Combining Data-driven Supervision with Human-in-the-loop Feedback for Entity Resolution

Wenpeng Yin, Shelby Heinecke, Jia Li
Nitish Shirish Keskar, Michael Jones, Shouzhong Shi
Stanislav Georgiev, Kurt Milich, Joseph Esposito, Caiming Xiong
Salesforce
{wyin,shelby.heinecke,jia.li}@salesforce.com

Abstract

The distribution gap between training datasets and data encountered in production is well acknowledged [2, 12, 17]. Training datasets are often constructed over a fixed period of time and by carefully curating the data to be labeled. Thus, training datasets may not contain all possible variations of data that could be encountered in real-world production environments. Tasked with building an entity resolution system - a model that identifies and consolidates data points that represent the same person - our first model exhibited a clear training-production performance gap. In this case study, we discuss our human-in-the-loop enabled, data-centric solution to closing the training-production performance divergence. We conclude with takeaways that apply to data-centric learning at large.

1 Introduction

The distribution discrepancy between training datasets and data encountered in production is widely known [2, 12, 17]. A common workflow for collecting training datasets is to curate data over a fixed period of time and then label them using annotators. This limits the number and kinds of variations that are present in a training dataset. Models relying only on training datasets may lead to underwhelming performance in production when confronted with perturbed or out-of-distribution samples. This training-production performance gap has been addressed more generally from several closely related perspectives including out-of-distribution learning [16], model robustness [8], and data augmentation [7]. Confronted with the training-production performance gap in a business setting, we found that an application-specific approach was needed.

We were tasked with building an entity resolution system - a model that identifies and consolidates data points that represent the same person [3]. For example, the model intends to recognize that a `Jane Doe` with address `123 Main St, San Francisco, CA` is the same person as `J. V. Doe` living on `123 Main St.` in Zipcode `94158`. Circumstantially, more information such as phone number or email address might be available too. Our model would be deployed in production and used as an automated data pre-processing step to potentially benefit many downstream analytics and modelling tasks. *For this initial development phase, our model intends to classify identities based on one field - an individual's name and leaves the incorporation of other contextual fields in future phases*. For training data, we were given a highly curated set of labeled name pairs consisting of examples such as `(''Joe'', ''Joseph'', match)` or `(''Joe'', ''Jane'', mismatch)`. For evaluation, we had a diverse set of *unlabeled* name pairs that included new instances of real-world name variations. Our first model exhibited the training-production performance gap, inspiring us to create a human-in-the-loop enabled, data-centric approach for this application.

In this case study, we discuss the details of our approach, a mix of data augmentation and data-derived rules, and the key takeaways applicable to data-centric learning more generally.

35th Conference on Neural Information Processing Systems (NeurIPS 2021), Sydney, Australia.

2 Challenges of Entity Resolution

Across documents and records, the name of an individual can have different surface forms including capitalization, spellings, accents, ordering and others. We summarize some of the complex variances below. (a) *Missing spaces/hyphens*: "`Mary Ellen`", "`MaryEllen`", and "`Mary-Ellen`" may represent the same person; (b) *Initials*: a name, such as "`James Earl Smith`", could also be represented with initials such as "`JE Smith`", "`J.E. Smith`", "`JE. Smith`" or "`J.E Smith`"; (c) *Out-of-order components*: such as "`Diaz Carlos Alfonzo`" vs. "`Carlos Alfonzo Diaz`"; (d) *Truncated name*: such as "`Livingston Charles`" vs. "`Living Charles`"; (e) *Nicknames*: a name's nickname is often a shorter string (e.g., "`Mike`" vs. "`Michael`") or can even be totally different by metaphone (e.g., "`William`" vs. "`Bill`"); (f) *Multilingual*: the users from different language backgrounds are allowed to type their names in native languages so that our system is expected to match multilingual names. For example, "`Schütze`" (German) matches with "`Schuetze`" (English); (g) *Junk words*: when the system requires a user to type in her or his name, the user sometimes just puts a random string as the name, such as "`too`", "`API`", "`such`", etc; (h) *Gender mismatch*: we should clearly reject the match of two names if they are typically used in different genders, such as "`Daniel`" (M) vs. "`Daniela`" (F).

3 Training Dataset

To provide supervision for developing a system, we manually annotated some data in different languages, including English, Arabic, French, German, Russian, Spanish and Portuguese. The data consisted of the most popular names within our consumer data. Also included were the most popular names by country (both male and female), as well as most popular nicknames. We then created multiple permutations of these names to annotate different matching patterns. The detailed statistics are presented in Table 1.

Table 1: Statistics of clean labeled data in different languages.

