Experiences and Lessons Learned from the SIGMOD Entity Resolution Programming Contests

Andrea De Angelis, Maurizio Mazzei,
Federico Piai, Paolo Merialdo
Roma Tre University
{name.surname}@uniroma3.it

Giovanni Simonini, Luca Zecchini,
Sonia Bergamaschi
University of Modena and Reggio Emilia
{name.surname}@unimore.it

Donatella Firmani
Sapienza University
donatella.firmani@uniroma1.it

Xu Chu, Peng Li, Renzhi Wu
Georgia Institute of Technology
{name.surname}@cc.gatech.edu

ABSTRACT

We report our experience in running three editions (2020, 2021, 2022) of the SIGMOD programming contest, a well-known event for students to engage in solving exciting data management problems. During this period we had the opportunity of introducing participants to the entity resolution task, which is of paramount importance in the data integration community. We aim at sharing the executive decisions, made by the people co-authoring this report, and the lessons learned.

1. INTRODUCTION

The SIGMOD conference organizes a programming contest every year for student teams from degree-granting institutions, able to attract many of the world's leading research groups active in the field of data management and sparking new ideas for future technologies (notably, ForestDB [1] came out of the contest in 2011). We had the opportunity to chair this event for three consecutive editions, on the occasion of SIGMOD 2020[1], 2021[2], and 2022[3]. In particular, our organizing team included the DBGroup of the Roma Tre University (2020 and 2021), the DBGroup of the University of Modena and Reggio Emilia (2021 and 2022), and the Chu Data Lab of the Georgia Institute of Technology (2022).

Entity Resolution (ER) is the task of detecting records in one or more datasets referring to the same real-world entity [5] and represents one of the main research topics of our groups. Therefore, we decided to focus our contests for the first time on this fundamental and challenging task. The first two editions were mainly focused on the matching step from ER pipeline, while the third one moved the attention to blocking (i.e., quickly filter out tuple pairs that are unlikely to match), which plays a paramount role for scaling ER in big data scenario. All editions registered a significant number of participants: 53 teams from 16 countries in 2020, 51 teams from 12 countries in 2021 (with ≈1500 submissions), and 60 teams from 10 countries in 2022 (with ≈2500 submissions). The Roma Tre team also organized two satellite challenges for the DI2KG workshops co-located with KDD 2019[4] and VLDB 2020[5].

2. DECISIONS

Choice of the Datasets. Providing original and challenging datasets can be a burdensome task. Yet, it is fundamental to increase the engagement and inspire original solutions, especially if the dataset comes together with a manually curated ground truth (i.e., *gold standard*).

In SIGMOD 2020, we started with a product dataset containing 30k camera specifications, collected from several different e-commerce websites [6]. The misalignment of the attributes of the dataset, which had never been publicly released before, contributed to making the task more challenging. In SIGMOD 2021, we decided to employ multiple product datasets, allowing companies to participate as technical sponsors and provide their own data. Challenges can raise interest among companies and although only one of them (i.e., Altosight[6]) was able to prepare the data in time, we received

[1] `http://www.inf.uniroma3.it/db/sigmod2020contest`

[2] `https://dbgroup.ing.unimore.it/sigmod21contest`

[3] `https://dbgroup.ing.unimore.it/sigmod22contest`

[4] `http://di2kg.inf.uniroma3.it/2019`

[5] `http://di2kg.inf.uniroma3.it/2020`

[6] `https://altosight.com`

several expressions of interest[7]. From the provided dataset[8], composed of 14k USB stick specifications, and from another product dataset containing 24k notebook specifications [6], we generated multiple subsets (one from the former, two from the latter), with a size between 500 and 1.5k records, to be used in the contest. In SIGMOD 2022, for the blocking task, we decided to generate two synthetic datasets[9] of about 1M records each, obtained from the ones employed in the 2021 contest using a script operating in two steps: *(i)* generating a new tuple by picking the first word from a randomly chosen tuple, the second one from another randomly chosen tuple, etc.; *(ii)* generating its matching tuples by randomly shuffling words, deleting some words, or changing the letter case. The third twin dataset in [6], comprising about 17k monitor specifications, was used only in the satellite events.

