure) obtained by both models across different values of the parameters are presented in Table 4. For each model and each dataset we reported the maximum, minimum and average of F-measure achieved across all parameters combinations.

It can be observed from the results that the SA can obtain a higher maximum F-measure in comparison to the SA+MLP. The difference is particularly visible for the Restaurant and DBLP-Scholar datasets. However, at the same time the SA+MLP obtained better average performance in 3 out of 4 datasets. It also obtained a greater minimum value of F-measure for each dataset with a significant difference in 3 out of 4 datasets. The AS+MLP is clearly less sensitive to the selection of its parameter as the average difference between the maximum and minimum F-measure obtained across different datasets is 0.034. At the same time, the average difference in performance obtained by the SA across different parameter values is 0.243. This indicates that the SA is capable of providing better performance in RL tasks when validation data is available in order to optimise its parameters. The SA+MLP performs slightly worse however, it is much more stable and may be a good alternative when we don't have enough data to carefully select the parameters values.

Table 4: Comparison of the results (F-measure) obtained by the SA and the SA+MLP models for different *margin* and *classification threshold* parameters.

Dataset	SA			SA+MLP		
Dataset	**Min.**	**Ave.**	**Max.**	**Min.**	**Ave.**	**Max.**
Restaurant	0.764	0.895	0.975	0.814	0.875	0.942
Cora	0.991	0.997	0.999	0.998	0.999	0.999
DBLP-ACM	0.559	0.88	0.965	0.945	0.956	0.963
DBLP-Scholar	0.510	0.751	0.857	0.762	0.788	0.812

3.4 Comparison with Baseline RL Methods

In our experiments we compare the proposed SNN models with three different approaches to RL.

Machine Learning based approach. We apply four machine learning algorithms for training classification models. Those include Support Vector Machine (SVM) with linear kernel (SVM-L), SVM with polynomial kernel (SVM-P), SVM with RBF kernel (SVM-R) and Random Forest (RF) (number of trees = 500, maximum depth = 16). We use five different similarity measures for generating the comparison vectors (Jaro, Smith-Waterman, Q-Gram, Jaro-Winkler and Levenshtein edit distance).

As the ML based RL models require the data to be structured and represented by the same set of attributes across all data sources, the structure of each of the datasets used in the experiments was kept when applied with any of the aforementioned ML methods.

Distributed representation of records (DeepER). This is a recently proposed approach which applies a distributed representation of words [8] for constructing a distributed representation of records. For each token (word) within an attribute value its distributed representation is obtained from one of the pretrained embedding dictionaries. A distributed representation of the attribute is then constructed by averaging embeddings of all its tokens. Following this, for each pair of records, their comparison vector is generated by computing cosine similarity between embeddings of their corresponding attributes. Finally, the aforementioned four classification models are trained with the comparison vectors labelled as a match or non-match. For a new pair of records, their comparison vector is first calculated following the same procedure, which is then passed to the ML model and classified as a match or no-match.

This model also assumes that the data is structured and the records are represented by the same attributes.

TF-IDF rule based approach. This is a RL method [10] which uses Log TF-IDF (Term Frequency Inverse Document Frequency) values for measuring the similarity between records. The Log TF-IDF measure [17] is formally defined as:

$$\mathrm{sim}(t_i, t_j) = \sum_{q \in t_i \cap t_j} w(t_i, q) \cdot w(t_j, q), \tag{7}$$

where

$$w(t, q) = \frac{w'(t, q)}{\sqrt{\sum_{q \in t} w'(t, q)^2}}, \tag{8}$$

and

$$w'(t, q) = \log(tf(t, q) + 1) \cdot \log(\frac{|R|}{df(q, R)} + 1) \tag{9}$$

where (t_1, t_2) represents a record pair, $w(t, q)$ is the normalised TF-IDF weight

of a term q in a record t, $tf(t,q)$ represents the term frequency of q in t, $|R|$ is the total number of records in the dataset R, $df(q)$ is the document frequency of the term q in the cohort. For each record pair from R their similarity is calculated according to Equation 7. If the obtained similarity is greater than a predefined threshold the records are classified as a match. The optimal value of the threshold is usually determined using labelled data. Note that this is our only competitor that can be applied with unstructured data and it does not require for the records to be represented by the same attributes.

