ICMR '20, October 26–29, 2020, Dublin, Ireland
Proceedings published June 8, 2020

An Active Learning Framework for Duplicate Detection in SaaS Platforms

Quy H. Nguyen*
AISIA Research Lab
Ho Chi Minh, Vietnam

Dac Nguyen*
AISIA Research Lab
Ho Chi Minh, Vietnam

Minh-Son Dao
National Institute of Information and Communications Technology
Tokyo, Japan

Duc-Tien Dang-Nguyen†
Department of Information Science and Media Studies
University of Bergen
Bergen, Norway

Cathal Gurrin
Dublin City University
Dublin, Ireland

Binh T. Nguyen‡
AISIA Research Lab
VNU HCM - University of Science
Ho Chi Minh, Vietnam

ABSTRACT

With the rapid growth of users' data in SaaS (Software-as-a-service) platforms using micro-services, it becomes essential to detect duplicated entities for ensuring the integrity and consistency of data in many companies and businesses (primarily multinational corporations). Due to the large volume of databases today, the expected duplicate detection algorithms need to be not only accurate but also practical, which means that it can release the detection results as fast as possible for a given request. Among existing algorithms for the deduplicate detection problem, using Siamese neural networks with the triplet loss has become one of the robust ways to measure the similarity of two entities (texts, paragraphs, or documents) for identifying all possible duplicated items. In this paper, we first propose a practical framework for building a duplicate detection system in a SaaS platform. Second, we present a new active learning schema for training and updating duplicate detection algorithms. In this schema, we not only allow the crowd to provide more annotated data for enhancing the chosen learning model but also use the Siamese neural networks as well as the triplet loss to construct an efficient model for the problem. Finally, we design a user interface of our proposed deduplicate detection system, which can easily apply for empirical applications in different companies.

CCS CONCEPTS

• **Information systems** → **Recommender systems**; **Search interfaces**; **Expert systems**.

*Both authors contributed equally to this research.
†Senior author
‡Corresponding author

ICMR '20, June 8–11, 2020, Dublin, Ireland

ACM ISBN 978-1-4503-7087-5/20/06…$15.00
https://doi.org/10.1145/3372278.3391933

KEYWORDS

datasets, triplet loss, active learning, duplicate removal

ACM Reference Format:
Quy H. Nguyen, Dac Nguyen, Minh-Son Dao, Duc-Tien Dang-Nguyen, Cathal Gurrin, and Binh T. Nguyen. 2020. An Active Learning Framework for Duplicate Detection in SaaS Platforms. In *Proceedings of the 2020 International Conference on Multimedia Retrieval (ICMR '20), June 8–11, 2020, Dublin, Ireland*. ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3372278.3391933

1 INTRODUCTION

Nowadays, in a data-driven age, customer data have been a valuable weapon as well as a powerful shield for many businesses and corporations. In multiple domains (such as, e.g., the gaming industry, finance-related fields, the inspection industry, and micro/macro insurance), data holds a unique position in informing the company what is going on and how to improve ongoing systems to support better decision making. Consequently, many organizations recognize the need to build necessary data pipelines and data warehouses and invest in building data science teams for exploring new insights from customers' data and investigating potential applications related to their products. It turns out that data aggregation plays a crucial role in every data science project. For SaaS platforms using microservices, especially when working with a large number of customers, every integration mechanism can cause duplication. It may occur more and more frequently when the platform is not mature enough, which can create incorrect data distribution, drain storage resources, or deteriorate the quality of the decision-making processes.

Machine learning has recently played an essential part in many organizations due to the capability to learn patterns and mine valuable insights from users' data. Also, it can help people to automatically perform tasks without the need for human intervention, such as classification and prediction. There are several factors influencing data quality for training efficient machine learning models, including the number of observations or samples collected as well as the associated ground-truth data. Sometimes, the cost of investing valuable training and testing data in one machine learning project is too high to be viable. It may need significant manual efforts, request more labors and time-consuming affairs, and sometimes require the involvement of domain experts to verify results.

For instance, when working in the duplicate document detection problem in a SaaS platform, one team needs to collect a lot of texts, read the content of different documents, and then manually create training and testing datasets, including pairs of documents which are duplicate or not. As it may cost a lot of time and resources to generate a high-quality dataset for every machine learning problem, the data collection process sometimes stops earlier. It creates highly imbalanced datasets and a lot of challenges for data scientists.