The datasets for the ER challenges were of medium-large size, which was acceptable given the emphasis on effectiveness. For the blocking challenge, we had to produce synthetic datasets, due to the lack of such massive labeled real-world datasets.

Partitioning of the Datasets. In most online programming challenges there are two main portions of the datasets: one that is visible and one that is hidden. The visible portion can be accessed by the participants during the challenge, while the hidden one can be accessed solely by the organizers and is aimed at determining the leaderboard. In an ER/blocking challenge there are naturally many strategies for hiding portions of a dataset: *(i)* declaring a subset of the ground truth hidden, while letting participants access all the records; *(ii)* declaring some records hidden, possibly resulting in some entities that are partially hidden (i.e., with only some records visible); *(iii)* declaring some entities hidden and hiding all the associated records and ground truth. We wanted the visible portion to contain both enough easy cases to encourage the participants and enough difficult ones to maintain the engagement. We also did not want the leaderboard to change completely with and without the hidden set. To make this assessment, we ran state-of-the-art tools (e.g., [11, 12]) with different sampling criteria for selecting which entities, records, and ground truth data had to be hidden.

In 2020, we hid a subset of the ground truth, which was visible only for a small subset of entities, selected according to their size (stratified sampling). The solutions were then evaluated on a hidden set of matches, disjointed in terms of edges from the public one. Subsequently, we decided to hide also some records and entities with the same selection criteria, resulting in only a portion of the records from the employed datasets (with the related gold standard) made available to the participants.

What worked best was a combination of the above strategies. Hiding records and entities is one natural way to increase the difficulty of the task. Other solutions could be hiding entities based on different properties than size or even anonymize them.

Evaluation Metrics. The literature is rich in evaluation criteria for ER and blocking systems, such as traditional F-measure, progressive F-measure, size of the training set, and of course running time. Also, for blocking systems only, authors proposed specific metrics such as quality and completeness.

First, we wanted participants to focus on effectiveness over efficiency. Second, the selected evaluation metric had to be effectively computed without accessing the participants' code (e.g., computing progressive F-measure requires to instrument comparison operations). Third, we wanted the selected metric to be well-known to increase engagement.

For these reasons, for the ER challenges we decided to use traditional F-measure as a primary evaluation score, considering the matches detected by the solutions. In case of multiple datasets, we decided to aggregate the results achieved on each dataset considering macro F-measure, since all entities have the same importance to us. According to the same principles, for the blocking challenge we decided to use recall as a primary evaluation score, computed on the first 1M and 2M candidates detected by the solutions on the notebook and Altosight synthetic datasets, respectively. In case of ties, running time measured during the final reproducibility test was used for tie-break to assign the money prizes in 2020 (when all top-5 teams reached a 0.99 F-measure), while in 2021 the runner-up prize was distributed between two teams. Furthermore, in all challenges running time was used to set a timeout, during the final reproducibility test only (2020) or while evaluating each submission.

We learned that using running time as a tie-break policy, given the relatively short duration of the contest, might lead some participants to spend possibly too much effort on technical issues which might be of secondary interest in an ER contest (e.g., data structure optimization). Finally, we learned that valuing effectiveness over efficiency can leave interesting things for the ER community (such as block-

[7] As effective ways to encourage companies to participate, we offered to help prepare their data as well as describing their activities to young students as potential future hires.

[8] `https://github.com/dbmodena/BrewER/tree/main/altosight_ds`

[9] `https://github.com/wurenzhi/SIGMOD2022_contest_datasets`

ing) outside the challenge. This lesson inspired us to focus the last contest directly on blocking.

Submission Format. One of the most difficult decisions to take was about the submission format, since this aspect can considerably affect the challenge. In fact, there are many challenges out there with wildly different submission formats. Some challenges ask participants to submit a plain-text solution with the problem result, others ask for code in the form for instance of a Python script, while others allow for a more complex format, such as a virtual machine.

