An Adaptable Framework for Entity Matching Model Selection in Business Enterprises

Alex Boyko
Faculty of Science
Vrije Universiteit Amsterdam
The Netherlands
alex@boyko.nl

Siamak Farshidi
Multiscale Networked Systems
University of Amsterdam
The Netherlands
s.farshidi@uva.nl

Zhiming Zhao
Multiscale Networked Systems
University of Amsterdam
The Netherlands
z.zhao@uva.nl

***Abstract*—Entity matching is the process of identifying data in different data sources that refer to the same real-world entity. A significant number of entity matching approaches have been introduced in the literature, which complicates the selection process. In this study, we propose a framework to support researchers in finding the best fitting entity matching model (s) based on the characteristics of their datasets. The proposed framework can be extended by adding more models, features, and use cases. To evaluate the framework, we have conducted a case study in the context of a business enterprise to support them with finding the right entity matching model out of five preselected models by the case study experts. The case study participants confirmed the framework's usefulness in assisting them in finding the best-fitting entity matching models. Having the knowledge regarding entity matching models readily available supports decision-makers at business enterprises in making more efficient and effective decisions that meet their requirements and priorities. Furthermore, such reusable knowledge can be employed by other researchers to develop new concepts and solutions for future challenges.**

***Index Terms*—model selection, decision-making process, entity matching, performance analysis, selection rules.**

I. Introduction

A fundamental problem in data cleaning and integration is dealing with uncertain and imprecise references to real-world entities [4]. An entity is a general term that can represent a person, product, organization, application, object, concept, etc. [39]. Similar entities can be perceived as different due to multiple reasons, such as incorrect data entry, incomplete data entry, or multiple possible representations of the entity, also known as ambiguity [11]. Identifying and matching real-world entities across data sources or within a single data source is known as entity matching (EM) [53]. Recent developments of big data both in industry and academia have become a driving force behind the research on entity matching.

In scientific literature, entity matching has many names: Entity alignment [53], entity resolution [38], record linkage [19], data matching [32], entity reconciliation [12], instance matching [42] are all synonyms of entity matching. Typically, entity matching is used within database-related tasks, such as deduplication, data merges, data processing, data visualization and within various Information Technology (IT) applications [33].

Recently, the main problem of entity matching has shifted from the lack of approaches to the lack of characterization and comparisons of EM approaches [1]. To give an overview, there exist many different types of EM solutions, such as rule-based methods [2], pairwise classification [48], clustering approaches [41] and more [20]. The data type plays an important role in entity matching, which means that some approaches empirically perform better with the structured data [28], some with textual data [31] and some with dirty data [6]. A solution can target a specific subclass of EM approaches, such as matching cross-lingual data [45] or matching bibliographic data [46]. Many competing solutions make it more difficult for the researchers to select the best-fitting entity matching model(s). Accordingly, an extendable and adaptable framework is required to support decision-makers at business enterprises with the EM model selection problem.

In this study, we designed a framework based on design science research to capture implicit and tacit knowledge regarding EM models systematically. We conducted a case study in the context of the ING banking organization in the Netherlands to evaluate the usefulness and efficiency of the framework in supporting the case study participants.

This study is structured as follows. Section II introduces the mixed research method that we have employed in this study. Section III elaborates on the constituent component and the workflows of the proposed framework. Section IV explains the evaluation process, results, and analysis of the framework at the ING banking organization. Section V discusses how we have addressed the research question, describes lessons learned, and highlights the threat to the validity of the study and how we have tackled them. Section VI positions our work among other existing model selection approaches in the literature. Finally, Section VII concludes this study and expresses the future direction of the research.

II. Research approach

Research methods are classified based on their data collection techniques (interview, observation, literature, etc.), inference techniques (taxonomy, protocol analysis, statistics, etc.), research purpose (evaluation, exploration, description, etc.), units of analysis (individuals, groups, process, etc.), and so forth [34]. Multiple research methods can be combined to achieve a fuller picture and a more in-depth understanding of the studied phenomenon by connecting complementary

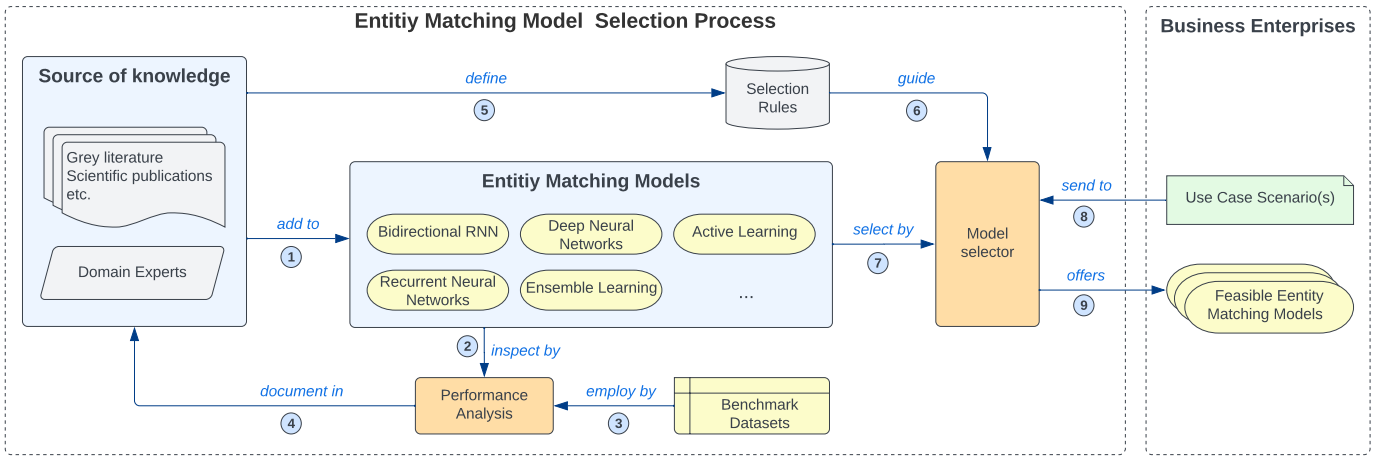

Fig. 1. The Framework for EM Model Selection

findings that conclude from the methods from the different methodological traditions of qualitative and quantitative investigation [24]. This section outlines the research questions and elaborates on a mixed method based on systematic literature review, design science research, case study research, and experiment to capture knowledge regarding EM models, answer the research questions, and build a framework for the EM model selection problem.

