Literature Review

Schema Matching using Pre-Trained Language

Ziyin Li, Yuhai Liu, Tianyin Fan

Introduction to Schema Matching Challenges:

Identified limitations in current schema-level approaches.

- **Limited Access to Customer Data Records:**
  - **Privacy Constraints:** Customers restrict access to individual data records, hindering data utilization in schema matching.
  - **Schema Information Access:** Customers provide schema details but not data records, impacting accuracy.
- **Disparity in Schema Sizes:**
  - **Large ISS vs. Smaller Customer Schema:** Varying sizes complicate matching, resulting in numerous potential matches for each attribute.
- **Complex Naming Conventions:**
  - **Terminology Complexity:** Abbreviations and industry-specific terms in customer schemas hinder automation.
  - **Automation Challenges:** Complex naming conventions pose difficulties in automating schema matching.
- **Challenges with Real Customer Schemata:**
  - **External Constraints:** Customer-imposed restrictions limit accuracy in mapping customer schemas to known ISSs.
  - **Accuracy Impact:** These limitations affect accuracy, especially when mapping without full data access.

These limitations highlight the challenges faced by existing schema-level approaches, emphasizing the need for innovative methods like the Learned Schema Matcher (LSM) to improve accuracy without accessing individual data records.

Introduction to Schema Matching Challenges:

Poor accuracy in existing methods

- This paper choose four heuristic-based representative schema-based matching methods: cupid, coma, s-match and similarity flooding.

	CUPID	COMA	SM	SF	LSD	MLM
RDB-Star	0.96	**1.00**	**1.00**	0.70	0.26	**1.00**
IPFQR	**1.00**	0.98	0.82	**1.00**	0.08	0.98
MovieLens-IMDB	0.64	0.54	**0.72**	0.71	0.00	0.64
Customer A	0.18	**0.21**	0.07	0.11	0.00	0.11
Customer B	**0.14**	0.02	0.06	0.02	0.00	0.08
Customer C	0.08	0.22	0.11	**0.23**	0.00	0.14
Customer D	0.27	**0.31**	0.13	0.14	0.00	0.14
Customer E	0.27	0.17	**0.28**	0.12	0.00	0.14

LSM incorporates active learning and an intelligent attribute selection strategy to obtain precise user feedback, thereby reducing human labelling costs. Experimental results demonstrate LSM's effectiveness, as it outperforms the best baseline approach by requiring fewer labels and reducing reviewing costs.

Motivation

- **Need for a New Approach**

- The paper assesses current schema matching methods reliant on schema-level information due to inaccessible customer data. These methods exhibit low accuracy in matching real customer schemata, highlighting the inadequacy when dealing with variations in attribute naming conventions and larger target schema sizes. Traditional methods perform well on synthetic and public schemata but poorly on real-world data. This gap in accuracy necessitates a new approach combining accurate models with human-in-the-loop intervention and leveraging pre-trained language models' natural language understanding capabilities. This insight drives the design of a novel linguistic schema matching approach, combining semi-supervised and active learning techniques to address these challenges.

LSM Process

- **Preparation:** Generates candidate pairs by computing the Cartesian product between attribute sets in the source and target schemas.
- **Featurization:** Converts candidate pairs into numerical vectors using various featurizers like word embedding, lexical, and a fine-tuned BERT featurizer, leveraging a pre-trained language model to capture linguistic similarities.
- **Model Training and Prediction:** Trains a meta-learning model using partially labeled data to predict matching labels for unlabeled attributes. The model's outputs are tuned based on data type compatibility and penalization for introducing new entities.
- **User Interaction:** Involves providing matching suggestions to users for unlabeled attributes. Users review and mark correct matches or label incorrectly matched attributes selected by LSM. LSM prioritizes labeling using **a least confident anchor strategy** for the most "informative" attributes in the schema. (details in next page)
- **Updating Labels:** User feedback updates candidate pair labels for the subsequent iteration.