Duplicate removal, by its nature, addresses a string metric that measures the level of similarity. Various studies [2, 4, 15] have used unsupervised learning methods incorporating different measurements, including Levenshtein, Jaro, and Jaro-Winkler distances. Those measurements are token-based, which may not distinguish well between two inputs that do not have a similar structure but have similar content, due to natural language characteristics. Meanwhile, Siamese neural networks [12] with the triplet loss can overcome the challenge because of its capability for learning abstract patterns on a higher dimension and indicating how similar two texts are in the inference step. It is worth noting that deep learning has recently surpassed traditional approaches in numerous domains. Thereby, we aim to apply a deep neural approach by using Siamese neural networks to address this this problem.

Updating machine learning models with new data is always essential to keep the stability of their performance and accuracy. Active learning [13] is one of the most potent techniques in machine learning to enable us to get the possibility of collecting more data with a cost-efficiency. It mimicks a practical learning process when one can receive new data frequently related to the problem and pay a potential cost to get the feedback from experts to create correct labels for new samples. In this paper, we present a new active learning schema for training and updating Siamese neural networks using the triplet loss (which needs to get positive, negative, and anchor samples for the training process) in the duplicate detection problem. For each iteration, our proposed system can interact with a human and ask the human to give it the correct label for each pair of entities requested. This inquiry is not arbitrary, but we expect that it can drive the performance of our proposed machine learning model when collecting more data with the support of users.

2 RELATED WORKS

Many studies [2, 4, 8, 15] focused on deduplicating using traditional distance metrics[1] such as Euclidean, Jaccard, Levenshtein, Jaro, and Jaro-Winkler distances. Those traditional string distances have some limitations on the understanding of natural language representation. Zhang and colleagues [5] view duplication detection as a classification problem. They use s Support Vector Machine (SVM) on a set of features, which are the cosine similarities of multiple attributes, including title, body, and title-body combination of documents. They conclude that the cosine similarity shows its superiority when measuring semantic and logical equivalent in Natural Language Processing, especially for comparing two texts.

In this work, instead of building a complete model to solve a specific problem, we want to have a generic framework that is adaptable to many situations, depending on how the user trains it. Learning from [1, 11, 14] and Dedupe.io[2], we realize that active learning is one of the most promising and fantastic ways to create the best learning scheme for the duplicate detection problem.

2.1 Active learning

Active learning is a data collection framework for training and updating machine learning models that leverages crowdsourcing for choosing the right examples to label and thus optimize the cost of the data annotation step [11, 13]. One of the popular active learning schemes is assuming that the learning algorithm can get the correct label of each sample it asks from its information annotator, also known as the worker, the expert, or the teacher. Later, it queries from unlabeled sources to obtain labels of uncertain samples, and one can combine new data labeled with the existing ones to have a new training dataset. Subsequently, one can use this dataset for enhancing the performance of the chosen machine learning model for a specific problem. For example, a two-class classifier is not sure of the right labels for a list of samples whose predicted probability is closest to 0.5 (the default threshold for classification) from the current learning model. Hence, the classifier can ask its teacher to determine the correct one and update the learning model appropriately.

2.2 Siamese models with the triplet loss

Siamese neural networks [12] are one of the useful deep neural networks; they can equip the triplet loss that can learn a similarity metric of inputs by transforming data onto a higher dimension in such a way as to maximize inter-class variation while minimizing intra-class variation [17]. Its deep metric learning's inputs are triplets (x^a, x^p, x^n), where x^a is a candidate, or an anchor, to figure positive points x^p as well as negative points x^n. The positive and negative terminologies indicate their associate labels; the same label is positive, otherwise negative. Triplet loss has been one of the most well-known loss functions which plays a vital role in many tasks in different applications, including computer vision [16], image retrieval [3, 10], face recognition [12], and sentence matching [6, 7]. For any triplet τ in all possible triplets $\mathcal{T}$, the network aims to minimize the following loss function:

$$\mathcal{L}_{\mathcal{T}} = \sum_{\tau \in \mathcal{T}} \max(0, \mathrm{d}_p - \mathrm{d}_n + \alpha), \tag{1}$$

where $\mathrm{d}_p = \mathrm{d}(\mathrm{f}(x^a), \mathrm{f}(x^p))$, $\mathrm{d}_n = \mathrm{d}(\mathrm{f}(x^a), \mathrm{f}(x^n))$, α is positive number, standing for a margin enforce between positive and negative pairs, $\mathrm{d}(\cdot, \cdot)$ is an arbitary metric, and $\mathrm{f}(\cdot)$ denotes neural network embedding.

