Hybrid Deep Pairwise Classification for Author Name Disambiguation

Kunho Kim
Pennsylvania State University
University Park, PA
kunho@cse.psu.edu

Shaurya Rohatgi
Pennsylvania State University
University Park, PA
szr207@psu.edu

C. Lee Giles
Pennsylvania State University
University Park, PA
giles@ist.psu.edu

ABSTRACT

Author name disambiguation (AND) can be defined as the problem of clustering together unique authors from all author mentions that have been extracted from publication or related records in digital libraries or other sources. Pairwise classification is an essential part of AND, and is used to estimate the probability that any pair of author mentions belong to the same author. Previous studies trained classifiers with features manually extracted from each attribute of the data. Recently, others trained a model to learn a vector representation from text without considering any structure information. Both of these approaches have advantages. The former method takes advantage of the structure of data, while the latter takes into account the textual similarity across attributes. Here, we introduce a hybrid method which takes advantage of both approaches by extracting both structure-aware features and global features. In addition, we introduce a novel way to train a global model utilizing a large number of negative samples. Results on AMiner and PubMed data shows the relative improvement of the mean average precision (MAP) by more than 7.45% when compared to previous state-of-the-art methods.

CCS CONCEPTS

• **Information systems** → *Similarity measures.*

KEYWORDS

Author Name Disambiguation, Pairwise Classification, Gradient Boosted Trees, Representation Learning

ACM Reference Format:
Kunho Kim, Shaurya Rohatgi, and C. Lee Giles. 2019. Hybrid Deep Pairwise Classification for Author Name Disambiguation. In *The 28th ACM International Conference on Information and Knowledge Management (CIKM '19), November 3–7, 2019, Beijing, China.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3357384.3358153

1 INTRODUCTION

Identifying and clustering unique author mentions in digital libraries and other datasets is important for several reasons. Author name queries are one of the frequent searches in digital library search engines[7]. Identifying unique authors enables the allows the search engine to retrieve publication records for that unique author. Another use is the study of the science of science on large-scale data, which typically has been studied on small hand-curated datasets. Identifying unique authors is a challenging problem because the name of the author can be represented in various forms (e.g., full name or with initials), and numerous individuals have same name representations.

Author name disambiguation (AND) is the task of identifying and clustering unique authors using the metadata of publication records. Let D be a set of publication records in digital libraries. Each publication record $d_i \in D$ has an author mention a_{ij} for each author, which consists of the publication metadata (e.g. title, venue, keyword, abstract, coauthor) and author metadata (e.g. name, affiliation). The objective of AND is to cluster all author mentions $\forall a \in A$ into a set of unique authors $C = \{c_1, c_2, \cdots, c_n\}$.

Recent methods for AND typically consist of two steps. First, a blocking method is applied to divide the entire author mentions A into smaller blocks of data in order to reduce the search space for the next step. Second, clustering is done for each block separately, and the union of the clustering is from all blocks in the set of unique authors C. For clustering, a *pairwise similarity* metric needs to be defined and measured between each author mention.

Here, we focus on the *pairwise similarity* estimation. Recently, supervised machine learning-based methods have been widely used. One approach is to calculate the similarity of each attribute (e.g. title, name, affiliation), and use that as a set of features to calculate the overall similarity for various models. There are also approaches that train deep neural networks for extracting features. Beside the model they use, the two approaches are different in that the former extracts features from each attribute separately, whereas the latter uses text without considering any structure information. Both approaches have advantages. The former approach takes advantage of the structure information, since a pair of records is likely to have similar text in some of the attributes (e.g. published in the same venue and/or same affiliation). The latter approach takes account of textual similarity across attributes. This type of similarity is more robust in estimating the similarity of author pair mentions which some of the attributes are missing. For example in PubMed, abstracts are often missing for older publications, and affiliations are available only for the first and last author.

Our contribution is that we introduce a hybrid method that takes advantage of both methods. We use structure aware features as well as global features extracted from both approaches, and compliment our pairwise estimator model with gradient boosted trees (GBT). Second, we introduce a novel deep neural network to improve the quality of the extracted global features (embeddings) of each

CIKM '19, November 3–7, 2019, Beijing, China

ACM ISBN 978-1-4503-6976-3/19/11…$15.00
https://doi.org/10.1145/3357384.3358153

author mention. We evaluate our method using the AMiner dataset [17] and our PubMed dataset. Results show statistically significant improvement in accuracy (up to 7.45%) compared with previous pairwise classification methods.