Plain-text submission was used in our first challenge, when each team only had to submit the CSV file containing the detected matches. This solution clearly leaves maximum freedom to the participants, who can make a quick start and use their own tools while focusing on the problem at hand. It also has the lightest load on the evaluation system, only requiring to process the solution file and therefore consuming less energy. The validation of the submitted solutions can be left as an offline task at the end of the challenge and carried out analogously to conference reproducibility efforts. As plain-text submission might not be compatible with strategies that hide more than the ground truth, our later two challenges featured a more sophisticated submission format that enabled participants to submit entire code and obtain the results on hidden records and entities. We designed our submission format on top of the evaluation system inherited from the previous contests (running in 2021 on two of our servers, in 2022 on MS Azure) in combination with the ReproZip[10] package [4]. Participants were required to submit the RPZ bundle created by tracking the system calls produced when running their solution on the public versions of the datasets. The submitted bundle was inserted in the evaluation queue and run in a Docker container on the evaluation server with a fixed timeout (e.g., 25/35 min.) during the ReproUnzip phase, replacing the public versions of the datasets with the secret ones.

In order to mitigate the complexity of using our submission format, we provided participants with a quick start package, containing all the necessary components to run a simple baseline solution. Moreover, in 2021 the Snowman[11] tool [13] was provided as a part of this package, to help the teams in evaluating the performance of their solutions.

ReproZip was chosen as it is fast and easy to use (a few commands are enough to create the bundle and reproduce the solution). Once participants were able to overcome the initial difficulties, the tool generally proved to be usable and effective. In order to allow a sort of debugging to address technical issues on the hidden datasets, we decided to make available to each team the `stderr` log files produced by its submissions. A Google Groups forum was also used to allow teams to interact with us and among themselves about frequent or challenging problems. In our experience, the latter submission format had a positive impact on the challenge, promoting reproducibility culture among participants and creating a more thrilling challenge experience.

Engagement. We observe that ER and blocking are not so widely-known tasks among students. Before the challenge, in order to attract more participants, we decided to use invitations to personal contacts, including the ones made during the organization of the satellite events, and spread-the-word. Both channels turned out to have played an essential role to bootstrap participation in all our challenges.

During the challenge, the main sources of engagement are the interaction with the leaderboard and the disclosure of new data (i.e., larger portions of the dataset or entire new datasets made visible).

While in 2020 the leaderboard was updated every 24 hours, resulting in daily submissions regularly distributed during the challenge, the next two leaderboards were instead uploaded in real-time, with an evaluation queue, resulting in submissions being more erratic, peaking in the last few days. Therefore, if during the first contest we gave participants the opportunity to create anonymous teams for testing preliminary or alternative approaches, in the last two challenges we forbade the use of satellite teams and limited the number of submissions per team, to prevent both the solution over-tuning and the overloading of the evaluation system.

Regarding data disclosure, while in 2020 and 2022 all public data was available from the beginning (with just a later release of a wider gold standard during the first contest), in 2021 we decided to structure the challenge into three phases, with the leaderboard being reset at the beginning of each one. During the first phase the participants started operating on a toy subset of the notebook dataset. Then, we released the first official notebook subset, joined later by the second notebook subset and the Altosight one to determine the final leaderboard.

An observed negative trend is that many teams tend to lose interest when the end of the challenge is approaching if the scores of the top teams appear too far to be reached. A possible intervention might be the introduction of an additional prize for creativity assigned by a committee, in order to reward the most original solution devised for the task at

[10] `https://github.com/VIDA-NYU/reprozip`

[11] `https://github.com/HPI-Information-Systems/snowman`

hand and maintain the engagement of those teams ranking far from the top of the leaderboard. Other solutions could be using additional metrics to evaluate solution such as code quality metrics (e.g., code complexity and test coverage) with their own prices.