It is not feasible to generate all possible triplets as the number of triplets is the cube of the number of training set's size, but it is possible to do sampling and generate more samples in every batch. However, triplet selection methods can influence the performance of a given learning model. As mentioned in Florian Schroff et al. [12], randomly select triplets could not yield a good result, and training time would be longer. In [9, 12], the authors suggested a strategy to find ones violate the loss equation 1, such as ranking triplets based on $\mathrm{d}_p - \mathrm{d}_n$, selecting hard triplets $\mathrm{d}_p < \mathrm{d}_n$. In this work, we will depict our sampling method in Algorithm 3.2.

[1] string distances can be found at recordlinkage.readthedocs.io

[2] https://dedupe.io/

3 OUR PROPOSED SYSTEM

In this section, we first introduce our active learning framework for Siamese neural networks. We use the triplet loss as the primary loss function to train or update our model, which needs three samples at once (one record is anchor, one record is duplicated compared to anchor, and the remaining one is not). Also, we present the proposed interface for deploying the active learning scheme as well as supporting users to get prediction results computed by trained models. We use the physical entity of organizations to demonstrate how we cooperate with the proposed active learning framework into the duplicate detection problem using the Siamese network. Data have two main attributes, "name" and "address," as depicted in Fig. 3 at the CSV Example window. All data are downloadable at Open Address[3].

3.1 Modeling

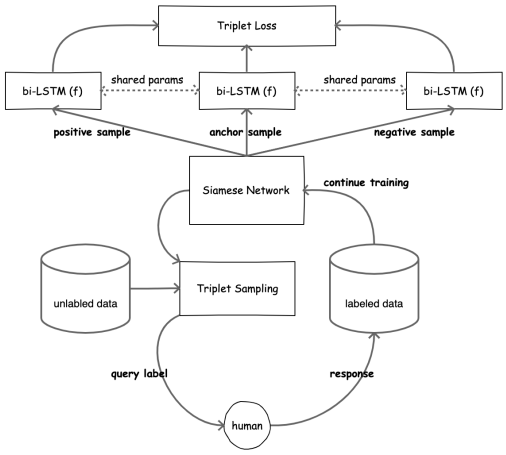

Figure 1: Our generic model that illustrates the feedback-loop and the triplet sampling mechanism.

In this work, we apply the active learning method to the Siamese Triplet Loss as the backbone. We intend to stack extensive layers and put the triplet-loss layer on the top to measure the similarity and figure out which one would be the most suitable. Our whole model can be shown in Fig. 1 and described in Algorithm 3.1.

Algorithm 3.1.

(1) *Assume that there is a model's version; otherwise, we initialize hyper-parameters of Siamese neural networks.*
(2) *The sampling module fetches unlabeled data.*
(3) *Based on the model's configuration, it samples triplets by using Algorithm 3.2.*
(4) *After obtaining new labels from users, the corresponding data as well as their labels are inserted into a labeled database.*
(5) *The Siamese neural networks then fetch the updated data from labeled database and retrain the current model. After this process, the duplicate detection system can update the new version of the model.*

[3]https://openaddresses.io/

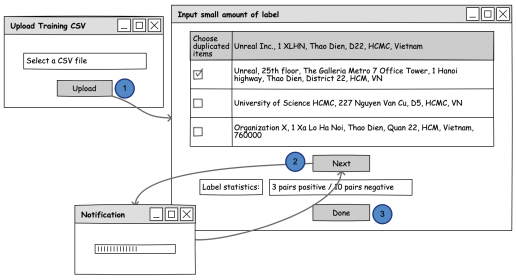

Figure 2: The interface of our active learning schema for training and updating Siamese networks using the triplet loss.

3.2 Empirical Triplet Generation

It is worth remarking that with the proposed active learning schema, the cost limitation for obtaining new labels still exists. To overcome this challenge, we can alleviate it by applying an empirical method to augment more triplets based on the number of positive pairs and negative pairs that are already known in each mini-batch. Assuming that a teacher gives two sets: one positive set $S^+ := \{ p+ \mid (x_+^a, x_+^p) \}$ and one negative set $S^- := \{ p+ \mid (x_-^a, x_-^n) \}$, we can generate triplets as defined in Algorithm 3.2.

Algorithm 3.2.