2 RELATED WORK

Pairwise classification methods are an essential part of AND since they estimate the probability of a pair of author mentions belonging to the same author. Several classification methods have been used, including naive Bayes [4], support vector machines (SVM) [5], random forests (RF) [15], and GBT [8, 10]. Those methods use a manually crafted set of features that mostly consist of a set of different textual similarity measures for each attribute.

In contrast, recent work has utilized deep neural networks (DNNs) for pairwise classification. Tran et al. [14] applied a DNN as a binary classifier consisting of multiple dense layers. Their method uses a manually crafted set of features similar to classification methods introduced above, but did not utilize the ability of a DNN to learn feature representations. Atarashi et al. [1] used a Hadamard product of bag-of-words (BoW) vector as the input to a DNN in order to automatically learn features. Also other methods learn the vector representation from graphs constructed from coauthorship and document similarity [3, 16]. A drawback to such approaches is scalability, since the graph is constructed on each block and trained separately. Recently, Zhang et al. [17] used a DNN to first learn a vector representation (global embedding) for each author mention and refined it with a graph auto encoder (local embedding). The AND process can be thought as a domain specific application of entity resolution (ER). There are has been recent work on exploring DNNs on general ER problems [2, 12].

3 PAIRWISE CLASSIFICATION

We will use supervised training on a labeled dataset of author mentions. Section 3.1 and Section 3.2 introduces two types of features, and Section 3.3 discusses the selection of classifier.

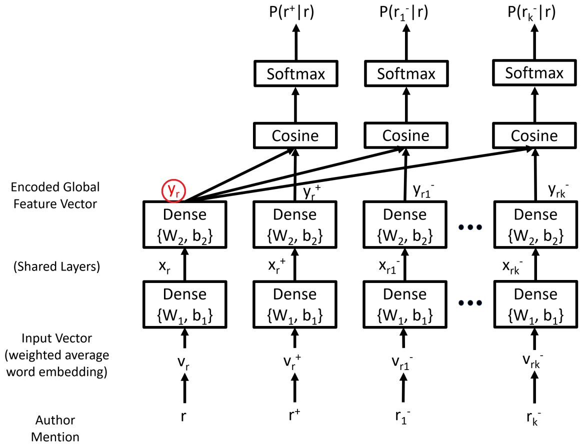
3.1 Structure-aware Features

Each author mention instance consists of a fixed set of attributes from publication and author metadata (e.g., title, venue, keyword, coauthor). It is likely that at least some of those values will have the same or similar terms if they are from the same author. For example, they may have the same venue that author mostly publishes in and similar terms used in the title. Thus, we use per-attribute similarity as our structure-aware feature. We measure two similarity metrics as used in the previous literature [9, 10, 13]:

- Cosine similarity of BoW vectors with term frequency-inverse document frequency (TF-IDF) weights.
- Cosine similarity of TF-IDF weighted average word embeddings. vector

We used our own trained word embedding from a continuous bag-of-words [11] with 100 dimensions, which is trained on all author mentions in the digital library.

Figure 1: Global features are extracted from the output of the second dense layer y_r (highlighted with a red circle).



3.2 Global Features

We also extract global features from the entire text for each author mention (i.e., a union of words from all attributes) without considering structure information. This allows us to consider the similarity of terms across attributes because semantically similar words can be also found across attributes from the same author.

We use the representation learning method of Zhang et al. [17] and extract feature vectors with a DNN with the architecture shown in Figure 1. For each author mention r, we use the TF-IDF weighted average of word embeddings, v_r. The input goes into two dense layers. The output of the second layer y_r is the encoded global feature vector.

The novelty of our model is the way it is trained. Zhang et al. [17] uses a a triplet (r, r^+, r^-) for training, where r^+, r^- is a positive/negative sample author mention of the same author. They use a triplet loss measured as the sum of $\text{sim}(y_r, y_r^+) - \text{sim}(y_r, y_r^-) + m$, where m is a margin and sim is a similarity measure. We instead use the cosine similarity of vectors.