We utilize both manual and automatic literature search methods. The manual search is defined by exploring top-tier journals and conferences in knowledge representation, data engineering, database technologies, and more. The most prominent journals and conferences were chosen based on their scope and ranking, with the minimum ranking scores of A* or A on the CORE Rankings portal[1]. For the automatic search, a search query was defined and run through a number of portals that publish scientific literature, including the ACM digital library, IEEE Xplore, Springer, and more. We manually picked relevant publications that fall within the scope of this study and identified topics of the domain that will be discussed in this paper.

In total, manual and automatic searches yielded 1,394 papers related to entity matching. After the initial review of the titles and abstracts, applying inclusion and exclusion criteria, besides performing scanning and skimming, only 160 papers remained in the pool of studied literature. The main focus of the chosen selection criteria was to identify recent papers ($<$ 5 years) that facilitate the progress of entity matching. Furthermore, we obtained additional literature by using the snowballing approach. After conducting a quality assessment of the chosen papers, there were 67 papers to be added to the literature knowledge base of our framework.

We learned that EM solutions are often designed to tackle specific challenges [10]. Many introduced methods lack generalizability and fail to deliver good results in different settings. Despite many proposed solutions, data matching remains an open challenge for many in the current state of research. Hence, the goal of our research is to cover multiple challenges of entity matching and to answer the main research question: *MRQ: How to design an entity matching framework using a real-world case study?* To achieve our goal, we split it into three research questions, each covering a distinct scope of entity matching pipeline. (RQ_1) How should we compare a set of entity matching models based on real-world datasets against each other? (RQ_2) How should we combine entity matching approaches to achieve better performance? (RQ_3) How should we evaluate the selected entity matching models based on decision-makers' requirements?

We have conducted a *systematic literature review* based on the guidelines of Kitchenham [27] to explore state of the art and conduct a gap analysis to position our study among other efforts in the literature. *Design science research* is an iterative process [44], has its roots in engineering [22], is broadly considered a problem-solving process [16], and attempts to produce generalizable knowledge about design processes and design decisions. In this study, we designed and implemented[2] the framework for the EM model selection problem based on design science research. *Case study research* is an empirical research method [23], [14] that investigates a phenomenon within a particular context in the domain of interest [51]. Case studies can describe, explain, and evaluate a hypothesis. A case study can be employed to collect data regarding a particular phenomenon, apply a tool, and evaluate its efficiency and effectiveness using interviews. Note, we followed the guidelines outlined by Yin [50] to conduct and plan the case study in this study. In addition to the case study, we conducted an experiment by employing open public datasets to evaluate the efficiency and effectiveness of the framework in suggesting entity matching models.

[1]https://www.core.edu.au/home

[2]Please access the source code through the following link: https://github.com/AlexBoykoVU/DBLP_Scholar_entity_matching

To sum up, we employed design science research to address MRQ (Section III). Next, as a systematic literature review was conducted to address RQ_1 (Section III-A), afterward, we performed an experiment to investigate how different EM models can be combined (RQ_2) to offer higher performance in different scenarios and settings (Section III-A). Finally, to address RQ_3, we conducted a case study in the context of the ING banking organization in the Netherlands (Section IV).

III. The Framework for EM Model Selection

We designed and implemented a framework based on design science research in this study to support decision-makers at business enterprises in finding the best fitting EM models according to their use case scenarios. Figure I shows the constituent components and workflows of the framework. Let us elaborate on the framework: (Step 1) The knowledge acquisition phase of the framework starts with document analysis and literature study to collect data regarding potential EM models. Domain experts could evaluate the collected data, and their tacit knowledge might be used to gain more insights into the EM models. Next, the selected EM models should be added to the pool of *Entity Matching Models*. (Steps 2, 3, and 4) The performance of the selected EM models should be evaluated based on a set of open-source benchmark datasets. The results of the performance analysis can be considered as part of *Source of Knowledge*. (Step 5) Domain experts will use the collected data regarding the EM models and their performance to define a set of rules for selecting them in different scenarios and settings (see Section III-A3). (Steps 6, 7, 8, and 9) The model selector is a simple inference engine in the framework that gets the use case scenarios of business enterprises (for example, see Section IV), the selection rules, and potential EM models as its inputs and then offers the best fitting EM model(s) to the decision-makers (business enterprises).

A. The Development Process of the Framework

We have conducted an extensive literature study to gain insight regarding the entity matching models in literature and to extract an initial set of selection rules that could be used in the framework [1]. Next, we have conducted expert interviews with ten domain experts from different business enterprises in the Netherlands to select a set of models and selection rules for the framework. Finally, we have selected five entity matching models (including, Dedupe, SIF, RNN, Attention, and Hybrid). Additionally, three rules have been defined to select the models. Additionally, we used an open-source benchmarking dataset, called DBLP-Scholar, to conduct the performance analysis phase of the framework (See Figure I). Finally, to evaluate the framework's usefulness, a case study in the context of a Dutch multinational banking and financial services corporation called ING has been conducted.

1) Entity Matching Model: A variety of entity matching models are available in the literature. This section introduces five models that we have used in this study to meet the case study participants' requirements. These models were chosen based on their benchmarking performance and unique features. These models are Dedupe (active learning), SIF, RNN, Attention, and Hybrid.

Active Learning - The Dedupe model is a semi-supervised machine learning model that implements active learning to perform entity matching on structured data. It works by first creating an initial set of entity pairs, then prompting the decision-maker for feedback about the given pairs, and then learning to produce a final set of matched entities. The implementation of Dedupe is based on the paper written by Bilenko [5] "Learnable similarity functions and their applications to clustering and record linkage." Dedupe is available as a Python library and as a standalone application. Apart from entity matching across multiple datasets, Dedupe is often used to perform deduplication of entities within the same dataset. Many recent entity matching approaches use the Dedupe model in their implementation [36]. Even though Dedupe implements parallelization to compute entity similarity, it does not scale well to millions of entities.

Deep Neural Networks - Smooth Inverse Frequency (SIF) [3] is a supervised model that considers the words present in each attribute-value pair to distinguish a match from a non-match. It works by computing the weighted average of the word vectors in the sentence and then removing the common components, which are the projections of the average vectors on their first singular vector. Original SIF implementation does not take the word order into account when computing similarity between entity pairs; however, there have been recent studies that aim to improve SIF by considering the word order too [52]. Other studies implement SIF within other projects, for example, as part of a chatbot [40], to use it in combination with principal component removal [25] and to cluster short texts [21].