Rewards. Previous SIGMOD programming contest rewards consist of a monetary prize for the winning and the runner-up teams and in a travel grant for the top-5 teams to attend the SIGMOD conference and present their solution during the poster session. Unfortunately, the first two challenges were run during COVID-19 pandemic and thus both the travel grant and the poster presentation were canceled, leaving room only for the two monetary prizes (7k and 3k USD in 2020 and 2021, 4k and 2k USD in 2022) and making the other finalist positions much less rewarding. Therefore, we also decided to invite the finalists to submit a paper describing their solution to the DI2KG 2020 workshop. This decision was more successful, with several finalists presenting their work at the workshop [2, 9, 18] and receiving extensive feedback from the audience.

The monetary prizes were offered by the contest sponsor. Microsoft sponsored all three editions of the competition and was joined in 2020 by Megagon Labs and in 2021 by SequoiaDB and Huawei.

For all the three challenges we had one winner and one runner-up, as traditionally done in previous SIGMOD programming contests. Nonetheless, we considered having a more complex reward structure, with different *tracks* that could acknowledge the complexity of ranking solutions with different approaches (e.g., supervised or unsupervised ones) and be more inclusive with respect to the different participant backgrounds and computing resources.

We implemented our track idea during our satellite challenge at DI2KG 2020, designing multiple tracks based on the following questions: *(i)* do you use supervised machine learning? *(ii)* do you rely on domain-specific knowledge (e.g., catalogs, thesauri, predefined patterns)? *(iii)* do you need human-in-the-loop (e.g., human oracles or crowdsourcing)?

3. SOLUTION HIGHLIGHTS

While most finalist solutions in the two ER contests shared several common aspects, they also presented an interesting variety of approaches. Differently from many solutions in literature, the best approaches were all optimized for the provided datasets. In general, much importance was given to the pre-processing operations, usually considering only few attributes deemed as useful (in fact, a real schema mapping was rarely performed), and the extraction of the relevant features (e.g., brand and model), basically carried out relying on regular expressions and human-designed rules or structures (e.g., lists of brands, dictionaries of aliases, etc.) used to inject domain knowledge. In some cases, this knowledge was acquired from Wikipedia or knowledge graphs and also semantics was employed by one solution, exploiting the skip-gram model to generate word embeddings. A blocking step was present in many solutions. Despite usually relying on basic functions in many cases this task played a fundamental role concurring to determine the matches (e.g., by grouping the products according to brand and model). The matching step reflected in most solutions a rule-based approach. In 2021 machine learning achieved a certain relevance, considering for example binary random forest classifiers to perform matching or combining XGBoost [3] with a rule-based matcher to solve the uncertainties of the latter. In some cases, the participants devised brand-specific models and relied on their own tools [17] to write labeling functions.

The blocking contest saw the design of more general and literature-based solutions. Of course, preprocessing still played a central role, including formatting and standardization, tokenization, and the resolution of inter- and intra-language synonyms. Regular expressions were still used in several cases to extract the key features. Then, different techniques were adopted to carry out the blocking itself, including sorted neighborhood, similarity joins [7, 8], sentence encoding using BERT [10], a neural architecture based on a distilled transformer [15], and exploiting additional training data [16] to perform supervised contrastive learning [14]. These techniques were often followed by a pair/block cleaning and ranking step (based on intra-pair similarity) to comply with the submission structure.

4. CONCLUSIONS

Running three SIGMOD programming contests has been an incredible opportunity. Setting up each contest required technical effort to adapt the preexisting webapp (4k Lines of Code) and implement the evaluation server (5k LOC). After set-up, the contest duration required reasonable organizational work. We are impressed by the diversity of high-performance solutions that were submitted during both the contests and the satellite events. This observation suggests that, despite having been studied by the data management community for long time, entity resolution may be still far from being solved, especially when targeting performance on specific datasets as it is expected in real-world applications.

REFERENCES

[1] J. Ahn, C. Seo, R. Mayuram, R. Yaseen, J. Kim, and S. Maeng. ForestDB: A Fast Key-Value Storage System for Variable-Length String Keys. *IEEE Transactions on Computers*, 65(3):902–915, 2016.