	English	French	German	Arabic	Russian	Spanish	Portuguese
`#match`	3,037	396	175	845	510	1,023	659
`#mismatch`	129	396	501	82	34	601	60

More matched pairs were included than mismatched pairs. To generate more mismatched instances, we used rules, such as deletion, insertion, and n-gram similarity, to collect about 20,000 mismatched pairs automatically. This labeled data is randomly split into *train* (70%), *dev* (20%) and *test* (10%).

4 Baseline Approach

The baseline is a system already in production. It is based on name transformation given predefined patterns: (i) *Longest common substring pattern*, e.g., ("`Jörn`" vs. "`Jorn`") → ("`jö2`", "`jo2`"), where digit "2" indicates the length of the common substring; (ii) *Position unigrams metaphone pattern*, e.g., ("`aab`" vs. "`abbc`") → ("`a1`" vs. "`b1c3`"). So, given the input name pair ("`aab`", "`abbc`"), the following transformed pairs will be created: `[(''aab'', ''abbc''), (''a2'', ''2bc''), (''a1'', ''b1c3'')]`. Then, in the training phase, each transformed pair will be mapped to the frequencies of belonging to the `match` and `mismatch` classes. For a test name pair, apply the same transformation extraction, and then feed into model, use the feature with highest probability to decide whether the input is a `match` or not.

Our goal was to build a model that outperforms this baseline on the held-out test set and real consumer data. Next, we describe our human-in-the-loop enabled process for a more advanced system.

5 System Development with Human-in-the-loop Evaluation

Our system was developed with the help of multiple rounds of human evaluations and feedback. We summarize the four versions in the following subsections.

5.1 V1: "pure deep learning system"

System. For our first version of system, we treat this name matching problem the same way as popular sentence matching tasks [18], such as paraphrase identification [6], textual entailment [4], etc. To be specific, we make use of the pretrained BERT model [5] [1] to take two names as the input and conduct binary classification (i.e., `match` vs. `mismatch`). The probability corresponding to the `match` class is used as the matching score of two input names. Table 2 lists the results of our system V1 and the baseline. Due to the clear improvement on overall recall, our system outperforms the baseline with a large margin (F1: 95.07 vs. 70.46).

Table 2: Comparison our systems (V1 and V4) with the baseline.

	Individual							Overall		
	English	French	German	Arabic	Russian	Spanish	Portuguese	baseline	V1	V4
R	94.71	99.87	85.96	95.91	93.30	90.56	92.90	56.00	94.55	**98.48**
P	95.44	100.0	100.0	97.40	95.46	94.76	99.10	95.00	95.60	**99.12**
F1	95.07	99.93	92.45	96.65	94.37	92.61	95.90	70.46	95.07	**98.79**

Human evaluation and feedback. As a part of the delivery pipeline, the V1 system was delivered to our engineering partners. Using approximately 2 million unlabeled name pairs of real customer data, the team checked for disagreements between the baseline system (Section 4) and our proposed model. Those instances were then broken up based on their 10th percentiles in the predicted matching score of V1, randomly sorted those grouped percentiles and manually reviewed and labeled 5% of the instances for each percentile group, which accounted for 8k in total. Despite the superiority of our V1 model in the clean data, the system exhibited unexpected error patterns: (a) *Single letter/period match*, such as ("`.`" vs. "`jeff`"), ("`A`" vs. "`Edward`"); (b) *Initial matches*, such as ("`A.J.`" vs. "`Deby`"), ("`T.J.`" vs. "`John Paul`"); (c) *Common nicknames did not match*, e.g., ("`Susan`" vs. "`Susanna`"), ("`John`" vs. "`Johnnie`"); (d) *Junk words match*, such as ("`Accounting`" vs. "`Debby`"), ("`Financial`" vs. "`Fay`"); (e) *Double name with double name*, such as ("`Juan Manuel`" vs. "`Jean Michel`").

We noticed that the real data contains instances that are much more diverse, such as the challenges discussed in Section 2, than that in the clean data. Many error patterns do not have corresponding labeled data that can provide supervision, e.g., nickname recognition.

5.2 V2: deep learning with data augmentation

System. To deal with the issues of V1, we try two threads: first, we conduct data augmentation specific to some error patterns, such as "double name with double name", "Single letter/period match", and "Initial matches"; second, we build our lists to cover junk words and nicknames, respectively. Specifically, our augmented data has 10k labeled examples for each error pattern; our junk work list has size 1,570, covering some stop words and some words that are clearly not used as person names; the nickname list consists of 3,458 name pairs in total. The augmented data is used to train the BERT model; the junk word list and the nickname list are used to determine the predictions: `match` if two names are in the nickname list, `mismatch` if one name is a junk word and the other is not equal to it.