Recurrent Neural Networks - In comparison to SIF, Recurrent Neural Networks (RNN) is another supervised model that [9] considers the sequences of words to identify matches. It works by learning two RNN to encode a variable-length sequence into a fixed-length vector representation and decode a given fixed-length vector representation back into a variable-length sequence. The encoder is an RNN that reads each symbol of an input sequence sequentially and changes the hidden state of the RNN at every symbol. As it reaches the end of the sequence, the hidden state of the RNN represents a summary of the whole input sequence. The decoder RNN of the proposed model is trained to generate the output sequence by predicting the next symbol given the hidden state.

Bidirectional RNN - Bidirectional RNN is a supervised hybrid model that considers the *alignment of sequences of words* in an entity. The hybrid model is essentially a combination of RNN and Attention models. It is initially proposed in the paper of Mudgal et al. [35], which concludes that a hybrid model can be up to 3 times slower to train than other neural network models. However, it has the potential to produce significantly more accurate results. The authors also conclude that a hybrid model could assign high weights to tokens that carry essential semantic information. In this case, the tokens are part of an entity. This approach is criticized by Kasai et al. [26], who explain that realistic entity matching tasks have limited

access to labeled data. The state-of-the-art performance is only achieved because the introduced hybrid model is trained on thousands of labeled data entries, which requires substantial labeling effort.

Ensemble Learning - Ensemble models are designed to combine the strong sides of many supervised and unsupervised standalone models and reduce their shortcomings simultaneously. There are nine ensemble approaches implemented in this study. It combines the standalone entity matching models using different ensemble learning techniques. (See Section III-A2)

Table I. Empirical results on DBLP-Scholar dataset. Note, all values are percentages.

Model Name	Max F1 score	Max Precision	Max Recall
Dedupe	96.48	96.89	96.07
SIF	96.85	96.59	97.12
RNN	97.77	97.95	97.59
Attention	97.82	95.98	99.72
Hybrid	98.35	97.02	99.72

2) Performance Analysis: The Hybrid model achieves the highest performance in terms of F-score and recall, while RNN achieves the highest precision in the DBLP-Scholar dataset (see Table I). Overall, the performance of all entity matching models is relatively high. However, none of the models achieves 100% F-score, which indicates that there is still room for improvement for entity matching solutions. Similarly, we test nine ensemble models with the DBLP-Scholar dataset. The aim is to observe if the relative performance of different ensemble models with the DBLP-Scholar dataset corresponds to the results obtained with the IT asset dataset. The results are displayed in Table II.

The average ensemble performed the best F-score (98.88%), similar to the ING IT asset test, where the average was the second-best model. Weighted and normalized averages also achieved high F-score performance with ING IT assets and DBLP-Scholar. In terms of precision, soft voting (99.63%) and hard voting (99.63%) produced the best results in the ING IT assets dataset. The weighted average (99.07%) had the best performance in terms of recall, which is also the case for ING IT assets. We conclude that the average ensemble is the best approach based on these two datasets. However, we would prefer soft voting to maximize precision and optimize recall. In theory, every model can achieve high precision by sacrificing recall and vice versa. However, soft voting and weighted average produce the best results with those metrics while maintaining good overall performance.

3) Selection Rules: According to the literature study, expert interviews, and the performance analysis we have conducted, we identified a list of selection rules. These rules guide the *Model Selector* with the decision-making process to find the best fitting EM model based on the use case scenarios of decision-makers at business enterprises. Part of the selection rules are presented as follows:

```
(R1) IF labeled data == FALSE THEN
    SOLUTIONS U {"Active Learning Model"}

(R2) IF the number of labeled data is less than K THEN
    SOLUTIONS U {"Active Learning Model"}

(R3) IF entities consist of one or two words THEN
    SOLUTIONS U {"Recurrent Neural Networks Model"}

(R4) IF the word order is important THEN
    SOLUTIONS U {"Bidirectional RNN Model"}

(R5) IF there is a time constraint THEN
    SOLUTIONS U {"Deep Neural Networks Model"}

(R6) IF the precision should be maximized THEN
    SOLUTIONS U {"Deep Neural Networks Model"}

(R7) IF the recall should be maximized THEN
    SOLUTIONS U {"Max. Ensemble Model"}

(R8) IF the F-score should be maximized THEN
    SOLUTIONS U {"Avg. Ensemble Model"}

    ...
```

One of the challenges of entity matching is the lack of labeled data. While the state of the art neural networks can produce good quality matching models, they also require a substantial amount of labeled data. In the absence of training data, rule 1 guides the selector to choose the Dedupe active learning model, which allows the decision-maker to create labeled entity pairs manually. Similarly, rule 2 aims to increase the volume of labeled data if the amount is below a particular value, k. This value is calculated using the volume and the schema overlap of the provided data.

The information that represents entities plays an essential role in entity matching. Recurrent Neural Networks are the most optimal model if an entity consists of a few words. If the word order plays an important role, then Bidirectional RNN should be chosen instead. If the decision-maker wants to prioritize precision, recall, or F-score in the final result, the model selector should choose one of the specified ensemble models.

Table II. Empirical results of ensemble models on DBLP-Scholar dataset. Note, all values are percentages.

Model Name	Max F1 score	Max Precision	Max Recall
Ensemble soft voting	97.64	**99.63**	95.72
Ensemble hard voting	98.31	**99.63**	97.02
Ensemble min	98.84	99.17	98.53
Ensemble max	98.66	99.17	98.16
Ensemble stacking (GradientBoostingClassifier)	98.16	99.02	97.32
Ensemble normalized average	98.84	98.70	98.98
Ensemble average	**98.88**	98.70	99.06
Ensemble weighted average	98.84	98.61	**99.07**
Ensemble bagging	98.29	98.89	97.71

IV. Case Study

The ING Group is a Dutch multinational banking and financial services corporation headquartered in Amsterdam. Its primary businesses are retail banking, direct banking, commercial banking, investment banking, wholesale banking, private banking, and insurance services.

To conduct the case study and assist the case study participants in defining their use case scenarios, we have arranged eleven interviews with different experts (five software architects and six developers) at the case study organization to understand their requirements. The case study participants stated that IT assets in their case represent applications, services, platforms, frameworks, and other tools that make up the IT landscape of ING. They gave us examples representing their IT assets (including authentication API, graph visualization platform, and credit risk calculator). To retrieve these assets, we developed a pipeline that queries some of the largest ING data sources, Configuration Management DataBase (CMDB) and API MarketPlace. Figure 2 depicts the schema of CMDB and Figure 3 depicts the schema of MarketPlace. The scope of this case study is matching CMDB configuration items of class business applications with the applications that are subscribed through the MarketPlace APIs. By design, CMDB stores every IT asset within MarketPlace. Since the information about the IT asset is stored without a common reference to the underlying entity, there is a need for an entity matching solution that can automatically match the assets.