Process In Detail: Anchor Strategy

Here's a breakdown:

a.Anchor Attributes: These are considered the most "informative" attributes in the schema. In LSM, the anchor set usually consists of primary keys (PKs) and foreign keys (FKs) as they carry crucial information regarding schema relationships.

b.Least Confidence Selection: Among the unlabeled anchor attributes, the LSM system selects a subset of N attributes to be labeled by employing the "least confidence" strategy. This strategy determines the N attributes with the least prediction confidence based on the Softmax function applied to the matching scores.

c.Calculation of Prediction Confidence: The prediction confidence (cs) for each unlabeled anchor attribute is calculated using Softmax on the matching scores generated by the model. This score represents how uncertain or confident the system is about the correctness of a suggested match for a particular attribute.

d.User Labeling: The system presents the selected subset of attributes to the user and asks for correct mappings to the target schema. Users then provide these mappings. For each correct match (as, at), the label for that pair is marked as 1, while other potential matches (as, a′t) are marked as -1.

•

Experimental Evaluation

Goals mainly want to answer the following questions:

1. What is the prediction quality of LSM model? (Section V-B)

2. What is the human labeling cost to map the full source schema to the ISS? (Section V-C)

3. What is the contribution of the BERT featurizer and the attribute descriptions to the overall performance? (Section V-D)

4. How is the performance of LSM affected in the presence of noise? (Section V-F)

5. How much time is spent re-training the model after the user provides new labels? (Section V-G).

Experimental Conclusion

1. Prediction quality:

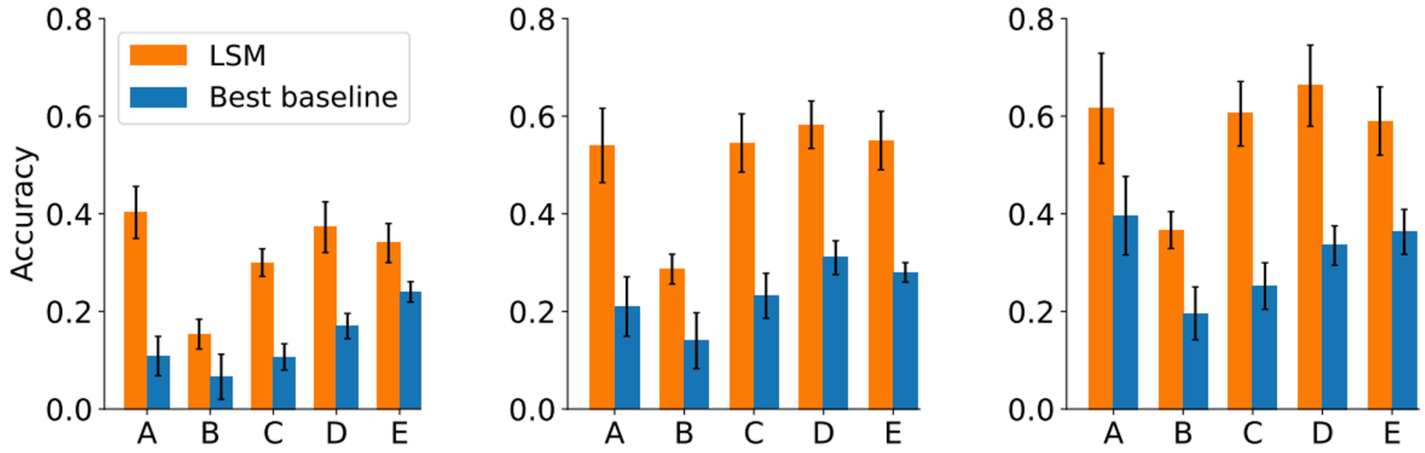
In the non-interactive experiment tested on Public Schemata and Customer Schemata, the result of LSM accurate rates is better than the baseline on different database (RDB-Star, IPFQR, MovieLens-IMDB) and on the customer's schema (Table IV).

Table IV: Top-k accuracy of LSM on the public schemata.