The results obtained by the three baseline methods on each of the four datasets are demonstrated in Table 5. We compare the baseline methods with the two models proposed in this paper, the SA and SA+ the MLP. For the ML based models, the best results obtained across all learning methods and similarity metrics used for constructing the comparison vectors are reported. For the ML based method and the DeepER models, the data had to be kept in its structured format. TF-IDF method is the only competitor that is applicable to unstructured data.

When comparing the SA with the TF-IDF based method, we observe that our proposed model performed much better for each of the datasets (the SA+MLP was slightly outperformed by the TF-IDF in two datasets). In comparison to the DeepER, the SA performed better in three out of four cases. The only dataset where the DeepER obtained slightly higher F-measure than the SA is DBLP-ACM dataset. The DeepER outperformed the SA+MLP in two out of four datasets. Finally, the ML based method outperformed the SA in two out of four cases and the SA+MLP in three out of four cases. We can note that for the Cora dataset, both the SA and the SA+MLP significantly outperformed any of the competitors. The obtained results demonstrate that the proposed SA approach to RL can be successfully applied with unstructured data. It has been slightly outperformed by the other two methods in some cases, however, it has a great advantage over them since it does not make any assumptions regarding the schema of the data. It can be easily applied with structured, semi-structured or unstructured records. It also allows for the data sources to be heterogeneous, i.e. represented in different formats. At the same time, the DeepER and the ML based methods require for the data to be structured and represented by overlapping attributes. The SA+MLP performed slightly worse but it was still on par with the other approaches.

4 Relevant Work

Record Linkage. The two key research directions that have been studied in the space of RL include: (1) the determination of time-efficient algorithms for RL [4], and (2) the development of methods for effective discovery of links [10] [3]. The former focuses on improving the speed of the RL process through reduction of the number of record comparisons (referred to as blocking). The latter field focuses on developing techniques for effective link discovery. In this work we focus on the second research problem, which is identifying links between records.

Table 5: Comparison of the performance (F-measure) of the SA and the SA+MLP models with the state-of-the-art ML, distributed tuple representation and the TFIDF based approaches.

Dataset	**SA**	**SA+MLP**	**Best-ML**	**DeepER**	**TFIDF**
Restaurant	0.975	0.942	0.97	0.972	0.947
Cora	0.999	0.999	0.946	0.955	0.809
DBLP-ACM	0.965	0.963	0.978	0.973	0.923
DBLP-Scholar	0.857	0.812	0.889	0.791	0.833

Work on RL models can be categorised as based on (1) declarative rules and (2) Machine Learning methods. With the first family of approaches generic rules are applied using similarity measures and thresholds in order to identify those pairs of records that are similar enough to be considered as matches [12] [31] [1]. An obvious advantage of those techniques is the fact that they can provide interpretable solutions, which is not the case with a ML based approach. In some work it was proposed to learn the rules by appropriate selection of the similarity functions and the thresholds [12] [31]. Even though it was possible to automate some of the steps, those methods still heavily rely on the expertise and knowledge of the data. Alternatively, ML based methods can automatically train a model to classify a comparison vector generated for a pair of records as a match or non-match [9]. Popular ML based approaches include genetic programming [25], active learning [32], SVM [9], self-learning [30] [15] [18]. In some recent work a distributed representation of records using word embeddings has been proposed for the task of RL [8] [28]. In [8] two techniques of computing distributed representation of attributes values were explored. The first one based on averaging embeddings of all tokens within an attribute, and the second using a recurrent neural network to convert each tuple into a numeric vector. Following this, a comparison vector was calculated for each pair of records. The similarity values between the corresponding attributes values (embeddings) were obtained using cosine similarity. Comparison vectors were further applied with the ML methods to train a RL model. In [28] word embeddings were applied to determine similarity between online user-generated content.

Even though ML based methods have been proved to work very well in the RL tasks, the strong assumption they make regarding the structure of the data makes them less and less applicable to increasing volumes of heterogeneous data sources.