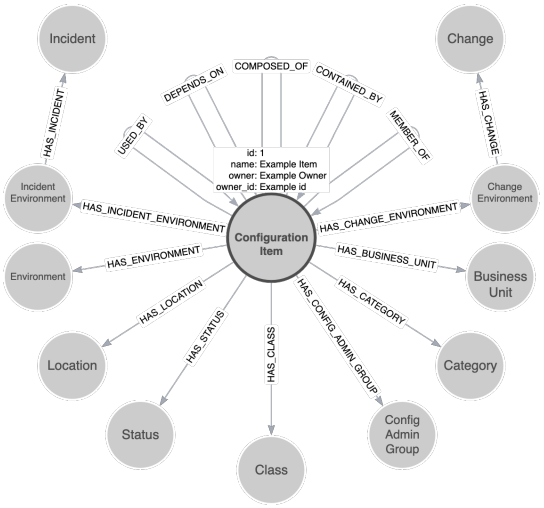

Fig. 2. CMDB schema.

Even though both data sources store some information about the IT assets, this information is not overlapping, which means the data sources are heterogeneous. The differences in schemata displayed in Figure 2 and Figure 3. The MarketPlace is the minor data source that stores a total of 5000 IT assets, while CMDB stores about 50,000 IT assets. In the current state, none of the CMDB information has any links to the MarketPlace, which means that these two data sources can be represented as two separate components. The data collection phase of the MarketPlace was performed manually, and since there is no standard naming protocol has been defined, the names of the corresponding assets often differ. Additionally, in MarketPlace, IT assets do not have any attributes. This makes it significantly more challenging to find a corresponding match, and we often need to rely on just the asset names. The MarketPlace also contains many IT assets created for

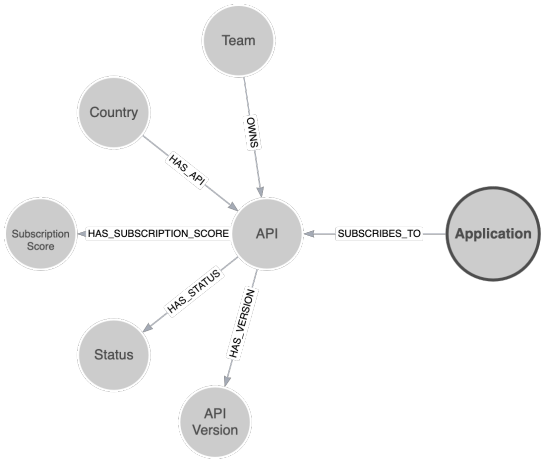

Fig. 3. API MarketPlace schema.

test purposes, which pollute the landscape. These noisy assets often contain words like 'test' or 'demo' in their names or have a random string of characters instead of the name. In Table III we identify ten challenges of entity matching within the context of ING IT assets data stores.

Table III. Characteristics of ING IT asset data that represent key challenges of entity matching.

Characteristics	Challenges
Heterogeneity	(1) Different schemata. (2) No existing links between data sources. (3) Different magnitude.
Irregularity	(4) No standard naming . (5) No external ontology used to capture semantics. (6) Data has been collected manually, typically by different users.
Noisiness	(7) Disconnected nodes. (8) Too many acronyms. (9) Spelling mistakes. (10) Multilingual entities.

The ING IT asset data was collected through crowdsourcing within the ING employee network. First, we collected the corporate keys[3], a unique id assigned to all ING employees, of every product owner[4], a team leader responsible for developing and maintaining an asset at ING. Both CMDB and MarketPlace contain corporate keys either about a product owner or an IT custodian[5], a person responsible for taking care of an asset, of an asset. Unfortunately, the corporate keys on both sides do not often match. After collecting the corporate keys from MarketPlace, we used them to send emails through Outlook. Each email was customized based on the recipient. The email would contain the recipient's name, the IT asset from MarketPlace that contains their product key information, and a message asking to help us find an exact match of the asset at the CMDB side. With this method, we received

[3]Corporate key is a unique id that is assigned to all ING employees.

[4]Product owner is a leader of the team responsible for developing and maintaining an asset.

[5]A person who has responsible for taking care of an asset.

Table IV. Empirical results of five machine learning approaches. Note, all values are percentages.

Model Name	Max F1 score	Max Precision	Max Recall
Dedupe	78.57	84.62	73.33
SIF	77.78	63.64	**100**
RNN	81.97	78.57	89.29
Attention	83.64	85.19	85.71
Hybrid	**85.19**	**100**	85.71

about 50 responses which contained connections for about 100 Marketplace IT assets with CMDB. To obtain more labeled results, we used Dedupe to generate 195 more entity pairs that perfectly match or completely disagree on multiple criteria. These were labeled as matches and non-matches, respectively. We obtained 295 labeled IT asset pairs that are used to train, validate and test entity matching models. The separation is done in the 60-20-20 manner when 60% (177 entity pairs) of the labeled dataset is used for training, 20% (59 entity pairs) for validation, and 20% (59 entity pairs) for testing.

After obtaining the labeled data, we test and compare the existing EM solutions. The primary performance metric used across many entity matching studies is precision, recall, and the computed F-score. This study will not test other performance metrics related to scalability, execution time, privacy, or security. All five machine learning models are tested independently, and each model is trained and tested using the same data and through the same sequence of steps. Each experiment is repeated ten times to eliminate the possibility of a random error. Table IV displays the max achieved results for each of the models. Note that the displayed max precision and max recall scores often come from different experiment iterations and might not be used when calculating the F-score.

The max F-score of the EM models varies between 78-85%. The hybrid model gives the best performance in terms of F-score score (85.19%) and precision (100.00%). These scores come from a relatively small testing dataset containing 59 entity pairs. These results should only indicate the performance, and with more testing data, the precision would likely drop below 100%. In terms of recall, SIF produced the highest score (100.00%). However, it comes with the cost of precision (63.64%), making it the worst-performing model in terms of the F-score. While the hybrid model achieves the best results out of all five models, the attention model comes close to the precision and does equally well in the recall. RNN also achieves comparable results to hybrid by having a higher recall (89.29%) but losing in terms of precision (78.57%). Dedupe is overall one of the worst-performing models. However, this model can perform entity matching even with the lack of labeled data. These five models are then used to create and test nine ensemble models. The results are displayed in Table V.

