Applying triplet loss to entity resolution models

Charissa Ding, Lisa Kim, Derek Zhao

1 Abstract / Introduction

Entity resolution refers to the task of identifying records in a dataset that refer to the same entity across different data sources. It is one of the most crucial stages in data integration, and researchers have spent the past few decades studying various methods in order to make this process both accurate and efficient.

The most fundamental technique used in entity resolution is to calculate similarity metrics between a pair of records and setting a threshold to decide whether the pair is a match [**?**]. Additional efforts have been built upon this method, including machine-learning mechanisms such as random forest [**?**] and decision tree ensembles [**?**], as well as crowdsourcing [**?**, **?**]. Some of the ongoing research on the subject utilizes deep learning algorithms [**?**] and introduces tree-based adversarial generation [Euegen Wu proposal???]. However, these models typically cannot achieve 100% accuracy and make incorrect predictions on some records [Euegen Wu proposal???].

In recent years, researchers have made significant progress in Face Recognition utilizing a deep convolutional network trained with triplet loss function, setting a new record accuracy of 99.63% [**?**]. In this prospectus, we propose to apply this method to the entity resolution problem, in hope to improve both accuracy and efficiency.

This problem is relevant because improving accuracy on entity resolution would greatly improve the overall performance of a data integration task, which is one of the key topics in this course.

2 One Sentence Summary

We plan to adapt the use of the triplet loss function which has shown success in facial recognition systems to a deep neural network trained on triplets of database records in hopes that such a model achieves better performance than current state of the art techniques for entity resolution, such as random forests.

3 Audience and Needs

This project will be of particular value to the academic community, database managers, and data wranglers. If successful, this project would suggest a new avenue for research in entity resolution. Furthermore, a computationally reasonable model that can resolve entities with high accuracy will lead to more efficient matching, more reliable de-duplication procedures, and more accurate table merging. At its most practical, database managers and data wranglers might be saved substantial amounts of time that would otherwise be spent writing rules and manually examining recors.

4 Approach

The problem of entity resolution bears some resemblance to that of facial recognition. In both cases, we are less concerned about what a record is or who a face belongs to and more interested in whether two records or two faces are the same. Recent innovations in facial recognition technology uses neural networks trained on a triplet loss function. In a triplet loss network for facial recognition, rather than predicting the class of an input image X, the network outputs a multi-dimensional vector embedding $f(X)$ of that image with the constraint that the Euclidean

distance between two embeddings of images of two different people is significantly large. That is, let $\mathbb{I}(X)$ be the identity of the person whose face appears in X. If $\mathbb{I}(X) = \mathbb{I}(Y)$ and $\mathbb{I}(X) \neq \mathbb{I}(Z)$, then $\|f(X) - f(Z)\| \gg \|f(X) - f(Y)\|$.

A training instance for such a network consists of a triplet of three images (A, P, N), an anchor, positive, and negative image, respectively, that satisfies $\mathbb{I}(A) = \mathbb{I}(P)$ and $\mathbb{I}(A) \neq \mathbb{I}(N)$. Given said triplet, the loss that they incur on a network is $\mathcal{L}(A, D, P) = \max(0, \|f(A) - f(P)\|^2 - \|f(A) - f(N)\|^2 + \alpha)$, where α is a threshold hyperparameter defined prior to training.

The above loss function has proven successful for face recognition, and we believe it can be applied similarly to entity resolution. To generate triplets, we will use a *discriminator model* to predict whether pairs of records refer to the same entity. The mispredictions of the discriminator model will either consist of false positives which can be used as the A, N records within a triplet or false negatives which can be used as the A, P records within a triplet. Ideally, we would like to show that given an initial discriminator model, using its mispredictions as training examples for a triplet loss network results in a neural model that outperforms the discriminator.

5 (Best Case) Impact

We can show that the algorithm produces better results than the state-of-the-art approach and scales well to larger datasets.

6 Milestones

1. Identify datasets tested in previous methodologies that are computational feasible to train on and that we can use as benchmarks.
2. Develop an adversarial procedure for generating triplets. This will entail developing a tree-based ensemble and corresponding featurization procedures to generate false positives and false negatives. The accuracy of this procedure may also serve as a benchmark.
3. Implement a neural network based on the triplet loss function and compare against benchmark used to generate "hard" negatives.
4. Develop data augmentation procedures to generate positive pairs and improve the performance of the triplet loss network.
5. Run experiments on various datasets and compare results with current widespread methods.

7 Obstacles

*Major obstacles are situations where we would consider **killing** the project. Minor obstacles are situations that would delay the project or increase the overall cost in energy, time, people, and money.*

7.1 Major obstacles

- If we cannot show that a triplet loss network can at the very least improve on the performance of the initial model used to generate false positives and negatives, it is highly unlikely that said network can compare favorably to the state of the art.

7.2 Minor obstacles

- The original purpose of the triplet loss function was to enable facial recognition systems to compare new images of faces against a database of faces the system has already trained on. In other words, a facial recognition system is responsible for determining whether a new face is the same as one it has been trained to recognize, but it is not responsible for determining whether two new faces are the same. It is possible the triplet loss function is effective for only the former and not the latter, in which case a triplet loss network would require significantly more training data and computational resources to be effective.
- A typical dataset or even pair of datasets with duplicate entities largely consists of negative pairings. While an initial model such as a random forest can be used to pick out negative pairings most effective for a tripet loss network, we may be constrained by the number of positive pairings, in which case we will need to devise data augmentation methods to generate additional positive pairings to increase the number of training triplets.

- We may discover that the neural network architecture needed to effectively embed records as vectors is prohibitively large for the computational resources available. If so, we should consider training on datasets with fewer attributes or using simpler benchmark models for initially detecting false positives and negatives.

8 Additional Resources

- Some computational time to run our neural network algorithm.
- Access to a GPU-enabled server in order to perform parallel processing in the neural network model.

9 Literature Review

- *Background for the project:* This study applies recent advances in the facial recognition to the problem of entity solution.
- *Work the project relies and builds on:* Some works on face recognition have introduced the use of triplet loss function to match the people, which have turned out successful. We could take their idea and apply it to entity matching. Also, the study in DeepER has suggested word embedding as a metric to compute the string similarities.
- *Direct competitors:* As mentioned above, the study in DeepER used word embedding and deep learning to compute the string similarities and train the model. The researchers of the study has used Stochastic Gradient Descent with back propogation, and their model has outperformed the current state-of-art ML.
- *Alternatives to achieve the broader goal:* hybrid human-machine workflow and tree-based adversarial generation can be alternatives to perform entity matching more efficiently, reliably, and robustly, but they do not advance the current machine learning mechanism that is commonly used for entity matching.

10 Define Success

> *When / How do you know if you have succeeded in this project? In other words, what is the minimum finding that would make this project a success and publishable?*

Due to novelty of the idea, whether or not this neural network model can outperform current benchmarks, it is worthwhile to explore and potentially provide reference to future research in the similar direction.