In contrast, our model maximizes the conditional likelihood of matching the sample mention r with a positive author mention r^+, similar to the previously proposed deep structured semantic model [6]. The likelihood is calculated as posterior probability of r^+ given sample r using a softmax function,

$$P(r^+|r) = \frac{\exp(\gamma \text{sim}(r, r^+))}{\sum_{r^* \in R} \exp(\gamma \text{sim}(r, r^*))} \tag{1}$$

where R is all author mentions in the digital library, and γ is a normalization term. We use the cosine similarity of encoded vectors y_r, y_r^* for similarity measure $sim(r, r*) = cos(y_r, y_r^*)$. The denominator term is approximated by calculating the sum of similarity of positive and k negative sample pairs from r^+ and $\{r_1^-, r_2^-, \cdots, r_k^-\}$. Then, our loss function is defined by minimizing the log likelihood,

$$\text{loss} = -\log \prod_{r, r^+} P(r^+|r) \tag{2}$$

This model has the advantage of utilizing a large number of negative samples (k for each sample) produced from mentions of other authors, while the triplet model [17] sees only a single one for each sample. Thus it uses only a small portion of negative data while training. The resulting feature vector from our model shows better accuracy on pairwise classification, as discussed in Section 5.1.

We trained our model with the Adam optimizer with all hyperparameters selected using grid search. The number of hidden nodes on two dense layers are respectively 128 and 64. The mini batch size is 64 and the learning rate 0.001, and the normalization term γ in the softmax function is 35. We use 4 negative samples for each labeled author mention r, and saw that there was no further significant improvement by adding more than 4. We train up to 200 epochs, and apply early stopping if the loss in the validation set is not improving after 5 consecutive epochs.

3.3 Classifier Selection

We train a binary classifier to classify whether a pair of mentions is from same author or not. To select the proper classifier that uses both structure-aware and global features, we tested two approaches:

- **Machine learning classifiers with pairwise features**: cosine similarity of global features sim_{global} is used as an additional feature with structure-aware features
- **Pairwise DNN classifier**: structure-aware features are used as an additional input (concatenated) to the DNN model

The first type of classifiers are often used in methods that use structure-aware features only [9, 10, 13]. We tested a SVM, RF, and GBT, as they are the most used for this task.

The second approach can be seen as similar to that of a pairwise DNN [1]. Let x, y be a pair of author mentions where s_{xy} are structure-aware features between two mentions, and g_x, g_y are global features of each. The input vector of the pairwise DNN model is the Hadamard product of global features concatenated with the structure-aware features, $\{g_x \bigcirc g_y; s_{xy}\}$. The DNN is composed of two dense layers (32 nodes) and binary classification is done by optimizing the cross entropy loss function.

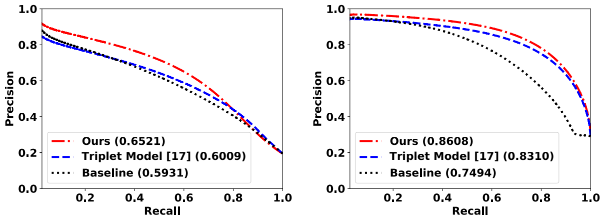
Experiment results shown in Section 5.2 indicate that the best results are from using the first approach with GBT. We believe it is because tree ensemble classifiers are known to work well on structured data, especially with several empty values, compared to DNN or linear classifiers.

4 DATA

We use two labeled datasets of digital libraries for evaluation. The first is from Zhang et al. [17] which consists of author mentions from publication records in AMiner. They manually labeled the data based on the publication list of authors’ homepage, email address, and affiliation. The second is our own dataset constructed from PubMed author mentions. We use principal investigator (PI) data from the National Institutes of Health (NIH), which has a unique identifier of PIs on NIH funded projects and their PubMed publications (given with publication identifier, PMIDs). With PMIDs, we can access the publication attributes, but we need to distinguish the PI from list of authors in order to extract author related attributes (e.g. full name, affiliation). We used the simple heuristic of checking the name compatibility [7] of the PI name and all author names. After finding all corresponding author mentions, we applied blocking using first name initials and last name, and used name blocks that had at least 3 different individuals in the block.