The difference in performance between the ensemble models is similar to the standalone models; however, the overall performance is higher. The F1 score ranges between 84-92%. This indicates that combining five entity matching models into ensemble models is beneficial since the worst performing model, soft voting with 82.35% F1 score, produces comparable results to the best standalone hybrid model. However, the best ensemble model, bagging, achieves higher results (91.52%) than the hybrid model by performing better in recall but falling short in precision. Arguably, bagging is the best performing model out of nine implemented models. Its performance fluctuates significantly with each consecutive run of the entity matching pipeline. This is because parts of testing data are assigned randomly to each model, and the results are combined. So by rearranging the order of the model or the order of the testing data, the model can produce a completely different result than the one displayed in Table V. With this in mind, either ensemble average or weighted average can be a better option for entity matching since the order of the models or the testing data does not matter for the final result. The averaging ensembles take all outputs of each model into account and combine them into one final score. Of course, while this might be a 'safer' option for entity matching, it comes at the cost of computation time. The averaging models must wait for each standalone model to produce their score before combining it into the final score. Accordingly, this does not scale well for most entity matching problems. On the contrary, the bagging ensemble does not have this problem, even with a much higher number of models included in the ensemble. The fact that only one model is computing the result for each entity pair makes this model produce its final result much faster than any other ensemble model mentioned in Table V.

Table V. The results of nine ensemble approaches. Note, all values are percentages.

Model Name	Max F1 score	Max Precision	Max Recall
Ensemble soft voting	82.35	**100**	70
Ensemble hard voting	84.37	96.42	75
Ensemble min	85.71	75	**100**
Ensemble max	87.27	85.71	88.89
Ensemble stacking (GradientBoostingClassifier)	87.72	89.29	86.21
Ensemble normalized average	88	78.57	**100**
Ensemble average	90.20	82.14	**100**
Ensemble weighted average	90.20	82.14	**100**
Ensemble bagging	**91.53**	96.43	87.10

The voting ensembles are two of the worst implemented ensemble models. The soft voting ensembles achieve an 82.35% F1 score, and hard voting achieves 84.37% F1. While their F1 scores are substantially worse than the other ensemble models, they are comparable to the best standalone models. The shortcoming of voting ensembles for entity matching is the low recall. The soft voting ensemble has 70% recall, which is the worst recall out of all ensemble and standalone models. However, this seems to be balanced by the high precision scores, where soft voting achieved 100%, and hard voting achieved 96.42%. This seems to align with the general pattern between precision and recall in entity matching.

Higher precision can result in lower recall and vice versa. Most standalone and ensemble models follow this general pattern. This makes sense if we think of entity matches, true positives, false positives, and false negatives. If we take a dataset that has an equal amount of positive and negative entity

matches, let us say 500 pairs that are a match and 500 that are not, and we assume that they are all a match, then the recall of this example will be 100%, and the precision is 50%. This can be used as a baseline for the entity matching the ING IT asset data because the ratio between positive and negative matches in the ground truth is one-to-one.

Another scenario is when out of possible 1000 entity matches, we only consider the entities that match flawlessly as a match. Out of 1000 pairs of entities, let us assume only 10 are a match, then the precision is likely to be close to 100%, and the recall will be extremely low since we only output 10 out of 500 correct matches. This trade-off is good to keep in mind since entity matching has different applications in the real world. For example, if we wanted to use matched entities to create an item-based recommendation engine for a movie database, we would probably favor recall over precision since the goal here is to recommend as many movies for the decision-maker to check and hope that one of them grabs their attention. Four ensemble models implemented in this study achieve 100% recall, meaning they could be entirely for the recommendations that value recall. Of course, there are different types of recommendation engines, and some of them value precision over recall, for example recommending medicine substitutes for patients. Soft voting is most likely the best choice for this task since it collects votes from every model, making the diverse model input. Soft voting also considers the probabilities of each model to give the correct prediction. The concept of soft voting can be interpreted as a weighted voting scheme.

The stacking ensemble performs quite well overall with 89.29% precision and 86.21% recall. It is possible to be one of the best all-around choices for entity matching on ING IT asset data. While there are problems that require a high minimum threshold for precision and recall, matching ING IT asset data most likely requires a balance between these two, with relatively high scores for both. Again, stacking ensemble does not achieve as high overall performance as bagging; however, it has minor variance in results and has high computation costs and programming complexity. Max ensemble has a similar performance to stacking, with an F1 score of 87.27%. It has a slightly better recall than stacking but a lower precision. Min ensemble performs relatively worse than most other models; however, it performs better than any standalone model. While the min ensemble achieved 100% recall, it does not achieve as good precision as any other model tested in this study, and it loses in terms of recall to other models that also achieved 100% recall. Arguably min ensemble is the worst-performing model with the ING IT asset data since there might not be a good application that would choose min ensemble over any other ensemble approaches implemented and tested in this study.

The second part of the experiment evaluates the EM model performance against human experts. We asked ten DevOps engineers at ING to assign similarity and confidence scores for a provided set of entity pairs. In this case, the confidence score is a weight in the weighted average computation between similarity scores. To visualize the difference between the ING expert scores and model scores, we plot them alongside each other in Figure 4. The plot contains many outliers, which indicates that the EM models and human experts can sometimes completely disagree in their decisions. In Figure 4, the blue line represents the expert scores, and the other lines represent EM models. All results are sorted based on the expert scores. This plot aims to visualize how close the scores produced by each EM model coincide with the expert scores. It can also be represented using the mean square error, as shown in Table IV.

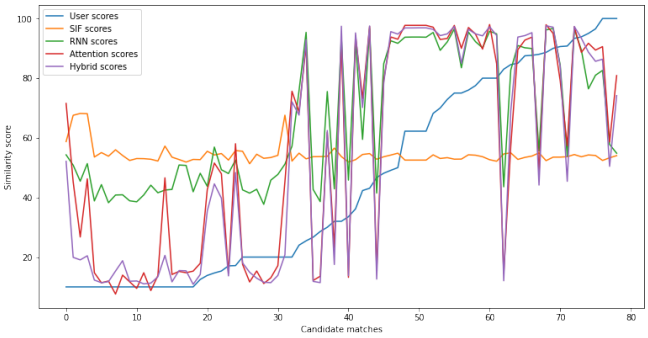

Fig. 4. A comparison between the case study participants' scores and the entity matching models.