Siamese Neural Networks. Siamese ANN were first proposed to solve a signature verification task as an image matching problem [2]. A SNN is an ANN, which is composed of two or more identical sub-networks (i.e. the same architecture and weights) that take different input vectors but are joined by a loss function at the top. Each of the network computes a high level representation of

the input vectors. The aim of the learning process is to find a similarity between the input vectors by comparing their high level representations. The learning can be performed using triplet or contrastive loss functions. It has been demonstrated that SNN can learn useful data descriptors that can be further used to compare between the inputs of the respective subnetworks. Its inputs can be anything from numerical data (in this case the subnetworks are usually formed by fully-connected layers), image data (with Convolutional Neural Networks as subnetworks) or even sequential data such as sentences or time signals (with Recurrent Neural Networks as subnetworks). SNNs have been successfully applied for a one-shot imagine recognition problem [19]. Siamese Recurrent Neural Networks with word embeddings have also been applied for learning sentence similarity [24].

5 Conclusion

In this work we propose a new Autoencoder-based SNN architecture for linking unstructured data. As opposed to previously introduced MLP-based Siamese model, with this approach a hybrid loss function is applied, which incorporates contrastive and reconstruction losses. The motivation behind the proposed design is to improve the generalisation of the supervised model by training the model for additional unsupervised task. For computing the vector representations of records (i.e. records embeddings) we applied the BERT language model. The experimental results demonstrate that the new proposed architecture of the Siamese model performs better in comparison to the SNN based on MLP proposed in [16].

SNN with contrastive loss function relies on two parameters, *margin* used during the training and $classification\ threshold$ used while classifying a pair of records as a match or a non-match. By evaluating the model with different values of the two parameters we were able to demonstrate that the model is very unstable in terms of the parameters selection. In order to alleviate this problem we propose another network architecture by integrating the Siamese Autoencoder with a MLP model. The Siamese Autoencoder is trained with the hybrid loss function and the new embedded representation of the records is further used for training the MLP for the record pairs classification task. In this way we eliminate the need for the $classification\ threshold$ parameter and also make the model much more robust to the selection of the *margin* parameter. Even though the classification model performs slightly worse than the distance based model, it is a good alternative in case when we don't have a sufficient validation dataset for optimisation of the parameters.

The two proposed approaches were also compared to three different RL methods including ML based models, another distributed records representation based approach and a rule based TF-IDF model. The proposed model outperformed the TF-IDF based method when applied to unstructured data. It also performed on par with the other two models, which required the data to be structured.

As the next step of this work we want to explore the integration of the proposed model with an embedding based blocking technique in order to provide an end-to-end linkage solution. The distributed representation of records used with our approach allows for addressing the blocking problem in a different manner, which has not yet been explored by the RL community.

References

1. Bilenko, M., Mooney, R., Cohen, W., Ravikumar, P., Fienberg, S.: Adaptive name matching in information integration. IEEE Intelligent Systems 18(5), 16–23 (2003)
2. Bromley, J., Guyon, I., LeCun, Y., Säckinger, E., Shah, R.: Signature verification using a" siamese" time delay neural network. In: Advances in neural information processing systems. pp. 737–744 (1994)
3. Christen, P.: Data matching: concepts and techniques for record linkage, entity resolution, and duplicate detection. Springer Science & Business Media (2012)
4. Christen, P.: A survey of indexing techniques for scalable record linkage and deduplication. IEEE transactions on knowledge and data engineering 24(9), 1537–1555 (2012)
5. Cohen, W., Ravikumar, P., Fienberg, S.: A comparison of string metrics for matching names and records. In: Kdd workshop on data cleaning and object consolidation. vol. 3, pp. 73–78 (2003)
6. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
7. Dong, X.L., Srivastava, D.: Big data integration. In: 2013 IEEE 29th international conference on data engineering (ICDE). pp. 1245–1248. IEEE (2013)
8. Ebraheem, M., Thirumuruganathan, S., Joty, S., Ouzzani, M., Tang, N.: Distributed representations of tuples for entity resolution. Proceedings of the VLDB Endowment 11(11), 1454–1467 (2018)
9. Elfeky, M.G., Verykios, V.S., Elmagarmid, A.K.: Tailor: A record linkage toolbox. In: Proceedings 18th International Conference on Data Engineering. pp. 17–28. IEEE (2002)
10. Elmagarmid, A.K., Ipeirotis, P.G., Verykios, V.S.: Duplicate record detection: A survey. IEEE Transactions on knowledge and data engineering 19(1), 1–16 (2007)
11. Getoor, L., Machanavajjhala, A.: Entity resolution: theory, practice & open challenges. Proceedings of the VLDB Endowment 5(12), 2018–2019 (2012)
12. Isele, R., Bizer, C.: Learning expressive linkage rules using genetic programming. Proceedings of the VLDB Endowment 5(11), 1638–1649 (2012)
13. Jaccard, P.: Distribution de la flore alpine dans le bassin des dranses et dans quelques régions voisines. Bull. Soc. Vaud. Sci. Nat. 37, 241–272 (1901)
14. Jaro, M.A.: Advances in record-linkage methodology as applied to matching the 1985 census of tampa, florida. Journal of the American Statistical Association 84(406), 414–420 (1989)
15. Jurek, A., Hong, J., Chi, Y., Liu, W.: A novel ensemble learning approach to unsupervised record linkage. Information Systems 71, 40–54 (2017)
16. Jurek-Loughrey, A.: Deep learning based approach to unstructured record linkage. In: Proceedings of the 22nd International Conference on Information Integration and Web-based Applications & Services. pp. 417–425 (2020)