Table 1: Dataset Statistics and List of Attributes

Dataset	AMiner	PubMed
# Blocks	600	2,000
# Authors	12,798	9,486
# Mentions	1,121,831	1,149,692
Attributes	title, venue, affiliation, coauthor, keyword	title, venue, affiliation, coauthor, keyword, year, abstract, MeSH, chemical



(a) AMiner

(b) PubMed

Figure 2: Precision-Recall Comparison for Global Feature Methods. MAP values are in parentheses.

Table 1 shows the statistics of two datasets. The AMiner dataset has higher average number of unique authors for each block than the PubMed dataset. On the other hand, the PubMed dataset has a richer set of attributes. The AMiner dataset has only Chinese authors (e.g. Hua Fu, Jian Shao, S. Lin), whereas the PubMed dataset consisted of various ethnicities (e.g. M. Schwartz, G. Luo, D. Bhattacharya).

5 EXPERIMENTS AND DISCUSSION

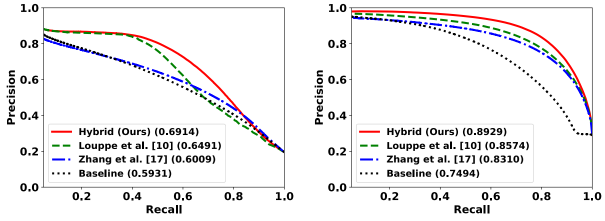
For evaluation we split the name blocks into training and test datasets with a 5:1 ratio. The training set is again split into 80% of the sampled pairs for the actual training set and the rest for the validation set. The validation set is used to check for early stopping and optimize hyperparameters. All evaluation is done by measuring the accuracy of each method with a mean average precision (MAP) measure of the precision-recall curve. This is tested on all possible pairs of author mentions within each name block of the test dataset.

5.1 Evaluation on Global Features

We evaluate the capability of identifying pairs of the same author using the cosine similarity measure of our proposed global feature vectors. We compare our proposed global features to the input TF-IDF weighted average word embedding vector of our proposed model (used as baseline) and the global embedding with the triplet model [17]. The result is shown in Figure 2. We can see a clear improvement for both datasets. The p-value of ours compared to the triplet model [17] is less than 0.001 for each dataset, which is statistically significant at level 0.05. The improvement shows our

Table 2: MAP Values of Classifiers with Proposed Features

Method	AMiner	PubMed
SVM	0.6795	0.8878
RF	0.6889	0.8907
GBT	**0.6914**	**0.8929**
Pairwise DNN [1]	0.6646	0.8771



(a) AMiner

(b) PubMed

Figure 3: Pairwise-Recall Comparison on Pairwise Classification Methods. The value in the parenthesis is MAP.

proposed model with multiple negative samples captures the global similarity better than previous methods.

5.2 Experiment on Selecting Classifiers

We tested all methods proposed in Section 3.3 by measuring the MAP of classification result. Table 2 shows that GBT has highest MAP for both datasets, following by random forest. Tree ensemble methods work well especially on publication data, which occasionally has empty values on some attributes. Although some of the decision trees can be affected by the empty values, other trees not using those attributes can still make the right decision. Thus, we use GBT on our hybrid pairwise classification method.

5.3 Evaluation on Pairwise Classification Methods

We compare our method to state-of-the-art methods, as shown in Figure 3. We compare with Louppe et al. [10], which uses tree ensemble classifiers with the set of similarity measures from each attribute. We use the set of features from Section 3.1 since they gave the best results on our experiment. Another is Zhang et al. [17] which learns vector representation of each author mention using various DNN models. In their original paper, the authors reported local embedding gives the best result. However in our experiments, global embedding gave the best results for the majority of our test dataset[1]. As such we compared our results with their global embedding results. Our hybrid method gives the best result, showing that using both structure-aware and global features improves the classification performance.

ACKNOWLEDGMENTS

We gratefully acknowledge partial support from the National Science Foundation and the National Bureau for Economic Research.

[1] The original implementation of [17] filtered out small size clusters while we keep them. The local embedding did not work well when those were included for evaluation.