Table VI. Mean squared error of the entity matching models.

Model Name	Mean squared error (%)
Dedupe	36.12
SIF	34.89
RNN	31.19
Attention	27.89
Hybrid	26.73

V. Discussion

The case study results show that the proposed framework can become an effective tool for researchers to find the matching model's most suitable entity. We show that the framework adapts well to different datasets and settings. The framework can choose which entity matching solutions are used for specific use cases and allows the user to replace existing models with newer ones. The framework has also been tested for generalizability with two datasets, where both of the experiments agree on the results.

A. Addressing the Research Questions

We combine results from a literature review, design science research, case study research, and an experiment to design an entity matching framework. We have successfully implemented a framework based on design science research and tested it with a specific use case and an open-source dataset. The case study research allowed us to gain insights into selecting and using entity matching tools within a business enterprise. We conducted several expert interviews to evaluate our hypothesis.

We conducted a systematic literature review to compare our set of entity matching models. We found five entity matching models, namely Dedupe, SIF, RNN, Attention, and Hybrid, by exploring state-of-the-art. Dedupe stands out the most from these models because it implements active learning to increase the number of labeled entity pairs. The other four models implement neural networks. SIF and Attention models do not consider the word order, often leading to misrecognized entity pairs. While Hybrid performs better than other models, the computation time is also considerably longer. In addition, the literature review revealed potential gaps for future exploration, which include progressive learning and transfer learning for entity matching.

To improve the performance of entity matching models, we combined them into nine ensembles. To test their performance, we used two datasets, ING IT assets and DBLP-Scholar. Most of the ensemble approaches outperformed state-of-the-art entity matching models. This is explained by ensemble models combining the strength of the standalone models, which minimizes their weaknesses. The bagging ensemble achieves the best performance with the IT asset data; however, it is also one of the least reliable models. The average ensemble shows a better consistent performance across each of the datasets. Soft voting, hard voting, and min achieve good precision and can be used for high precision tasks. Weighted average and normalized average can be used for high recall tasks.

To evaluate the selected entity matching models, we conducted a case study within the context of the ING banking organization. We conducted interviews with ten domain experts during this case study to help us evaluate the selected entity matching models. The experts were asked to perform entity matching manually by assigning similarity and confidence scores to the presented entity pairs. The mean square error has to be computed for each model. The two models with the lowest mean square error are Hybrid and Attention. During the case study, we also evaluated a number of possible requirements that users might input into the framework.

B. Lessons learned

Selecting entity matching models is a challenging task that requires more research to be done. Many entity matching approaches require a substantial amount of labeled data to produce adequate results. However, labeling entity pairs manually is time-consuming and challenging since it requires sufficient knowledge about the underlying data. As part of the ING case study, most entity pairs were labeled with low confidence scores due to the lack of in-depth knowledge. In addition, enterprise datasets may also contain significant noise. While noisy entities can be straightforward, some data can be challenging to filter out by non-experts. If a dataset contains many entities created for testing purposes, it is difficult to identify meaningful relationships among them. Lastly, a fundamental problem with creating ensemble approaches is that many existing entity matching solutions are not open-source or do not provide sufficient detail for the reproducibility of the results. This issue was also pointed out by Koutras et al. [30]. The existing entity matching solutions require specific parameter settings and in-depth knowledge, which the paper often does not provide.

C. Threats to validity

1) Construct validity: To compare the performance of EM models, we measure their precision and recall and compute the F-score on the testing data. This, however, only gives a shallow overview of the performance of an EM model. It might be more beneficial to compute precision, recall, or Hits [49]. These metrics are often used with recommendations or search engines. However, it can be applied to entity matching as well. The only requirement is that the output of the entity matching solution should contain top k matches for every entity that is being matched. This way, we can more accurately evaluate how well the solution performs entity matching.

2) External validity: As part of this study, we focus on the generalizability of entity matching solutions. When performing entity matching with a different dataset or in a different setting, the solution should adapt to the specific requirements. Our proposed solution adapts well to the volume and the quality of the labeled data provided. The framework also generalizes well with use cases that have underlying time constraints. Moreover, the framework is designed to stay relevant in the future by allowing users to replace the existing models with newer ones. The framework does not generalize well concerning the input format. Currently, the framework accepts data only in CSV file format, and it should be structured in a specific way that can be readable by the EM models. However, all experiments can be reproduced by following a predefined set of steps and the same datasets that are used in this study.

3) Internal validity: Throughout the experiments, we question the improved performance of the ensemble models over the existing entity matching solutions. Specifically, we test whether this improvement comes from combining the model into ensembles or is due to other reasons, like chance. To explore this cause and effect, we conducted two experiments that showed similar results, where ensemble models outperformed existing EM solutions with both datasets. We also ran each experiment ten times to eliminate any random outliers shown in the performance of every model. One possible threat to the internal validity is the lack of labeled data, which causes the performance of each model to fluctuate by 2-3% per experiment. The threshold value for identifying whether a pair of entities is a match change per dataset/model. This value is chosen by selecting the optimal value in the precision-recall curve. However, there is no proof that this value is the best possible value. Confirming this would require more experiments.

4) Conclusion validity: The conclusions of this research come from the literature review, case study, and an experiment with the open-source dataset. The decision about the performance of the ensemble model is derived using empirical evaluation. The domain experts and empirical evidence verify the claims about the adaptivity and performance of the proposed framework. The conclusion on the improved entity matching

performance serves only as an indication because there is not enough data to make solid proof of the advantages of using ensemble models with enterprise datasets.

VI. Related Work

In this study, We have conducted a *systematic literature review* based on the guidelines of Kitchenham [27] to explore state of the art and conduct a gap analysis to position our study among other efforts in the literature. Accordingly, we have observed a comprehensive list of studies conducted in the literature regarding EM models and their improvements. A subset of the selected studies is presented in this section.

Some studies employed machine learning approaches to select a set of models based on their performance in a particular setting. For instance, Chiew et al. [8] introduced a framework for selecting ensemble features in phishing detection systems. They also proposed an algorithm to determine the optimal number of features automatically. Cai et al. [7] suggested a feature selection approach based on deep learning for removing irrelevant and redundant data to reduce computation time and improve learning accuracy. Shen et al. [43] developed an unsupervised model selection algorithm, based on the technique of weighted rank aggregation, to choose the parameter settings automatically and model inter-entity relationships, and capture entity type information. Moreover, some researchers conducted an extensive literature study and suggested benchmarks to select models based on their documented characteristics. For example, Köpcke and Rahm [29] presented a framework for entity matching and compared eleven EM models based on a collection of features against each other. Additionally, we observed that some studies such as [47], [15], [13] employed Multi-Criteria Decision-Making (MCDM) approaches for building decision models for selecting models based on decision-makers' requirements and preferences.