	Best Baseline			LSM		
	Top-1	Top-3	Top-5	Top-1	Top-3	Top-5
RDB-Star	0.95	1.00	1.00	0.98	0.98	1.00
IPFQR	1.00	1.00	1.00	1.00	1.00	1.00
MovieLens-IMDB	0.53	0.72	0.75	0.65	0.83	0.87

An example of Prediction Quality

the figure illustrate that observation results indicate that for all modes, LSM outperforms the optimal baseline in terms of Top-1, Top-3, and Top-5 accuracy. In addition, as the proportion of labels increases, the accuracy of both methods shows an upward trend, but the increase in LSM is greater, further widening the gap with the optimal baseline. LSM can achieve high matching accuracy when only a small amount of manual labels are used. In contrast, the best baseline method performs poorly when dealing with these patterns, requiring more labels to achieve similar accuracy. Therefore, we can conclude that LSM outperforms the best baseline method in terms of Top-k accuracy in customer mode A-E.



(a) Top-1 (b) Top-3 (c) Top-5

Fig. 4: Top-k accuracy of LSM vs the best base line on customer schemata A-E

Experimental Conclusion

2. Human labeling cost:

Correctly match around 70% of the customer schema attributes, and using the LSM method smart selection strategy, our model outperforms not only the best baseline but also the random selection strategy by reducing the total labels required by up to 11%

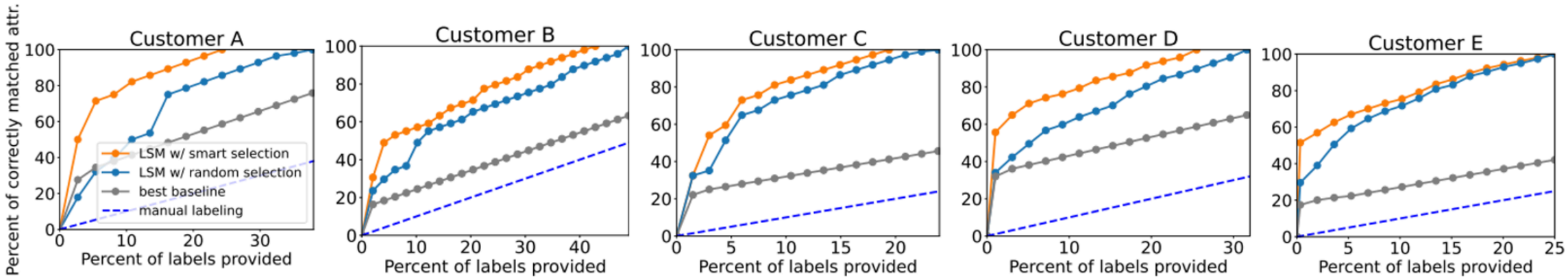


Fig. 5: Percentage of the attributes correctly matched vs. percentage of human labels provided.

Experimental Conclusion

3. BERT featurizer contribute overall performance:

the results of ablation experiments on BERT feature extractors on different customer modes to see the correct matching percentage for each attribute description. The experimental results show that using the BERT feature extractor can significantly improve the accuracy of attribute matching. In all customer patterns, using the BERT feature extractor improved the correct matching rate by an average of 11%. This indicates that the BERT feature extractor is a key component for our model. So, in this paper, the BERT featurizer is a critical component of LSM model. Inthe figure, disable the BERT featurizer in our model (denoted as LSM w/o BERT), the user may need to provide up to 17% more labels (Customer B) to map their full schema to ISS.