(1) *Initialize* $\mathcal{T}_{\text{random}}$.
(2) *Rank* $\mathcal{T}_{\text{random}}$ *based on* $\mathrm{d}_p - \mathrm{d}_n$.
(3) *Select top k triplets, query users for the associated labels (Fig. 2 at Input Small Amount of Label window).*
(4) *We then get a positive anchor* x_+^a *in* S^+ *as well as a positive sample* x_+^p *associated with it. We search all pairs in* S^- *as follows:*
   - *If we can find* $x_+^a = x_-^a$, *we yield* (x_+^a, x_+^p, x_-^n) *and append it to* $\mathcal{T}_{\text{random}}$.
   - *Otherwise, we randomly pick* a_-^n *in* S^-, *then yield* (x_+^a, x_+^p, x_-^n), *append it to* $\mathcal{T}_{\text{random}}$.

   Finally, $\mathcal{T}_{\text{random}} := \{ p \mid (x^a, x^p, x^n)$ *is a Triplet* $\}$.
(5) *For each pair in* S^-, *we apply an augmentation transformation "*g*"(as discussed at Section 3.2.1) on* x_-^a *to obtain* $(x_-^a, \mathrm{g}(x_-^a), x_-^n)$ *and insert it to* $\mathcal{T}_{\text{random}}$.
(6) *For each triplet in* $\mathcal{T}_{\text{random}}$, *we apply the same transformation on either* x^a, x^p, *or* x^n *and append it to* $\mathcal{T}_{\text{random}}$.

3.2.1 Augmentation Transformation. In Algorithm 3.2, we mention the augmentation scheme. There are two approaches: one is producing duplicate records; another is light random editing. It is important to note that four possible reasons to create duplicate records are wrong wording, spelling errors, acronyms, and synonyms. Operators of random editing are random insertion, swap, and deletion.

3.3 System Architecture

We depict our active learning application's interface in Fig. 2 for a duplicate address detection system using organization data in a SaaS platform.

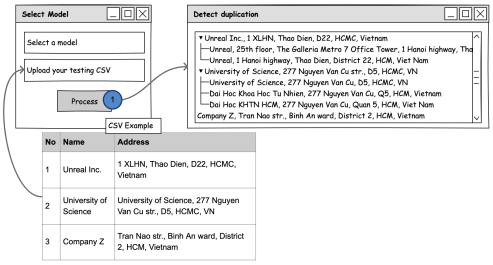

Figure 3: Users upload CSV and specify a trained model to perform their duplicate removal task. If a record has variant versions, the application will present as an expandable tree.

3.3.1 Training. In general, there are three main windows shown in Fig. 2, such as loading data, answer queries, and notification. We leverage active learning to annotate labels. To do so, there are three instructed steps, marked with the numbers inside blue circles.

(1) Users first feed a training CSV file (containing organization name and organization address) to the application.
(2) After uploading, the answer queries interface will be displayed. In each iteration, our application can select a random organization with the associated detail and recommend the suspected records. Next, users need to select the correctly duplicate items by choosing a stick on the corresponding rows, or leave blank if not. The main goal of this step is sampling data and taking answers to generate triplets for this batch. The next button is used when users complete an answer and want to fetch a new question. By this manner, our system creates a human-loop, hence in this step, we have a notification popup to show the percentage progress on the current batch when the deep learning model is running
(3) The last step is when users do not want to spend more time to label, and our application has enough minimum positive pairs and negative pairs. Users will be directed to an inference interface where they can upload testing data and explore final results.

3.3.2 Inference. Our inference phase (Fig. 3) is relatively simple as training data now hold labels from the earlier phase and mature model (the application has gradually updated it). If users want to detect duplications on their testing CSV file, they can upload it onto the system via a window and select an appropriate model version (among the existent ones) in our system. After clicking the Process button, they can leave for our application to manage until the final results are displayed.

4 CONCLUSION

In this paper, we address one of the most ubiquitous problems in the SaaS platforms, duplicate detection. We have proposed an active learning schema for training and updating a Siamese neural network using the triple loss. We have also conceptualized our application's user interface that helps users to add new labels for updating the existent detection algorithms and run duplicate detection conveniently for given input data. In the future, we aim at improving our proposed active learning schema to make sure the triple selection step is more generalized, and we can avoid potential bias that influences the performance of the learning models.

ACKNOWLEDGMENT

This research is conducted under the Collaborative Research Agreement between National Institute of Information and Communications Technology and University of Science, Vietnam National University at Ho Chi Minh City. We acknowledge the support of Science Foundation Ireland under grant number SFI/13/RC/2106 and L. Meltzers Høyskolefonds, UiB 2019/2259-NILSO.