Studies based on benchmarking and statistical analysis are typically time-consuming approaches and mainly apply to a limited set of alternatives and criteria. Decision-making based on such analysis can be challenging as decision-makers cannot assess all their requirements and preferences simultaneously, especially when the number of requirements and alternatives is significantly high. Furthermore, benchmarking and statistical analysis are likely to become outdated soon and should be kept up to date continuously, which involves a high-cost process. Most of the MCDM techniques are mainly appropriate for specific case studies. Furthermore, the results of MCDM approaches are valid for a specified period; therefore, such studies will be outdated after a while. Additionally, the pairwise comparison is typically considered the main method to assess the weight of criteria in MCDM techniques. Typically, MCDM approaches are not scalable, so in modifying the list of alternatives or criteria, the whole process of evaluation should be redone. Therefore, these methods are costly and applicable to only a few criteria and alternatives. This study proposed an extendable and adaptable framework that combines tacit and explicit knowledge from literature studies and domain experts besides performance analysis to select EM models for decision-makers at business enterprises. We believe that the captured knowledge regarding models should constantly be updated, and new EM models should always be added to the EM models pool.

VII. Conclusion

In this study, we present an EM model selection framework, which automatically adapts to the features of the provided dataset. As the result of a systematic literature review, the framework implements a number of EM approaches, including active learning, deep neural networks, recurrent neural networks, and more. The framework is evaluated using a case study from ING and an open-source dataset. The literature study also revealed a research gap concerning ensemble learning in the context of entity matching. We implement nine ensemble models and add them to the model selection pool as part of this research. According to the empirical evidence, ensemble models can outperform standalone EM models on both datasets. Another advantage of the proposed framework is its ability to evolve by replacing old EM models with improved ones.

We believe that further evaluation of the framework with more datasets is required, such as DBLP-ACM, Amazon-GoogleProducts, and Abt-Buy open-source datasets. However, it is also important to assess the framework in the enterprise setting because it provides unique challenges to entity matching. Moreover, it should be possible to add more entity matching models and evaluate their effect on the overall performance of the framework. Some potential candidates that can be added to the EM model pool are sequence-to-sequence [37], hierarchical [17] and multi-perspective [18] matching models. It is also possible to conduct further research on the enterprise applications of entity matching, either within a recommendation system or a search engine that can parse a user query and return matched entities based on that query. Another potential application is a data visualization tool that displays the links between the matched entities, possibly with a similarity score as a weight of the link.

Acknowledgment

This work has been partially supported by the ING bank, and partially funded by the European Union's Horizon 2020 research and innovation program, by the project of ARTICONF (825134), ENVRI-FAIR (824068), and BLUECLOUD (862409). The authors would like to give special thanks to ICAI "AI for Fintech Lab", the iGraph squad, and all the participants of the entity matching evaluation.

References

[1] *State-of-the-Art Instance Matching Methods for Knowledge Graphs*. Zenodo, Oct. 2021.

[2] H. A. Ahmad and H. Wang. An effective weighted rule-based method for entity resolution. *Distributed and Parallel Databases*, 36(3):593–612, 2018.

[3] S. Arora, Y. Liang, and T. Ma. A simple but tough-to-beat baseline for sentence embeddings. In *International conference on learning representations*, 2017.

[4] I. Bhattacharya and L. Getoor. *Entity Resolution*, pages 402–408. Springer US, Boston, MA, 2017.

[5] M. Bilenko. Learnable similarity functions and their applications to clustering and record linkage. In *AAAI*, volume 4, pages 981–982, 2004.

[6] U. Brunner and K. Stockinger. Entity matching on unstructured data: an active learning approach. In *2019 6th Swiss Conference on Data Science (SDS)*, pages 97–102. IEEE, 2019.

[7] J. Cai, J. Luo, S. Wang, and S. Yang. Feature selection in machine learning: A new perspective. *Neurocomputing*, 300:70–79, 2018.

[8] K. L. Chiew, C. L. Tan, K. Wong, K. S. Yong, and W. K. Tiong. A new hybrid ensemble feature selection framework for machine learning-based phishing detection system. *Information Sciences*, 484:153–166, 2019.

[9] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation, 2014.

[10] V. Christophides, V. Efthymiou, T. Palpanas, G. Papadakis, and K. Stefanidis. End-to-end entity resolution for big data: A survey. *arXiv preprint arXiv:1905.06397*, 2019.

[11] A. Doan and A. Y. Halevy. Semantic integration research in the database community: A brief survey. *AI magazine*, 26(1):83–83, 2005.

[12] J. Enríquez, F. Domínguez-Mayo, M. Escalona, M. Ross, and G. Staples. Entity reconciliation in big data sources: A systematic mapping study. *Expert Systems with Applications*, 80:14–27, 2017.

[13] S. Farshidi and S. Jansen. A decision support system for pattern-driven software architecture. In *Proceedings of the 14th European Conference on Software Architecture, ECSA 2020,*, volume 1, pages 1–12. ACM, 2020.

[14] S. Farshidi, S. Jansen, and M. Deldar. A decision model for programming language ecosystem selection: Seven industry case studies. *Information and Software Technology*, 139:106640, 2021.

[15] S. Farshidi, S. Jansen, and S. Fortuin. Model-driven development platform selection: Four industry case studies. http://dx.doi.org/10.17632/fbg29x5vkk.1, 2020.

[16] D. Fortus, J. Krajcik, R. C. Dershimer, R. W. Marx, and R. Mamlok-Naaman. Design-based science and real-world problem-solving. *International Journal of Science Education*, 27(7):855–879, 2005.

[17] C. Fu, X. Han, J. He, and L. Sun. Hierarchical matching network for heterogeneous entity resolution. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, pages 3665–3671, 2021.

[18] C. Fu, X. Han, L. Sun, B. Chen, W. Zhang, S. Wu, and H. Kong. End-to-end multi-perspective matching for entity resolution. In *IJCAI*, 2019.

[19] B. Gautam, O. R. Terrades, J. M. Pujades, and M. Valls. Knowledge graph based methods for record linkage. *ArXiv*, abs/2003.03136, 2020.