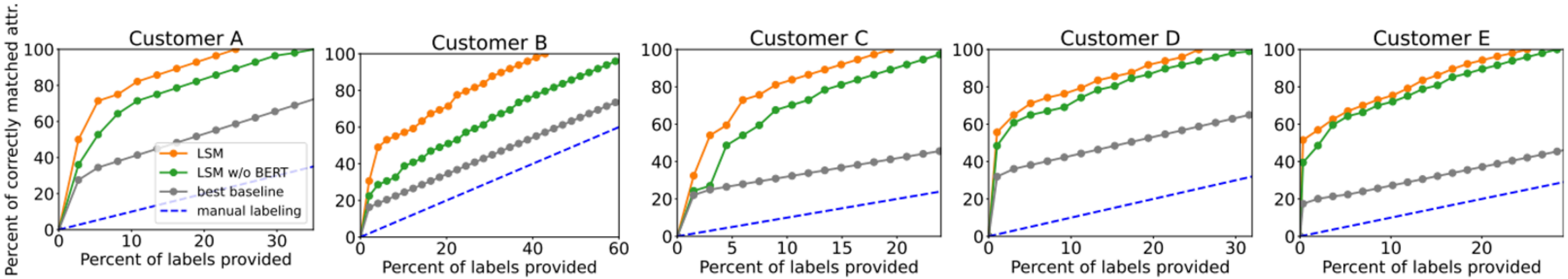


Fig. 6: Ablation study on the BERT Featurizer using various customer schemata.

Experimental Conclusion

4. LSM performance affect by noise

Figure 8 shows the performance of LSM under different noise rates. Specifically, Figure 8 shows the percentage of LSM correctly matching attributes when the noise rates are 0 (raw LSM), 0.1, 0.2, and 0.3. From the graph, it can be observed that as the noise rate increases, the number of attributes correctly matched by LSM decreases. Specifically, when the noise rate is 0.1, the number of correctly matched attributes in LSM decreases to 90%. When the noise rate is 0.2, the number of correctly matched attributes decreases to 80%. When the noise rate is 0.3, the number of correctly matched attributes decreases to 70%. These data points are represented by dashed lines. However, even at a noise rate of 0.3, LSM still outperforms the best baseline model (represented by a solid line). It should be noted that these results were obtained using the BERT Featurizer.

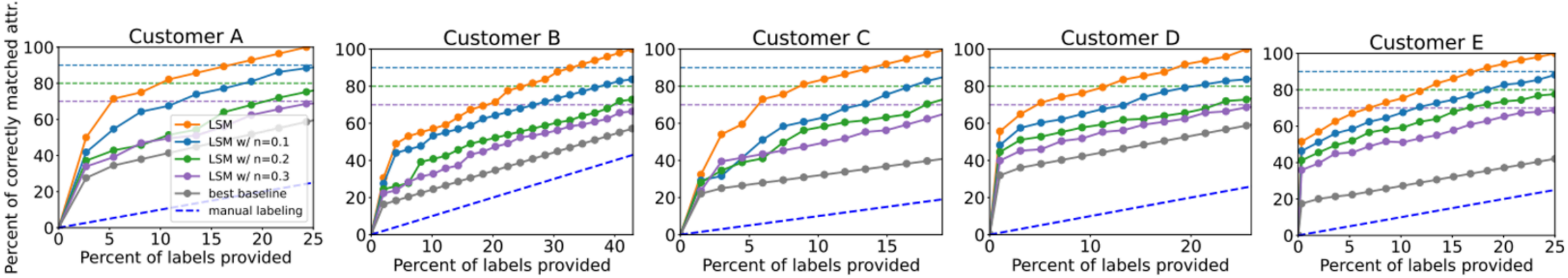


Fig. 8: Performance of LSM in the presence of noise with varying noise rates n.

Experimental Conclusion

5. LSM re-training response time

Figure 9 shows the response time of LSM as the number of matching labels varies. The experiment was run on an Azure cloud server with 8 core CPUs, 112 GiB of memory, and a Tesla P100. In Figure 9, we can see that as the number of tags increases, the response time also increases accordingly. However, it is worth noting that even with an increase in the number of tags, the response time of LSM is still relatively fast, and in most cases, the response time is less than 10 second. This indicates that our model has good performance and efficiency when processing large amounts of data. In addition, we also noticed that the response time is more affected by the number of candidate pairs than by the number of labels. This is because we used both labeled and unlabeled examples for training in semi supervised settings, so the number of candidate pairs directly affects the complexity of the training process. In summary, LSM has good response time and the ability to process large amounts of data.