Human evaluation and feedback. The evaluation team used the same data as in V1, and the same way to sample pairs for human evaluation. The following new issues were reported: (a) *Inconsistency*. Overall, our V2 system performed better than V1 for those patterns, this means our deep learning model (i.e., BERT) indeed learned supervision to some extent, but inconsistently so. This is mainly related to patterns that were supposed to be solved by data augmentation, such as "Initial matches", "Single letter/period match", etc. As an example: "`Maria`" is found to match with its initial "`M`" but "`Dale`" does not match with "`D.`". And some false positive cases were found: "`Service Order`" matches with "`S.`"; (b) *Gender mismatches*. For example, in V2 we found "`Daniel`" matches with "`Daniela`" and "`Danielle`" (both the latter are feminine). This means the training data cannot provide enough signal about name genders; (c) *Lacking the Metaphone information*. For example, the

[1] Initialized by the "distilbert-base-multilingual-cased" model.

system can neither recognize that "Christian" matches with "Kristian", nor "Pavel" matches with "Pawel".

5.3 V3

System. To deal with the issues found for V2, we adopt the following solutions. (i) For the inconsistency issue, we rely on rules and prioritize rule-based decision over data-base learning (take the data-based decision only if the rules cannot handle it); (ii) For the gender mismatch, we first collect (name, gender) mapping from the Social Security website; [2] and (iii) We detect the metaphone[3] of names and compute name similarity based on that. As a part of the human evaluation and feedback, the following error patterns were provided: (a) *Exact names do not match*. Our V3 system predicts "mismatch" for the pairs (".", ".") and ("will", "Will") because V3 will detect them as junk words. However, the production team prefers to claim them match although they are not regular names; (b) *False positive from machine learning predictions*. The system V3 tends to predict match for pairs ("Cristina" vs. "Christian"), ("Marisa" vs. "Maria"), etc., which were unable to be handled by rules.

5.4 V4: the final system

Our final system, i.e., V4, is derived by utilizing some simple while effective tricks: we use rules to handle the "Exact names do not match"; for "False positive from machine learning predictions", those pairs were unable to be handled by rules, therefore the matching score fully comes from the BERT classifier. In the past versions, we used a threshold 0.5 to decide if an input pair is "match" or not. Based on the error cases, we tighten the threshold and fix it to 0.7. After the four rounds of iteration, our system V4 gets further improvement over the V1 (98.79 vs. 95.07 by F1 in Table 2).

Remaining debatable issues. (a) *One name as the sub-name of the other*, such as ("Jose M" vs. "M") and ("Jose Maria" vs. "Maria"). Some researchers think the first pair should be mismatch while the second one be match; (b) *Gender first or sub-name first?* Considering the pair ("Jose Maria" vs. "Maria") again, "Jose Maria" is often used as a Spanish male first name, in this case, the prediction should be mismatch since Maria is a common female name; however, if we want to emphasize more on the sub-name pattern, the prediction should be match.

6 Related Work

Most prior works, such as [1, 10, 9, 13, 15, 14, 11], tried to collect adversarial training data by human annotators to address the error patterns found in human-in-the-loop. We tried too but finally gave up due to two reasons: (i) collecting large-scale adversarial data is even more costly than enlarging the training data with data augmentation; (ii) it performed worse than rules for some error patterns in which a simple rule can get high precision, such as gender-aware predictions.

7 Discussion

In this case study, we shared our human-in-the-loop-enabled, data-centric approach for building a production quality entity resolution model. In contrast to some research that work on clean labeled data and claim state of the art performance, we are trying to handle a real-world problem which inspires us to rethink the following questions:

What is data? By "data", the community usually refer to labeled or unlabeled data. Here, we advocate that rules are also a sort of precise expression of data. More specifically, If the issue can be exactly expressed by concise rules, that means rules are actually the objective status of model training on data. Therefore, we think "data" should be extended as "data+rules".

How much data is needed? Labeled data is always limited. However, the challenges in the real world are often beyond the imagination. Data-centric learning can play a bigger role once combined with the human-in-the-loop interactions.

[2] https://www.ssa.gov/oact/babynames/limits.html

[3] By "org.apache.commons.codec.language.Metaphone".

References

[1] Max Bartolo, Alastair Roberts, Johannes Welbl, Sebastian Riedel, and Pontus Stenetorp. Beat the AI: investigating adversarial human annotation for reading comprehension. *Transactions of ACL*, 8:662–678, 2020.