6 CONCLUSION

We devised a hybrid pairwise classification method for the author name disambiguation that estimates the probability that a pair of author mentions are the same author. Two types of features are used. The first is structure-aware features extracted from the similarities of each attribute. The second is global features extracted from texts across these attributes which finds semantic similarity across attributes. In addition, we propose a selection of the pairwise classifier, and show that using Gradient Boosted Trees (GBT) performs best on our proposed features. Evaluation on Aminer and PubMed datasets shows a 7.45% relative improvement compared with previous methods using MAP on pairwise classification.

REFERENCES

[1] Kyohei Atarashi, Satoshi Oyama, Masahito Kurihara, and Kazune Furudo. 2017. A Deep Neural Network for Pairwise Classification: Enabling Feature Conjunctions and Ensuring Symmetry. In *Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD).* 83–95.

[2] Muhammad Ebraheem, Saravanan Thirumuruganathan, Shafiq Joty, Mourad Ouzzani, and Nan Tang. 2018. Distributed representations of tuples for entity resolution. *Proceedings of the VLDB Endowment* 11, 11 (2018), 1454–1467.

[3] Xiaoming Fan, Jianyong Wang, Xu Pu, Lizhu Zhou, and Bing Lv. 2011. On graph-based name disambiguation. *Journal of Data and Information Quality* 2, 2 (2011), 10.

[4] Hui Han, C. Lee Giles, Hongyuan Zha, Cheng Li, and Kostas Tsioutsiouliklis. 2004. Two supervised learning approaches for name disambiguation in author citations. In *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries (JCDL).* 296–305.

[5] Jian Huang, Seyda Ertekin, and C. Lee Giles. 2006. Efficient Name Disambiguation for Large-scale Databases. In *Proceedings of the 10th European Conference on Principle and Practice of Knowledge Discovery in Databases(PKDD’06).* 536–544. https://doi.org/10.1007/11871637_53

[6] Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. 2013. Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of the 22nd ACM international conference on Conference on information and knowledge management (CIKM).* 2333–2338.

[7] Madian Khabsa, Pucktada Treeratpituk, and C. Lee Giles. 2015. Online Person Name Disambiguation with Constraints. In *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries (JCDL).* 37–46.

[8] Kunho Kim, Athar Sefid, Bruce A Weinberg, and C Lee Giles. 2018. A Web Service for Author Name Disambiguation in Scholarly Databases. In *2018 IEEE International Conference on Web Services (ICWS).* 265–273.

[9] Michael Levin, Stefan Krawczyk, Steven Bethard, and Dan Jurafsky. 2012. Citation-based bootstrapping for large-scale author disambiguation. *Journal of the American Society for Information Science and Technology* 63, 5 (2012), 1030–1047.

[10] Gilles Louppe, Hussein T Al-Natsheh, Mateusz Susik, and Eamonn James Maguire. 2016. Ethnicity sensitive author disambiguation using semi-supervised learning. In *International Conference on Knowledge Engineering and the Semantic Web.* Springer, 272–287.

[11] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* (2013).

[12] Sidharth Mudgal, Han Li, Theodoros Rekatsinas, AnHai Doan, Youngchoon Park, Ganesh Krishnan, Rohit Deep, Esteban Arcaute, and Vijay Raghavendra. 2018. Deep learning for entity matching: A design space exploration. In *Proceedings of the 2018 International Conference on Management of Data (SIGMOD).* 19–34.

[13] Mark-Christoph Müller. 2017. Semantic Author Name Disambiguation with Word Embeddings. In *International Conference on Theory and Practice of Digital Libraries (TPDL).* Springer, 300–311.

[14] Hung Nghiep Tran, Tin Huynh, and Tien Do. 2014. Author name disambiguation by using deep neural network. In *Asian Conference on Intelligent Information and Database Systems.* Springer, 123–132.

[15] Pucktada Treeratpituk and C. Lee Giles. 2009. Disambiguating Authors in Academic Publications Using Random Forests. In *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries (JCDL).* 39–48.

[16] Baichuan Zhang and Mohammad Al Hasan. 2017. Name disambiguation in anonymized graphs using network embedding. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management (CIKM).* 1239–1248.

[17] Yutao Zhang, Fanjin Zhang, Peiran Yao, and Jie Tang. 2018. Name Disambiguation in AMiner: Clustering, Maintenance, and Human in the Loop. In *Proceedings of the 24th ACM International Conference on Knowledge Discovery & Data Mining (KDD).* 1002–1011.