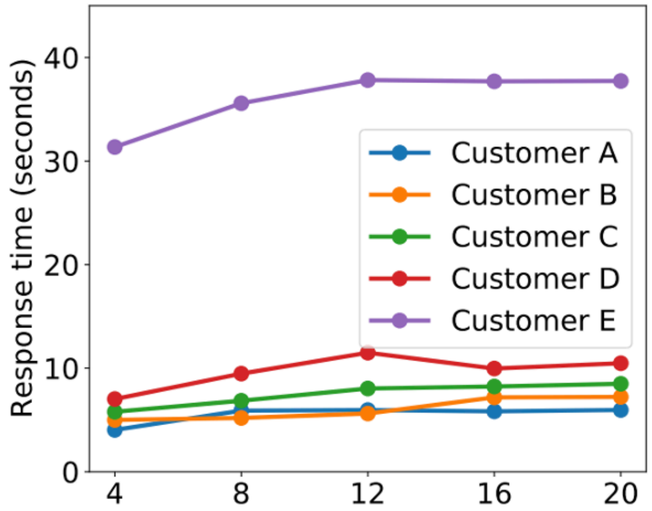


Fig. 9: Response time of the LSM as the number of labels is varied.

Discussion

Hyper-parameter Tuning: Discusses the challenge of tuning parameters tailored to schemas, emphasizing simplicity for usability and performance.

ML in Schema Matching: Addresses overfitting risks due to limited data, highlighting an active learning approach and a lightweight model. Pre-training a BERT featurizer aids domain understanding.

Response Time: Acknowledges challenges in achieving quick response times with large schemata. Proposed model offers efficient ranking but acknowledges increased response time with more input attributes, still considered acceptable compared to manual matching.

Design Space: Considers alternatives like large language models or domain-specific pre-training. Prefers fine-tuning a smaller model like BERT for now, keeping other options for future exploration.

Discussion

Limitations of the LSM

the LSM heavily relies on schema-only information for matching and does not address problems where the user has no control or understanding of their schema, such as when it is generated by a third-party application. Data access would likely be required to solve such problems.

Discussion

shortcomings in some aspects

1. The paper overlooks the semantic matching challenge between source and target data. It suggests integrating semantic understanding into pattern matching for improved data consistency. Although the original research employed a finely tuned BERT model, the Learned Schema Matcher could potentially enhance its universality by considering semantic insights about the ISS.

2. The proposed Learned Schema Matcher algorithm might struggle with intricate mapping transformations, leading to reduced performance. Future research could focus on enhancing algorithms to manage complex mappings more effectively.

Discussion

shortcomings in some aspects

3. The Least Confident Anchor method has limitations:

- Limited Context: It considers single attribute predictions, disregarding contextual relations among attributes, possibly missing the correct anchor attribute.
- Data Quality Dependency: Relies on high-quality data; noise or errors can affect its accuracy in selecting anchor attributes.
- Handling Unknowns: Primarily focuses on known attributes, potentially struggling when dealing with unknown cases or attributes, leading to suboptimal outcomes.

Looking ahead

leveraging pre-trained language models like BERT [2] can enhance the LSM method in several ways:

- Rapid Adaptation: Pre-trained models can swiftly adapt to LSM tasks, leveraging prior language knowledge.
- Enhanced Generalization: They can generalize across tasks and domains, aided by data augmentation for better performance.
- Reduced Data Needs: Leveraging existing language knowledge allows fine-tuning on smaller datasets, reducing data requirements.
- Improved Performance: Fine-tuning enhances model performance and speeds up training, making schema matching more efficient.

Continuous algorithm optimization in LSM applications promises more efficient automated schema matching tools.

Reference

[1] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” arXiv preprint arXiv:1810.04805, 2018.

[2] Yuliang Li, Jinfeng Li, Yoshihiko Suhara, and etc. “Deep Entity Matching with Pre-Trained Language Models”, arXiv preprint arXiv: 2004.00584, 2020.