[2] Yoshua Bengio, Tristan Deleu, Nasim Rahaman, Nan Rosemary Ke, Sébastien Lachapelle, Olexa Bilaniuk, Anirudh Goyal, and Christopher J. Pal. A meta-transfer objective for learning to disentangle causal mechanisms. In *Proceedings of ICLR*, 2020.

[3] Vassilis Christophides, Vasilis Efthymiou, Themis Palpanas, George Papadakis, and Kostas Stefanidis. An overview of end-to-end entity resolution for big data. *ACM Computing Surveys*, 53(6):127:1–127:42, 2021.

[4] Ido Dagan, Oren Glickman, and Bernardo Magnini. The PASCAL recognising textual entailment challenge. In *Machine Learning Challenges, Evaluating Predictive Uncertainty, Visual Object Classification and Recognizing Textual Entailment, First PASCAL Machine Learning Challenges Workshop, MLCW*, volume 3944, pages 177–190, 2005.

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of NAACL-HLT*, pages 4171–4186, 2019.

[6] Bill Dolan, Chris Quirk, and Chris Brockett. Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources. In *Proceedings of COLING*, pages 350 – 356, 2004.

[7] Karan Goel, Albert Gu, Yixuan Li, and Christopher Ré. Model patching: Closing the subgroup performance gap with data augmentation. In *Proceedings of ICLR*, 2021.

[8] Karan Goel, Nazneen Fatema Rajani, Jesse Vig, Zachary Taschdjian, Mohit Bansal, and Christopher Ré. Robustness gym: Unifying the NLP evaluation landscape. In *Proceedings of NAACL-HLT*, pages 42–55, 2021.

[9] Divyansh Kaushik, Douwe Kiela, Zachary C. Lipton, and Wen-tau Yih. On the efficacy of adversarial data collection for question answering: Results from a large-scale randomized study. In *Proceedings of ACL/IJCNLP*, pages 6618–6633, 2021.

[10] Douwe Kiela, Max Bartolo, Yixin Nie, Divyansh Kaushik, Atticus Geiger, Zhengxuan Wu, Bertie Vidgen, Grusha Prasad, Amanpreet Singh, Pratik Ringshia, Zhiyi Ma, Tristan Thrush, Sebastian Riedel, Zeerak Waseem, Pontus Stenetorp, Robin Jia, Mohit Bansal, Christopher Potts, and Adina Williams. Dynabench: Rethinking benchmarking in NLP. In *Proceedings of NAACL-HLT*, pages 4110–4124, 2021.

[11] Hannah Rose Kirk, Bertram Vidgen, Paul Röttger, Tristan Thrush, and Scott A. Hale. Hatemoji: A test suite and adversarially-generated dataset for benchmarking and detecting emoji-based hate. *CoRR*, abs/2108.05921, 2021.

[12] Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, Tony Lee, Etienne David, Ian Stavness, Wei Guo, Berton Earnshaw, Imran Haque, Sara M. Beery, Jure Leskovec, Anshul Kundaje, Emma Pierson, Sergey Levine, Chelsea Finn, and Percy Liang. WILDS: A benchmark of in-the-wild distribution shifts. In *Proceedings of ICML*, volume 139, pages 5637–5664, 2021.

[13] Christopher Potts, Zhengxuan Wu, Atticus Geiger, and Douwe Kiela. Dynasent: A dynamic benchmark for sentiment analysis. In *Proceedings of ACL/IJCNLP*, pages 2388–2404, 2021.

[14] Sasha Sheng, Amanpreet Singh, Vedanuj Goswami, Jose Alberto Lopez Magana, Wojciech Galuba, Devi Parikh, and Douwe Kiela. Human-adversarial visual question answering. *CoRR*, abs/2106.02280, 2021.

[15] Bertie Vidgen, Tristan Thrush, Zeerak Waseem, and Douwe Kiela. Learning from the worst: Dynamically generated datasets to improve online hate detection. In *Proceedings of ACL/IJCNLP*, pages 1667–1682, 2021.

[16] Jindong Wang, Cuiling Lan, Chang Liu, Yidong Ouyang, and Tao Qin. Generalizing to unseen domains: A survey on domain generalization. In *Proceedings of IJCAI*, pages 4627–4635, 2021.

[17] Mei Wang and Weihong Deng. Deep visual domain adaptation: A survey. *Neurocomputing*, 312:135–153, 2018.

[18] Wenpeng Yin, Hinrich Schütze, Bing Xiang, and Bowen Zhou. ABCNN: attention-based convolutional neural network for modeling sentence pairs. *Transactions of ACL*, 4:259–272, 2016.