17. Kejriwal, M., Miranker, D.P.: An unsupervised algorithm for learning blocking schemes. In: 2013 IEEE 13th International Conference on Data Mining. pp. 340–349. IEEE (2013)
18. Kejriwal, M., Miranker, D.P.: Semi-supervised instance matching using boosted classifiers. In: European Semantic Web Conference. pp. 388–402. Springer (2015)
19. Koch, G., Zemel, R., Salakhutdinov, R.: Siamese neural networks for one-shot image recognition. In: ICML deep learning workshop. vol. 2 (2015)
20. Köpcke, H., Thor, A., Rahm, E.: Evaluation of entity resolution approaches on real-world match problems. Proceedings of the VLDB Endowment 3(1-2), 484–493 (2010)
21. Le, L., Patterson, A., White, M.: Supervised autoencoders: Improving generalization performance with unsupervised regularizers. Advances in neural information processing systems 31, 107–117 (2018)
22. Levenshtein, V.I.: Binary codes capable of correcting deletions, insertions, and reversals. In: Soviet physics doklady. vol. 10, pp. 707–710 (1966)
23. Mitchell, J., Lapata, M.: Composition in distributional models of semantics. Cognitive science 34(8), 1388–1429 (2010)
24. Mueller, J., Thyagarajan, A.: Siamese recurrent architectures for learning sentence similarity. In: Thirtieth AAAI Conference on Artificial Intelligence (2016)
25. Ngomo, A.C.N., Lyko, K.: Unsupervised learning of link specifications: deterministic vs. non-deterministic. In: Proceedings of the 8th International Conference on Ontology Matching-Volume 1111. pp. 25–36. CEUR-WS. org (2013)
26. O'Hare, K., Jurek-Loughrey, A., Pires, C.: High-value token-blocking: Efficient blocking method for record linkage. ACM Transactions on Knowledge Discovery from Data (2021)
27. Pennington, J., Socher, R., Manning, C.: Glove: Global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). pp. 1532–1543 (2014)
28. Schneider, A.T., Mukherjee, A., Dragut, E.C.: Leveraging social media signals for record linkage. In: Proceedings of the 2018 World Wide Web Conference on World Wide Web. pp. 1195–1204. International World Wide Web Conferences Steering Committee (2018)
29. Shannon, C.E.: A mathematical theory of communication. ACM SIGMOBILE Mobile Computing and Communications Review 5(1), 3–55 (2001)
30. Sherif, M.A., Ngomo, A.C.N., Lehmann, J.: W ombat–a generalization approach for automatic link discovery. In: European Semantic Web Conference. pp. 103–119. Springer (2017)
31. Wang, J., Li, G., Yu, J.X., Feng, J.: Entity matching: How similar is similar. Proceedings of the VLDB Endowment 4(10), 622–633 (2011)
32. Wang, Q., Vatsalan, D., Christen, P.: Efficient interactive training selection for large-scale entity resolution. In: Pacific-Asia Conference on Knowledge Discovery and Data Mining. pp. 562–573. Springer (2015)
33. Winkler, W.E.: String comparator metrics and enhanced decision rules in the fellegi-sunter model of record linkage. pp. 354–359 (1990)