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

[21] A. Hadifar, L. Sterckx, T. Demeester, and C. Develder. A self-training approach for short text clustering. In *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*, pages 194–199, 2019.

[22] A. R. Hevner, S. T. March, J. Park, and S. Ram. Design science in information systems research. *MIS quarterly*, pages 75–105, 2004.

[23] S. Jansen. Applied multi-case research in a mixed-method research project: Customer configuration updating improvement. In *Information Systems Research Methods, Epistemology, and Applications*, pages 120–139. IGI Global, 2009.

[24] R. B. Johnson and A. J. Onwuegbuzie. Mixed methods research: A research paradigm whose time has come. *Educational researcher*, 33(7):14–26, 2004.

[25] A. Karipbayeva, A. Sorokina, and Z. Assylbekov. A critique of the smooth inverse frequency sentence embeddings. *arXiv preprint arXiv:1909.13494*, 2019.

[26] J. Kasai, K. Qian, S. Gurajada, Y. Li, and L. Popa. Low-resource deep entity resolution with transfer and active learning. *arXiv preprint arXiv:1906.08042*, 2019.

[27] B. Kitchenham. Procedures for performing systematic reviews. *Keele, UK, Keele University*, 33(2004):1–26, 2004.

[28] P. V. Konda. *Magellan: Toward building entity matching management systems*. The University of Wisconsin-Madison, 2018.

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

[30] C. Koutras, G. Siachamis, A. Ionescu, K. Psarakis, J. Brons, M. Fragkoulis, C. Lofi, A. Bonifati, and A. Katsifodimos. Valentine: Evaluating matching techniques for dataset discovery. In *2021 IEEE 37th International Conference on Data Engineering (ICDE)*, pages 468–479. IEEE, 2021.

[31] Y. Li, J. Li, Y. Suhara, A. Doan, and W.-C. Tan. Deep entity matching with pre-trained language models. *arXiv preprint arXiv:2004.00584*, 2020.

[32] X. Liao, J. Bottelier, and Z. Zhao. A Column Styled Composable Schema Matcher for Semantic Data-Types. *Data Science Journal*, 18:25, June 2019.

[33] X. Liao and Z. Zhao. Unsupervised Approaches for Textual Semantic Annotation, A Survey. *ACM Computing Surveys*, 52(4):1–45, July 2020.

[34] J. R. Meredith, A. Raturi, K. Amoako-Gyampah, and B. Kaplan. Alternative research paradigms in operations. *Journal of operations management*, 8(4):297–326, 1989.

[35] S. Mudgal, H. Li, T. Rekatsinas, A. Doan, Y. Park, G. Krishnan, R. Deep, E. Arcaute, and V. Raghavendra. Deep learning for entity matching: A design space exploration. In *Proceedings of the 2018 International Conference on Management of Data*, pages 19–34, 2018.

[36] A. Ngueilbaye, H. Wang, D. A. Mahamat, and I. A. Elgendy. Sdler: stacked dedupe learning for entity resolution in big data era. *The Journal of Supercomputing*, pages 1–25, 2021.

[37] H. Nie, X. Han, B. He, L. Sun, B. Chen, W. Zhang, S. Wu, and H. Kong. Deep sequence-to-sequence entity matching for heterogeneous entity resolution. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 629–638, 2019.

[38] G. Papadakis, G. Mandilaras, L. Gagliardelli, G. Simonini, E. Thanos, G. Giannakopoulos, S. Bergamaschi, T. Palpanas, and M. Koubarakis. Three-dimensional entity resolution with jedai. *Information Systems*, 93:101565, 2020.

[39] H. Rosales-Méndez, B. Poblete, and A. Hogan. What should entity linking link? In *AMW*, 2018.

[40] S. Ruan, L. Jiang, J. Xu, B. J.-K. Tham, Z. Qiu, Y. Zhu, E. L. Murnane, E. Brunskill, and J. A. Landay. *QuizBot: A Dialogue-Based Adaptive Learning System for Factual Knowledge*, page 1–13. Association for Computing Machinery, New York, NY, USA, 2019.

[41] A. Saeedi, M. Nentwig, E. Peukert, and E. Rahm. Scalable matching and clustering of entities with famer. *Complex Systems Informatics and Modeling Quarterly*, pages 61–83, 2018.

[42] C. Shao, L.-M. Hu, J.-Z. Li, Z.-C. Wang, T. Chung, and J.-B. Xia. Rimom-im: A novel iterative framework for instance matching. *Journal of computer science and technology*, 31(1):185–197, 2016.

[43] J. Shen, J. Xiao, X. He, J. Shang, S. Sinha, and J. Han. Entity set search of scientific literature: An unsupervised ranking approach. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 565–574, 2018.

[44] H. A. Simon. *The Sciences of the Artificial (3rd Ed.)*. MIT Press, Cambridge, MA, USA, 1996.

[45] Z. Sun, W. Hu, and C. Li. Cross-lingual entity alignment via joint attribute-preserving embedding. In *International Semantic Web Conference*, pages 628–644. Springer, 2017.

[46] A. Thor and E. Rahm. Moma-a mapping-based object matching system. In *CIDR*, volume 2007, pages 247–258, 2007.

[47] M. Y. L. Vazquezl, L. A. B. Peñafiel, S. X. S. Muñoz, and M. A. Q. Martinez. A framework for selecting machine learning models using topsis. In *International Conference on Applied Human Factors and Ergonomics*, pages 119–126. Springer, 2020.

[48] Y. Yan, S. Meyles, A. Haghighi, and D. Suciu. Entity matching in the wild: A consistent and versatile framework to unify data in industrial applications. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*, pages 2287–2301, 2020.

[49] K. Yang, S. Liu, J. Zhao, Y. Wang, and B. Xie. Cotsae: Co-training of structure and attribute embeddings for entity alignment. In *AAAI*, 2020.

[50] R. K. Yin. The case study as a serious research strategy. *Knowledge*, 3(1):97–114, 1981.

[51] R. K. Yin. *Case study research and applications: Design and methods*. Sage publications, 2017.

[52] Y. M. Yuan Ye, Liu Jiming. Research on calculation method of text similarity based on smooth inverse frequency. *The Journal of China Universities of Posts and Telecommunications*, 27(2):56–64, 2020.

[53] Y. Zhu, H. Liu, Z. Wu, and Y. Du. Relation-aware neighborhood matching model for entity alignment. In *AAAI*, 2021.