[2] M. Blacher, J. Klaus, M. Mitterreiter, J. Giesen, and S. Laue. Fast Entity Resolution With Mock Labels and Sorted Integer Sets. In *DI2KG@VLDB 2020*, volume 2726 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2020.

[3] T. Chen and C. Guestrin. XGBoost: A Scalable Tree Boosting System. In *KDD 2016*, pages 785–794. ACM, 2016.

[4] F. Chirigati, R. Rampin, D. Shasha, and J. Freire. ReproZip: Computational Reproducibility With Ease. In *SIGMOD 2016*, pages 2085–2088. ACM, 2016.

[5] V. Christophides, V. Efthymiou, T. Palpanas, G. Papadakis, and K. Stefanidis. An Overview of End-to-End Entity Resolution for Big Data. *ACM Computing Surveys*, 53(6):127:1–127:42, 2021.

[6] V. Crescenzi, A. De Angelis, D. Firmani, M. Mazzei, P. Merialdo, F. Piai, and D. Srivastava. Alaska: A Flexible Benchmark for Data Integration Tasks. *arXiv:2101.11259*, 2021.

[7] D. Deng, G. Li, H. Wen, and J. Feng. An Efficient Partition Based Method for Exact Set Similarity Joins. *PVLDB*, 9(4):360–371, 2015.

[8] D. Deng, Y. Tao, and G. Li. Overlap Set Similarity Joins with Theoretical Guarantees. In *SIGMOD 2018*, pages 905–920. ACM, 2018.

[9] N. Deng, W. Luan, H. Liu, and B. Tang. CheetahER: A Fast Entity Resolution System for Heterogeneous Camera Data. In *DI2KG@VLDB 2020*, volume 2726 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2020.

[10] J. Devlin, M. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL-HLT 2019*, volume 1, pages 4171–4186. ACL, 2019.

[11] A. Doan, P. Konda, P. Suganthan G. C., Y. Govind, D. Paulsen, K. Chandrasekhar, P. Martinkus, and M. Christie. Magellan: Toward Building Ecosystems of Entity Matching Solutions. *Communications of the ACM*, 63(8):83–91, 2020.

[12] L. Gagliardelli, G. Simonini, D. Beneventano, and S. Bergamaschi. SparkER: Scaling Entity Resolution in Spark. In *EDBT 2019*, pages 602–605. OpenProceedings.org, 2019.

[13] M. Graf, L. Laskowski, F. Papsdorf, F. Sold, R. Gremmelspacher, F. Naumann, and F. Panse. Frost: A Platform for Benchmarking and Exploring Data Matching Results. *PVLDB*, 15(12):3292–3305, 2022.

[14] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan. Supervised Contrastive Learning. In *NeurIPS 2020*, volume 33 of *Advances in Neural Information Processing Systems*, pages 18661–18673. Curran Associates, Inc., 2020.

[15] S. Mukherjee, A. H. Awadallah, and J. Gao. XtremeDistilTransformers: Task Transfer for Task-agnostic Distillation. *arXiv:2106.04563*, 2021.

[16] A. Primpeli, R. Peeters, and C. Bizer. The WDC Training Dataset and Gold Standard for Large-Scale Product Matching. In *EC-NLP@WWW 2019*, WWW (Companion Volume), pages 381–386. ACM, 2019.

[17] R. Wu, P. Sakala, P. Li, X. Chu, and Y. He. Demonstration of Panda: A Weakly Supervised Entity Matching System. *PVLDB*, 14(12):2735–2738, 2021.

[18] L. Zecchini, G. Simonini, and S. Bergamaschi. Entity Resolution on Camera Records without Machine Learning. In *DI2KG@VLDB 2020*, volume 2726 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2020.

ACKNOWLEDGEMENTS

We are deeply grateful to Siria Sarmiento for data curation work, to Alberto Corni for his technical support, to the Snowman team, to the sponsors, and to the SIGMOD chairs and previous contest organizers for the helpful discussion and support.