REFERENCES

[1] Hamed H. Aghdam, Abel Gonzalez-Garcia, Joost van de Weijer, and Antonio M. López. 2019. Active Learning for Deep Detection Neural Networks. arXiv:cs.CV/1911.09168
[2] Ahmed Elmagarmid, Panos Ipeirotis, and Vassilios Verykios. 2007. Duplicate Record Detection: A Survey. *Knowledge and Data Engineering, IEEE Transactions on* 19 (02 2007), 1 – 16. https://doi.org/10.1109/TKDE.2007.250581
[3] Junshi Huang, Rogério Schmidt Feris, Qiang Chen, and Shuicheng Yan. 2015. Cross-domain Image Retrieval with a Dual Attribute-aware Ranking Network. *CoRR* abs/1505.07922 (2015). arXiv:1505.07922 http://arxiv.org/abs/1505.07922
[4] Matthew A. Jaro. 1989. Advances in Record-Linkage Methodology as Applied to Matching the 1985 Census of Tampa, Florida. *J. Amer. Statist. Assoc.* 84, 406 (1989), 414–420. https://doi.org/10.1080/01621459.1989.10478785 arXiv:https://www.tandfonline.com/doi/pdf/10.1080/01621459.1989.10478785
[5] Bruno Martins. 2011. A Supervised Machine Learning Approach for Duplicate Detection over Gazetteer Records. In *GeoSpatial Semantics*, Christophe Claramunt, Sergei Levashkin, and Michela Bertolotto (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 34–51.
[6] Jonas Mueller and Aditya Thyagarajan. 2016. Siamese Recurrent Architectures for Learning Sentence Similarity. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI'16)*. AAAI Press, 2786–2792.
[7] Paul Neculoiu, Maarten Versteegh, and Mihai Rotaru. 2016. Learning Text Similarity with Siamese Recurrent Networks. In *Proceedings of the 1st Workshop on Representation Learning for NLP*. Association for Computational Linguistics, Berlin, Germany, 148–157. https://doi.org/10.18653/v1/W16-1617
[8] Thorsten Papenbrock, Arvid Heise, and Felix Naumann. 2015. Progressive Duplicate Detection. *Knowledge and Data Engineering, IEEE Transactions on* 27 (05 2015), 1316–1329. https://doi.org/10.1109/TKDE.2014.2359666
[9] Omkar M. Parkhi, Andrea Vedaldi, and Andrew Zisserman. 2015. Deep Face Recognition. In *Proceedings of the British Machine Vision Conference (BMVC)*. BMVA Press, Article 41, 12 pages. https://doi.org/10.5244/C.29.41
[10] Filip Radenovic, Giorgos Tolias, and Ondrej Chum. 2017. Fine-tuning CNN Image Retrieval with No Human Annotation. *CoRR* abs/1711.02512 (2017). arXiv:1711.02512 http://arxiv.org/abs/1711.02512
[11] Yuji Roh, Geon Heo, and Steven Euijong Whang. 2018. A Survey on Data Collection for Machine Learning: a Big Data - AI Integration Perspective. *CoRR* abs/1811.03402 (2018). arXiv:1811.03402 http://arxiv.org/abs/1811.03402
[12] Florian Schroff, Dmitry Kalenichenko, and James Philbin. 2015. FaceNet: A Unified Embedding for Face Recognition and Clustering. *CoRR* abs/1503.03832 (2015). arXiv:1503.03832 http://arxiv.org/abs/1503.03832
[13] Burr Settles. 2009. *Active Learning Literature Survey.* Computer Sciences Technical Report 1648. University of Wisconsin–Madison. http://axon.cs.byu.edu/~martinez/classes/778/Papers/settles.activelearning.pdf
[14] Oriane Siméoni, Mateusz Budnik, Yannis Avrithis, and Guillaume Gravier. 2019. Rethinking deep active learning: Using unlabeled data at model training. arXiv:cs.CV/1911.08177
[15] Esko Ukkonen. 1992. Approximate string-matching with q-grams and maximal matches. *Theoretical Computer Science* 92, 1 (1992), 191 – 211. https://doi.org/10.1016/0304-3975(92)90143-4
[16] Jiang Wang, Yang Song, Thomas Leung, Chuck Rosenberg, Jingbin Wang, James Philbin, Bo Chen, and Ying Wu. 2014. Learning Fine-grained Image Similarity with Deep Ranking. *CoRR* abs/1404.4661 (2014). arXiv:1404.4661 http://arxiv.org/abs/1404.4661
[17] Kilian Q. Weinberger and Lawrence K. Saul. 2009. Distance Metric Learning for Large Margin Nearest Neighbor Classification. *JMLR* (2009), 207–244. http://dl.acm.org/citation.cfm?id=1577069.1577078